BLACK CELL
Protecting critical infrastructures

MITRE ATT&CK for Enterprise (v11) Gap Analysis Report

ACME Kft.

Title	MITRE ATT&CK for Enterprise (v11) Gap Analysis Report
Prepared by	Daniel Griffiths
Reviewed by	Tibor Luter
Approved by	Tibor Luter
Date	19 January 2023
Document ID	-
Classification	TLP: AMBER

Document history			
Version	**Date**	**Short description**	**Author**
V1.0	19 January 2023	Initial.	Daniel Griffiths

External documents			
Version	**Date**	**Document type**	**Author**
-	-	-	-

Table of Contents

1. Introduction

The MITRE ATT&CK framework provides a comprehensive catalogue of the current tactics and techniques employed by adversaries throughout the cyber kill chain. It is used widely throughout the cyber security industry as it provides an objective point of reference for mapping cyber-attacks and determining defensive coverage. For these reasons, the MITRE ATT&CK framework is an excellent tool for determining an IT (or even OT) environment's security maturity.

In this assessment we will use the data sources, tactics, techniques, and mitigations that constitute the ATT&CK framework to map out the security readiness of your environment, in addition to providing actionable intelligence and mitigation steps to improve your organizations standing.

2. MITRE ATT&CK Coverage Analysis

2.1. Methodology

The assessment began with a questionnaire that allowed us to map out the various security log sources and infrastructure elements present in the environment. Using this information in conjunction with vendor documentation and the collective decades of experience Black Cell has in operating SOCs, we are able to connect the capabilities of log sources to the data source components defined in the ATT&CK framework. Using the quantities provided in the questionnaire we are also able to determine the applicability of data sources as well as their coverage of infrastructure elements. The first metric used in this assessment is a coverage report, that shows what percentage of applicable entities produce logs (or are covered by other infrastructure elements that produces logs, e.g.: a firewall) that satisfy a given ATT&CK data source component.

Often times it is simplest to illustrate a concept through an example. If your infrastructure contains 100 Windows endpoints and 50 of them have process creation logging enabled, then your coverage of the "Process: Process Creation" data source will be 50%. If your infrastructure contains 100 network devices, of which 90 are routed through a suitable firewall, then your coverage of the "Network Traffic: Network Connection Creation" data source component will be 90%.

The next metric used in the assessment is the percentage of applicable log sources that are connected to a centralized log management solution. This is calculated by simply comparing the expected number of devices against the number of devices that actually send logs to the SIEM system.

The final metric used in the assessment is the coverage of detection capabilities over the applicable attack surface. The calculation of this metric starts with identifying the detection capabilities present in the environment. These capabilities are determined in two ways. The first being a review of the detection rules present in the SIEM system (if available) and identifying what adversary techniques they protect against. Then for each detection capability, an attack surface is established based on the number of applicable entities present in the questionnaire.

Then using the attack surface and the log collection coverage, it is determined what percentage of applicable entities are protected by each detection capability.

The second method for determining detection coverage is the analysis of publicly available vendor documentations that provide details about the detection capabilities of the various security appliances present in the environment. A coverage percentage is then determined in the same fashion as with the SIEM analysis.

Yet again, it may be easiest to illustrate this concept with an example. If you have a SIEM rule that detects the “Exfiltration Over Physical Medium” technique; you have 400 devices susceptible to data exfiltration using physical media; and 200 of these devices send logs to the SIEM system, then your detection coverage for technique T1052 will be 50%.

The final overall coverage score given to a technique is the normalized weighted sum of three components. The first component is the total coverage of all data sources required to exhaustively detect a given technique. The second component is the total log collection coverage of the data sources present in the first component. The third component is the detection coverage of the given technique.

These final scores are used to colour each technique of the MITRE ATT&CK matrix to visually indicate which techniques pose the most substantial threat to your environment. The scores are then weighted using a sector specific heatmap (described in a later section), to provide you with an action plan for improving your security standing.

2.2. Data Source Coverage

Using the methodology described in the previous section, we analysed the log sources present in your environment, mapped them to MITRE ATT&CK data source components and listed them in the following table.

It is important to note that Bitdefender’s logging capabilities are not included in this section. This decision was made due to the fact that the complete set of endpoint logs cannot be actively streamed to a centralised log collection system. Although investigation packages containing detailed logs can be manually collected, these cannot be reliably sent to a SIEM system for further correlation, analysis, and storage.

MITRE ATT&CK Data Source Component	Log Source Coverage	Log Collection Coverage
Active Directory: Active Directory Credential Request	100.00%	100.00%
Active Directory: Active Directory Object Access	100.00%	100.00%
Active Directory: Active Directory Object Creation	100.00%	100.00%
Active Directory: Active Directory Object Deletion	100.00%	100.00%
Active Directory: Active Directory Object Modification	100.00%	100.00%
Application Log: Application Log Content	66.67%	33.33%
Certificate: Certificate Registration	100.00%	0.00%
Cloud Service: Cloud Service Disable	100.00%	0.00%
Cloud Service: Cloud Service Enumeration	100.00%	100.00%
Cloud Service: Cloud Service Metadata	100.00%	0.00%
Cloud Service: Cloud Service Modification	100.00%	0.00%
Cloud Storage: Cloud Storage Access	14.29%	14.29%
Cloud Storage: Cloud Storage Creation	14.29%	14.29%
Cloud Storage: Cloud Storage Deletion	14.29%	7.14%
Cloud Storage: Cloud Storage Enumeration	14.29%	11.90%
Cloud Storage: Cloud Storage Metadata	14.29%	9.52%
Cloud Storage: Cloud Storage Modification	14.29%	14.29%
Cluster: Cluster Metadata	0.00%	0.00%

Command: Command Execution	45.98%	45.98%
Container: Container Creation	70.00%	0.00%
Container: Container Enumeration	64.00%	18.00%
Container: Container Metadata	96.49%	59.65%
Container: Container Start	8.00%	0.00%
Domain Name: Active DNS	100.00%	100.00%
Domain Name: Domain Registration	100.00%	100.00%
Domain Name: Passive DNS	100.00%	100.00%
Drive: Drive Access	45.45%	45.45%
Drive: Drive Creation	45.45%	45.45%
Drive: Drive Modification	45.45%	45.45%
Driver: Driver Load	45.98%	45.98%
Driver: Driver Metadata	45.98%	45.98%
File: File Access	37.14%	29.99%
File: File Creation	72.99%	57.47%
File: File Deletion	72.99%	72.99%
File: File Metadata	67.37%	55.82%
File: File Modification	47.80%	34.23%
Firewall: Firewall Disable	100.00%	0.00%
Firewall: Firewall Enumeration	100.00%	0.00%
Firewall: Firewall Metadata	100.00%	0.00%
Firewall: Firewall Rule Modification	100.00%	0.00%
Firmware: Firmware Modification	91.95%	91.95%
Group: Group Enumeration	100.00%	98.28%
Group: Group Metadata	100.00%	98.28%

Group: Group Modification	100.00%	98.28%
Image: Image Creation	N/A	N/A
Image: Image Deletion	N/A	N/A
Image: Image Metadata	N/A	N/A
Image: Image Modification	N/A	N/A
Instance: Instance Creation	100.00%	100.00%
Instance: Instance Deletion	100.00%	100.00%
Instance: Instance Enumeration	95.00%	30.00%
Instance: Instance Metadata	95.00%	30.00%
Instance: Instance Modification	95.00%	30.00%
Instance: Instance Start	N/A	N/A
Instance: Instance Stop	N/A	N/A
Internet Scan: Response Content	80.00%	20.00%
Internet Scan: Response Metadata	80.00%	20.00%
Kernel: Kernel Module Load	0.00%	0.00%
Logon Session: Logon Session Creation	100.00%	96.57%
Logon Session: Logon Session Metadata	100.00%	96.57%
Malware Repository: Malware Content	100.00%	0.00%
Malware Repository: Malware Metadata	100.00%	0.00%
Module: Module Load	45.98%	45.98%
Named Pipe: Named Pipe Metadata	100.00%	100.00%
Network Share: Network Share Access	22.81%	14.03%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Content	64.66%	0.00%
Network Traffic: Network Traffic Flow	64.66%	0.00%

Persona: Social Media	100.00%	100.00%
Pod: Pod Creation	6.00%	8.00%
Pod: Pod Enumeration	66.00%	4.00%
Pod: Pod Metadata	44.00%	22.00%
Pod: Pod Modification	44.00%	6.00%
Process: OS API Execution	49.89%	48.17%
Process: Process Access	45.98%	45.98%
Process: Process Creation	39.25%	37.38%
Process: Process Metadata	39.25%	37.38%
Process: Process Modification	42.27%	41.24%
Process: Process Termination	39.25%	37.38%
Scheduled Job: Scheduled Job Creation	0.00%	0.00%
Scheduled Job: Scheduled Job Metadata	28.63%	13.57%
Scheduled Job: Scheduled Job Modification	28.63%	13.57%
Script: Script Execution	0.00%	0.00%
Sensor Health: Host Status	36.78%	31.06%
Service: Service Creation	0.23%	0.23%
Service: Service Metadata	0.23%	0.23%
Service: Service Modification	0.23%	0.23%
Snapshot: Snapshot Creation	42.30%	40.05%
Snapshot: Snapshot Deletion	42.30%	40.05%
Snapshot: Snapshot Enumeration	42.30%	40.05%
Snapshot: Snapshot Metadata	42.30%	40.05%
Snapshot: Snapshot Modification	36.67%	34.54%

User Account: User Account Authentication	100.00%	98.28%
User Account: User Account Creation	100.00%	98.28%
User Account: User Account Deletion	100.00%	98.28%
User Account: User Account Metadata	100.00%	98.28%
User Account: User Account Modification	66.10%	63.61%
Volume: Volume Creation	100.00%	100.00%
Volume: Volume Deletion	100.00%	100.00%
Volume: Volume Enumeration	100.00%	100.00%
Volume: Volume Metadata	100.00%	100.00%
Volume: Volume Modification	100.00%	100.00%
WMI: WMI Creation	45.98%	45.98%
Web Credential: Web Credential Creation	0.00%	0.00%
Web Credential: Web Credential Usage	0.00%	0.00%
Windows Registry: Windows Registry Key Access	100.00%	0.00%
Windows Registry: Windows Registry Key Creation	45.98%	45.98%
Windows Registry: Windows Registry Key Deletion	45.98%	45.98%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Email: Message Trace	100.00%	100.00%
Email: Threat Protection	100.00%	100.00%
CTI: Cyber Threat Data	100.00%	100.00%
CTI: ATO Information	100.00%	100.00%

Table 1: Log source and log collection coverage scores

Please note, that the email and CTI related data source components are not part of the standard MITRE ATT&CK set. They are used in this assessment to assess certain techniques more accurately.

2.3. Detection Capabilities

To determine the detection coverage of your environment we analysed the capabilities of the security solutions in your infrastructure. Each solution was extensively reviewed by either analysing the alert rule logic present in the system; reviewing system configurations; or reviewing vendor documentation in conjunction with MITRE guidance. The specifics of the assessments are described in each solution specific section.

2.3.1. FortiGate Firewalls

The capabilities of the FortiGate firewalls present in the environment were assessed by reviewing documentation available on the vendor's website[1] and scrutinizing the manufacturer stated MITRE ATT&CK mitigation coverage[2] whilst comparing these documents against the mitigation specifics listed in the ATT&CK documentation[3]. The MITRE ATT&CK mapped detection capabilities identified are listed in the following table.

Technique ID	Technique Name	Comment
T1001	Data Obfuscation	
T1007	System Service Discovery	
T1018	Remote System Discovery	
T1021	Remote Services	
T1041	Exfiltration Over C2 Channel	
T1047	Windows Management Instrumentation	Needs extending with additional tools and techniques
T1057	Process Discovery	
T1204	User Execution	
T1548	Abuse Elevation Control Mechanism	
T1583	Acquire Infrastructure	
T1592	Gather Victim Host Information	

1 https://www.fortinet.com/products/next-generation-firewall
2 https://www.fortiguard.com/mitre-mapping
3 https://attack.mitre.org/mitigations/enterprise/

T1613	Container and Resource Discovery	
T1001	Data Obfuscation	
T1007	System Service Discovery	
T1018	Remote System Discovery	
T1021	Remote Services	

Table 2: FortiGate detection coverage

2.3.2. ESET Antivirus

Engenuity is MITRE's initiative for testing the detection capabilities of various vendor's security solutions. To determine the detection capabilities of ESET Antivirus, we used the testing results from MITRE Engenuity[4] and took only those techniques that are listed under the "Antivirus/Antimalware" mitigation[5]. This is due to the fact that MITRE Engenuity evaluated ESET's EDR which adds additional functionality on top of the standard antivirus' capabilities. Unfortunately, Windows Defender specific detection capability information is not readily available. The MITRE ATT&CK mapped detection capabilities identified are listed in the following table.

Technique ID	**Technique Name**	**Comment**
T1006	Direct Volume Access	
T1011	Exfiltration Over Other Network Medium	
T1030	Data Transfer Size Limits	
T1176	Browser Extensions	
T1591	Gather Victim Org Information	
T1534	Internal Spearphishing	

Table 3: ESET Antivirus detection coverage

2.3.3. Bitdefender EDR

The capabilities of Bitdefender were also determined based on its MITRE Engenuity evaluation[6]. Unlike ESET, the results did not require any modifications in accordance with ATT&CK mitigation information. The MITRE ATT&CK mapped detection capabilities identified are listed in the following table.

---

[4] https://attackevals.mitre-engenuity.org/enterprise/participants/eset
[5] https://attack.mitre.org/mitigations/M1049/
[6] https://attackevals.mitre-engenuity.org/enterprise/participants/bitdefender

Technique ID	Technique Name	Comment
T1014	Rootkit	
T1059	Command and Scripting Interpreter	
T1074	Data Staged	
T1211	Exploitation for Defense Evasion	
T1213	Data from Information Repositories	
T1529	System Shutdown/Reboot	
T1534	Internal Spearphishing	
T1561	Disk Wipe	
T1564	Hide Artifacts	
T1589	Gather Victim Identity Information	
T1599	Network Boundary Bridging	
T1602	Data from Configuration Repository	
T1014	Rootkit	
T1059	Command and Scripting Interpreter	
T1074	Data Staged	
T1211	Exploitation for Defense Evasion	
T1213	Data from Information Repositories	
T1529	System Shutdown/Reboot	
T1534	Internal Spearphishing	
T1561	Disk Wipe	
T1564	Hide Artifacts	
T1589	Gather Victim Identity Information	
T1599	Network Boundary Bridging	

T1602	Data from Configuration Repository	
T1083	File and Directory Discovery	

Table 4: Bitdefender detection coverage

2.3.4. Microsoft Sentinel Alerts

The alert rules enabled in Sentinel were manually reviewed to determine their adequacy. We found that some enabled use cases did not have adequate log sources (or extractions) to reliably trigger, therefore these use cases were not included among the detection capabilities. The MITRE ATT&CK mapped detection capabilities identified in Sentinel are listed in the following table.

Technique ID	Technique Name	Comment
T1010	Application Window Discovery	
T1041	Exfiltration Over C2 Channel	
T1055	Process Injection	
T1106	Native API	
T1124	System Time Discovery	
T1125	Video Capture	
T1484	Domain Policy Modification	
T1535	Unused/Unsupported Cloud Regions	
T1568	Dynamic Resolution	
T1573	Encrypted Channel	
T1593	Search Open Websites/Domains	
T1608	Stage Capabilities	
T1571	Non-Standard Port	

Table 5: Sentinel detection coverage

2.4. MITRE ATT&CK Score Matrix

Using a normalized weighted sum of the data source coverage, log collection coverage and detection coverage scores, we have produced the following ATT&CK matrix that is coloured to indicate which areas of your environment have good security coverage. Green indicates a good level of maturity whilst red indicates a need for additional work. White techniques are not applicable to your environment.

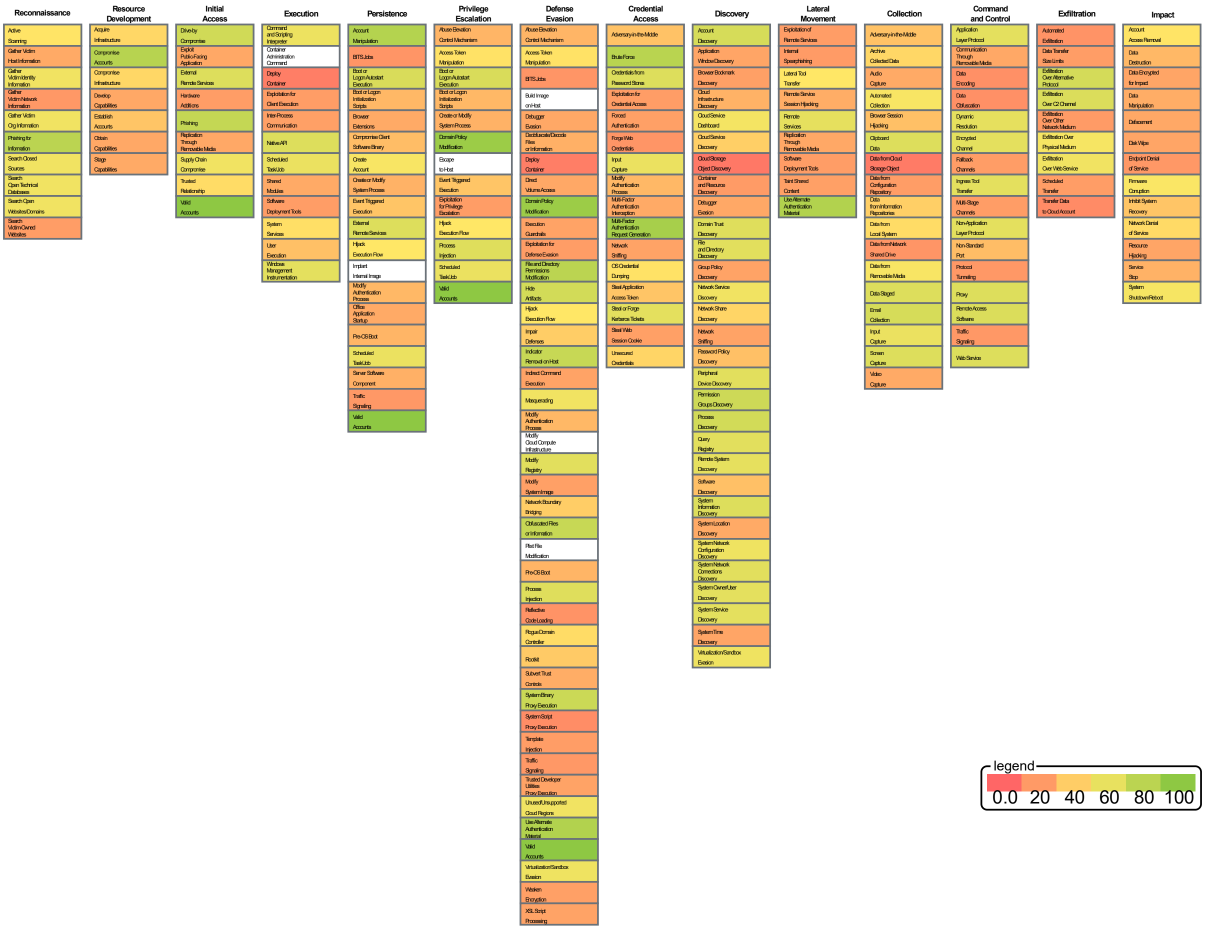

Figure 1: Overall MITRE ATT&CK coverage matrix

3. Sector Specific Analysis of Adversary TTPs

Black Cell strives to always be one step ahead of cyber criminals, which means we must always stay up to date on the threats that target our customers. However, our customers come in all shapes and sizes, therefore the adversaries that target one customer will likely use very different techniques than those who target another. In order to provide accurate and tailored recommendations, it is not enough to audit your infrastructure; we must also look towards other organizations within your market sector. In parallel with the infrastructure assessment, we have analysed the most notable threats that have led to the successful compromise of other organizations in your sector. We have collected, analysed, and mapped the TTPs used in these attacks to the MITRE ATT&CK framework in order to provide a heatmap of which techniques pose the greatest threat to your organization.

The usefulness of threat intelligence can be measured in its ability to deny cyber-attacks when adequate mitigations are in place. An excellent illustration of this concept is David Bianco's Pyramid of Pain. This simple diagram shows the relationship between the types of indicators we might use to detect an adversary's activities and how much effort or "pain" it will cause them when you are able to deny them the use of those indicators.

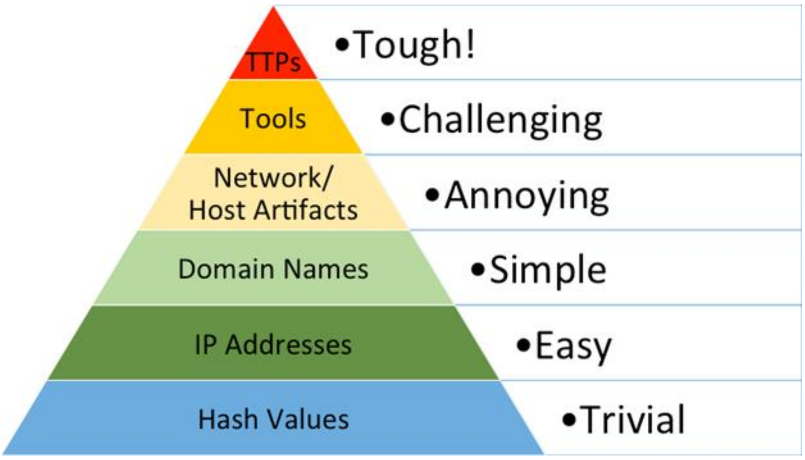

Figure 2: The Pyramid of Pain

When we are able to detect and mitigate TTPs, we are covering entire adversary behaviours, not just their tools. From a pure effectiveness standpoint, this is ideal. If we are able to prevent or react to adversary TTPs in a timely fashion, we can force them to do the most time-consuming thing possible, learn new behaviours. Therefore, with the results of this assessment in combination with the analysis of sector specific TTPs, you will receive actionable intelligence about where to focus your efforts, in order to cause as much possible headache for would-be attackers.

3.1. Methodology

There are numerous sources of historical data and high-quality analyses of cyber threats that can be used to map out sector specific TTPs. Therefore, out analysis starts with the aggregation of appropriate data in terms of quantity and quality from a range of sources. Our data gathering starts with a search of the clear web, which is essentially everything that is indexed by the most popular search engines. For our research we used Google Dork because it strongly supports targeted OSINT work. Dorking (or Google Hacking) is a technique used by security researchers that utilizes specialized queries written in Google's own query language, to find highly specialized resources. For further data enrichment we used a deep web metasearch engine, called SearX. Where applicable we also used cyber-attack information from Cyber Intel Matrix, which is a CTI platform that crawls TOR, I2P and Zeronet/Freenet sources among others.

After having identified the most substantial incidents (and threats), we used the previously described data collection methodologies to determine the specific approaches and procedures that led to the successful cyber-attack. Mapping these procedures to ATT&CK techniques is trivial and is sometimes even included in publicly available analyses. We also collected any available signatures to identify the malwares and tools that were used. Many of these tools could be directly searched in the MITRE ATT&CK resources to determine exactly what techniques they enable.

It is also not uncommon to find threat actors that operate exclusively in a given sector. It is therefore worthwhile identifying the APT groups or other criminal gangs behind the cyber-attacks under review, in order to identify trends in the methods they employ. This threat profile may contain exploitation tools, malwares, and typical techniques that they have used in previous attacks.

Finally, it is also necessary to review the security gaps that victimized the affected entity. Often times searches for such information will not be fruitful, however when this information can be gathered, it is incredibly useful. The security gaps and inadequacies that resulted in successful cyber-attacks, serve as excellent points of reflection, allowing us to consider how these gaps apply to our own environments and enable us to learn from others' mistakes.

In summary our data collection process can be broken down into the following steps.

1. Find the most relevant cyber incidents and threats.
2. Gather all available information about the incidents.
   2.1. Pinpoint the tools or malwares that were used.
   2.2. Determine attack procedures and methodologies that were used.
   2.3. Map this information to ATT&CK techniques.
3. Identify the threat actors (APT, criminal groups) and build a threat profile.
   3.1. Collect information about their tools and attack procedures.
   3.2. Map this information to ATT&CK techniques.
4. Determine the inadequacies of the victim.
   4.1. Map these security gaps to ATT&CK techniques.

Not all the information collected is of equal value. Some attack information is more impactful, and others are less relevant. Therefore, it is important to quantify the collected information in a from that can be further analysed. Part of this process is simply the mapping of attack information to ATT&CK techniques; however, we also need to assign some sort of a numerical score to each cyber threat.

As such, the following scoring system was devised. Each cyber threat was given an impact score in the range of 1-5. A score of 1 indicates the incident could be resolved in a matter of days. A score of 3 indicates that substantial and lasting damage was sustained by the victim. A score of 5 indicates a substantial risk to human life or lasting societal damage.

The threats were also given an evasion score in a range of 1-5. A score of 1 indicates the threats could have been detected by relatively simplistic signature-based detections tools, whilst a score of 5 indicates that highly sophisticated detection evasion methods were used.

A similar complexity score was also assigned to each threat, that indicates the competence, experience, and knowledge level of the adversary. A score of 1 indicates the adversary is only capable of using existing tools (colloquially a "script kiddie"), whilst a score of 5 means the adversary is capable of writing custom tailored malware.

Another important score is the proven historical successfulness of the threat. A score of 1 indicates no or partial success, while 5 indicates perfect execution and complete success in achieving its goals.

Finally, due to the volume of the data and the diversity of data sources, we also assign an accuracy multiplier, that reflects our certainty and confidence in our findings. The final scores are then mapped to ATT&CK techniques and normalised to a scale of 1-7 (1 being critical severity threats and 7 being low severity threats), before being displayed on the heatmap.

3.2. Identified Relevant Cyber Attacks

In our analysis of the logistics sector we identified the following relevant cyber threats.

3.2.1. Conti

Conti actors are known to exploit legitimate remote monitoring and management software and remote desktop software as backdoors to maintain persistence on victim networks. The actors use tools already available on the victim network—and, as needed, add additional tools, such as Windows Sysinternals and Mimikatz—to obtain users' hashes and clear-text credentials, which enable the actors to escalate privileges within a domain and perform other post-exploitation and lateral movement tasks. In some cases, the actors also use TrickBot malware to carry out post-exploitation tasks. The Cybersecurity and Infrastructure Security Agency (CISA) and the Federal Bureau of Investigation (FBI) have observed the increased use of Conti ransomware in more than 400 attacks on U.S. and international organizations.

3.2.2. BlackByte

BlackByte is ransomware as a service (RaaS) that first emerged in July 2021. Operators have exploited ProxyShell vulnerabilities to gain a foothold in the victim's environment. BlackByte has similarities to other ransomware variants such as Lockbit 2.0 that avoid systems that use Russian and a number of Eastern European languages, including many written with Cyrillic alphabets. The operators behind this ransomware have been very active since it first emerged. Since November 2021, they have targeted multiple U.S. and global organizations, including a number in energy, agriculture, financial services and the public sector. The ransomware group was made aware of the public decryptor, and this led them to create a newer version of BlackByte that uses multiple keys for each session. The encryption happens without communication with any external IPs.

3.2.3. NanoCore

The NanoCore remote access Trojan (RAT) was first discovered in 2013 when it was being sold in underground forums. The malware has a variety of functions such as keylogger, a password stealer which can remotely pass along data to the malware operator. It also has the ability to tamper and view footage from webcams, screen locking, downloading and theft of files, and more.

3.2.4. LockBit

LockBit was first observed in September 2019. Since then, it has evolved: LockBit 2.0 appeared in 2021, and the 3.0, the current version, was discovered in June 2022. LockBit ransomware has been implicated in more cyberattacks this year than any other ransomware, making it the most active ransomware in the world. And while the average ransomware payment is nearly $1 million per incident, LockBit victims pay an average ransom of approximately $85,000—indicating that LockBit targets small-to-medium-sized organizations. LockBit seeks initial access to target networks primarily through purchased access, unpatched vulnerabilities, insider access, and zero-day exploits. "Second-stage" LockBit establishes control of a victim's system, collects network information, and achieves primary goals such as stealing and encrypting data.

3.2.5. DarkSide

DarkSide is a ransomware group that was first noticed in July 2020, targeting companies all around the world. The gang conducts reconnaissance and takes precise efforts to guarantee that its attack tools and tactics will not be detected on monitored devices and endpoints. Colonial Pipeline, one of the largest and most important oil pipelines in the U.S., was compromised in a ransomware attack last May that remains one of the largest cyber attacks against U.S. critical infrastructure. The pipeline was shut down for six days as gasoline shortages impacted parts of the East Coast.

3.3. Scores and Heatmap

The following scores were assigned to each cyber threat.

Threat	Impact	Evasion	Complexity	Successfulness	Accuracy	**Score**
Conti	3	3	4	3	1	**13**
BlackByte	3	4	3	2	0,5	**6**
NanoCore	3	3	2	3	1	**11**
LockBit	4	4	4	3	1,5	**22,5**
DarkSide	4	3	5	3	1	**15**

Table 8: Sector specific threat scores

Below you can find the MITRE ATT&CK heatmap of the logistics sector. Red techniques indicate critical threats to this sector, while green techniques are less severe.

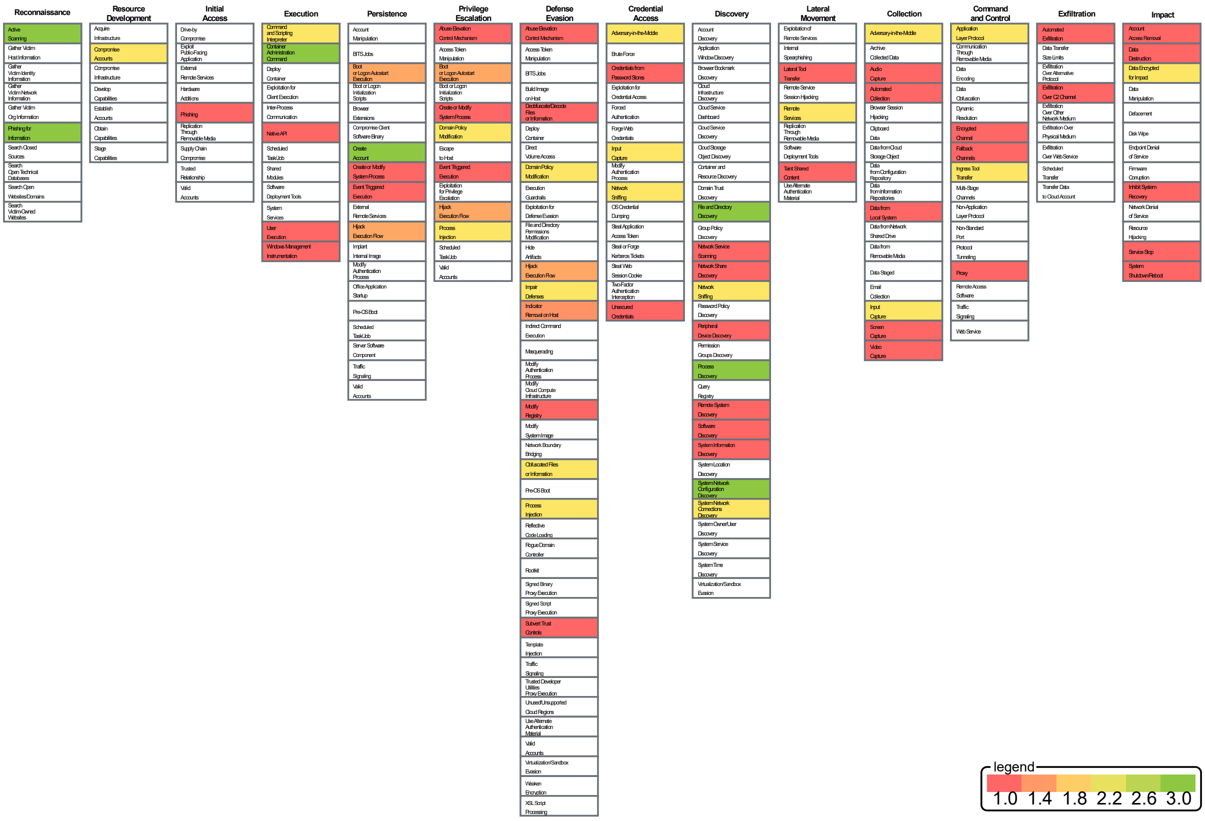

Figure 3: Sector specific threat heatmap

4. Conclusion

4.1. Heatmap Action Plan

Using the results of the MITRE ATT&CK coverage assessment and the sector specific threat heatmap, we are able to produce a summary matrix, that indicates which techniques require the most urgent mitigation in your environment. Techniques coloured in red indicate threats that should be addressed first, whilst green techniques indicate threats that can be addressed later.

Figure 4: Action plan heatmap

4.2. Detailed Technique Breakdown

In the following sections you can find all the details related to each technique that is relevant to your organization.

4.2.1. Reconnaissance

4.2.1.1. Gather Victim Identity Information (T1589)

Technique Information	
Technique ID	T1589
Technique Name	Gather Victim Identity Information
Technique Description	Adversaries may gather information about the victim's identity that can be used during targeting. Information about identities may include a variety of details, including personal data (ex: employee names, email addresses, etc.) as well as sensitive details such as credentials. Adversaries may gather this information in various ways, such as direct elicitation via [Phishing for Information](T1598).

	Information about users could also be enumerated via other active means (i.e. [Active Scanning](T1595)) such as probing and analyzing responses from authentication services that may reveal valid usernames in a system. Information about victims may also be exposed to adversaries via online or other accessible data sets (ex: [Social Media](T1593.001) or [Search Victim-Owned Websites](T1594)). Gathering this information may reveal opportunities for other forms of reconnaissance (ex: [Search Open Websites/Domains](T1593) or [Phishing for Information](T1598)), establishing operational resources (ex: [Compromise Accounts](T1586)), and/or initial access (ex: [Phishing](T1566) or [Valid Accounts](T1078)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
CTI: Cyber Threat Data	100.0%	100.0%
CTI: ATO Information	100.0%	100.0%
Email: Message Trace	100.0%	100.0%
Email: Threat Protection	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**
Detection Capability Present	**No**

Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Monitor for suspicious network traffic that could be indicative of probing for user information, such as large/iterative quantities of authentication requests originating from a single source (especially if the source is known to be associated with an adversary/botnet). Analyzing web metadata may also reveal artifacts that can be attributed to potentially malicious activity, such as referer or user-agent string HTTP/S fields. Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.2. Gather Victim Network Information (T1590)

Technique Information	
Technique ID	T1590
Technique Name	Gather Victim Network Information
Technique Description	Adversaries may gather information about the victim's networks that can be used during targeting. Information about networks may include a variety of details, including administrative data (ex: IP ranges, domain names, etc.) as well as specifics regarding its topology and operations. Adversaries may gather this information in various ways, such as direct collection actions via [Active Scanning](T1595) or [Phishing for Information](T1598). Information about networks may also be exposed to adversaries via online or other accessible data sets (ex: [Search Open Technical Databases](T1596)). Gathering this information may reveal opportunities for other forms of reconnaissance (ex: [Active Scanning](T1595) or [Search Open Websites/Domains](T1593)), establishing operational resources (ex: [Acquire Infrastructure](T1583) or [Compromise Infrastructure](T1584)), and/or initial access (ex: [Trusted Relationship](T1199)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis

Overall Score	**19.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**34/100**
Overall Log Source Coverage	**58.25%**
Overall Log Collection Coverage	**15.15%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.3. Gather Victim Org Information (T1591)

Technique Information	
Technique ID	T1591
Technique Name	Gather Victim Org Information
Technique Description	Adversaries may gather information about the victim's organization that can be used during targeting. Information about an organization may include a variety of details, including the names of divisions/departments, specifics of business operations, as well as the roles and responsibilities of key employees. Adversaries may gather this information in various ways, such as direct elicitation via [Phishing for Information](T1598). Information about an organization may also be exposed to adversaries via online or other accessible data sets (ex: [Social Media](T1593.001) or [Search Victim-Owned Websites](T1594)). Gathering this information may reveal opportunities for other forms of reconnaissance (ex: [Phishing for Information](T1598) or [Search Open Websites/Domains](T1593)), establishing operational resources (ex: [Establish Accounts](T1585) or [Compromise Accounts](T1586)), and/or initial access (ex: [Phishing](T1566) or [Trusted Relationship](T1199)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
CTI: Cyber Threat Data	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**

Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.4. *Gather Victim Host Information (T1592)*

Technique Information	
Technique ID	T1592
Technique Name	Gather Victim Host Information
Technique Description	Adversaries may gather information about the victim's hosts that can be used during targeting. Information about hosts may include a variety of details, including administrative data (ex: name, assigned IP, functionality, etc.) as well as specifics regarding its configuration (ex: operating system, language, etc.). Adversaries may gather this information in various ways, such as direct collection actions via [Active Scanning](T1595) or [Phishing for Information](T1598). Adversaries may also compromise sites then include malicious content designed to collect host information from visitors. Information about hosts may also be exposed to adversaries via online or other accessible data sets (ex: [Social Media](T1593.001) or [Search Victim-Owned Websites](T1594)). Gathering this information may reveal opportunities for other forms of reconnaissance (ex: [Search Open Websites/Domains](T1593) or [Search Open Technical Databases](T1596)), establishing operational resources (ex: [Develop Capabilities](T1587) or [Obtain Capabilities](T1588)), and/or initial access (ex: [Supply Chain Compromise](T1195) or [External Remote Services](T1133)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Internet Scan: Response Content	80.0%	20.0%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**

Sector Specific Priority	**31/100**
Overall Log Source Coverage	**80.0%**
Overall Log Collection Coverage	**20.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Internet scanners may be used to look for patterns associated with malicious content designed to collect host information from visitors. Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.5. *Search Open Websites/Domains (T1593)*

Technique Information	
Technique ID	T1593
Technique Name	Search Open Websites/Domains
Technique Description	Adversaries may search freely available websites and/or domains for information about victims that can be used during targeting. Information about victims may be available in various online sites, such as social media, new sites, or those hosting information about business operations such as hiring or requested/rewarded contracts. Adversaries may search in different online sites depending on what information they seek to gather. Information from these sources may reveal opportunities for other forms of reconnaissance (ex: [Phishing for Information](T1598) or [Search Open Technical Databases](T1596)), establishing operational resources (ex: [Establish Accounts](T1585) or [Compromise Accounts](T1586)), and/or initial access (ex: [External Remote Services](T1133) or [Phishing](T1566)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
CTI: Cyber Threat Data	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**

Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Application Developer Guidance	Application developers uploading to public code repositories should be careful to avoid publishing sensitive information such as credentials and API keys.
Audit	Scan public code repositories for exposed credentials or other sensitive information before making commits. Ensure that any leaked credentials are removed from the commit history, not just the current latest version of the code.
Implement Detection/Monitoring Capabilities	Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.6. Search Victim-Owned Websites (T1594)

Technique Information	
Technique ID	T1594
Technique Name	Search Victim-Owned Websites
Technique Description	Adversaries may search websites owned by the victim for information that can be used during targeting. Victim-owned websites may contain a variety of details, including names of departments/divisions, physical locations, and data about key employees such as names, roles, and contact info (ex: [Email Addresses](T1589.002)). These sites may also have details highlighting business operations and relationships. Adversaries may search victim-owned websites to gather actionable information. Information from these sources may reveal opportunities for other forms of reconnaissance (ex: [Phishing for Information](T1598) or [Search Open Technical Databases](T1596)), establishing operational resources (ex: [Establish Accounts](T1585) or [Compromise Accounts](T1586)), and/or initial access (ex: [Trusted Relationship](T1199) or [Phishing](T1566)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Application Log: Application Log Content	66.67%	33.33%
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**60.36%**
Overall Log Collection Coverage	**19.7%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Monitor for suspicious network traffic that could be indicative of adversary reconnaissance, such as rapid successions of requests indicative of web crawling and/or large quantities of requests originating from a single source (especially if the source is known to be associated with an adversary). Analyzing web metadata may also reveal artifacts that can be attributed to potentially malicious activity, such as referer or user-agent string HTTP/S fields.

4.2.1.7. *Active Scanning (T1595)*

Technique Information	
Technique ID	T1595
Technique Name	Active Scanning
Technique Description	Adversaries may execute active reconnaissance scans to gather information that can be used during targeting. Active scans are those where the adversary probes victim infrastructure via network traffic, as opposed to other forms of reconnaissance that do not involve direct interaction. Adversaries may perform different forms of active scanning depending on what information they seek to gather. These scans can also be performed in various ways, including using native features of network protocols such as ICMP. Information from these scans may reveal opportunities for other forms of reconnaissance (ex: [Search Open Websites/Domains](T1593) or [Search Open Technical Databases](T1596)), establishing operational resources (ex: [Develop Capabilities](T1587) or [Obtain Capabilities](T1588)), and/or initial access (ex: [External Remote Services](T1133) or [Exploit Public-Facing Application](T1190)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**48.0%**
Status	**Needs imminent remediation**

Sector Specific Priority	**39/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Monitor for suspicious network traffic that could be indicative of scanning, such as large quantities originating from a single source (especially if the source is known to be associated with an adversary/botnet). Analyzing web metadata may also reveal artifacts that can be attributed to potentially malicious activity, such as referer or user-agent string HTTP/S fields. Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.8. *Search Open Technical Databases (T1596)*

Technique Information	
Technique ID	T1596
Technique Name	Search Open Technical Databases
Technique Description	Adversaries may search freely available technical databases for information about victims that can be used during targeting. Information about victims may be available in online databases and repositories, such as registrations of domains/certificates as well as public collections of network data/artifacts gathered from traffic and/or scans. Adversaries may search in different open databases depending on what information they seek to gather. Information from these sources may reveal opportunities for other forms of reconnaissance (ex: [Phishing for Information](T1598) or [Search Open Websites/Domains](T1593)), establishing operational resources (ex: [Acquire Infrastructure](T1583) or [Compromise Infrastructure](T1584)), and/or initial access (ex: [External Remote Services](T1133) or [Trusted Relationship](T1199)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
CTI: Cyber Threat Data	100.0%	100.0%
CTI: ATO Information	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**

Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.9. Search Closed Sources (T1597)

Technique Information	
Technique ID	T1597
Technique Name	Search Closed Sources
Technique Description	Adversaries may search and gather information about victims from closed sources that can be used during targeting. Information about victims may be available for purchase from reputable private sources and databases, such as paid subscriptions to feeds of technical/threat intelligence data. Adversaries may also purchase information from less-reputable sources such as dark web or cybercrime blackmarkets. Adversaries may search in different closed databases depending on what information they seek to gather. Information from these sources may reveal opportunities for other forms of reconnaissance (ex: [Phishing for Information](T1598) or [Search Open Websites/Domains](T1593)), establishing operational resources (ex: [Develop Capabilities](T1587) or [Obtain Capabilities](T1588)), and/or initial access (ex: [External Remote Services](T1133) or [Valid Accounts](T1078)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
CTI: Cyber Threat Data	100.0%	100.0%
CTI: ATO Information	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**

Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.
Implement Detection/Monitoring Capabilities	Much of this activity may have a very high occurrence and associated false positive rate, as well as potentially taking place outside the visibility of the target organization, making detection difficult for defenders. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access.

4.2.1.10. Phishing for Information (T1598)

Technique Information	
Technique ID	T1598
Technique Name	Phishing for Information
Technique Description	Adversaries may send phishing messages to elicit sensitive information that can be used during targeting. Phishing for information is an attempt to trick targets into divulging information, frequently credentials or other actionable information. Phishing for information is different from [Phishing](T1566) in that the objective is gathering data from the victim rather than executing malicious code. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass credential harvesting campaigns. Adversaries may also try to obtain information directly through the exchange of emails, instant messages, or other electronic conversation means. Phishing for information frequently involves social engineering techniques, such as posing as a source with a reason to collect information (ex: [Establish Accounts](T1585) or [Compromise Accounts](T1586)) and/or sending multiple, seemingly urgent messages.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Application Log: Application Log Content	66.67%	33.33%
Email: Message Trace	100.0%	100.0%
Email: Threat Protection	100.0%	100.0%

Technique Analysis	
Overall Score	**79.0%**
Status	**Good maturity**
Sector Specific Priority	**10/100**
Overall Log Source Coverage	**79.2%**
Overall Log Collection Coverage	**46.67%**
Detection Capability Present	**Yes**
Detection Sources	• Microsoft Defender for Office

Mitigations	
Name	**Description**
Software Configuration	Use anti-spoofing and email authentication mechanisms to filter messages based on validity checks of the sender domain (using SPF) and integrity of messages (using DKIM). Enabling these mechanisms within an organization (through policies such as DMARC) may enable recipients (intra-org and cross domain) to perform similar message filtering and validation.
User Training	Users can be trained to identify social engineering techniques and spearphishing attempts.
Implement Detection/Monitoring Capabilities	Depending on the specific method of phishing, the detections can vary. Monitor for suspicious email activity, such as numerous accounts receiving messages from a single unusual/unknown sender.

	Filtering based on DKIM+SPF or header analysis can help detect when the email sender is spoofed. When it comes to following links, monitor for references to uncategorized or known-bad sites. URL inspection within email (including expanding shortened links) can also help detect links leading to known malicious sites. Monitor social media traffic for suspicious activity, including messages requesting information as well as abnormal file or data transfers (especially those involving unknown, or otherwise suspicious accounts).

4.2.2. Resource Development

4.2.2.1. Acquire Infrastructure (T1583)

Technique Information	
Technique ID	T1583
Technique Name	Acquire Infrastructure
Technique Description	Adversaries may buy, lease, or rent infrastructure that can be used during targeting. A wide variety of infrastructure exists for hosting and orchestrating adversary operations. Infrastructure solutions include physical or cloud servers, domains, and third-party web services. Additionally, botnets are available for rent or purchase. Use of these infrastructure solutions allows an adversary to stage, launch, and execute an operation. Solutions may help adversary operations blend in with traffic that is seen as normal, such as contact to third-party web services. Depending on the implementation, adversaries may use infrastructure that makes it difficult to physically tie back to them as well as utilize infrastructure that can be rapidly provisioned, modified, and shut down.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Internet Scan: Response Metadata	80.0%	20.0%
Domain Name: Passive DNS	100.0%	100.0%
Domain Name: Domain Registration	100.0%	100.0%
Domain Name: Active DNS	100.0%	100.0%
Internet Scan: Response Content	80.0%	20.0%

Technique Analysis	
Overall Score	**43.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**24/100**
Overall Log Source Coverage	**92.0%**
Overall Log Collection Coverage	**68.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	Consider use of services that may aid in tracking of newly acquired infrastructure, such as WHOIS databases for domain registration information. Once adversaries have provisioned infrastructure (ex: a server for use in command and control), internet scans may help proactively discover adversary acquired infrastructure. Consider looking for identifiable patterns such as services listening, certificates in use, SSL/TLS negotiation features, or other response artifacts associated with adversary C2 software. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Command and Control.

4.2.2.2. Compromise Infrastructure (T1584)

Technique Information	
Technique ID	T1584
Technique Name	Compromise Infrastructure
Technique Description	Adversaries may compromise third-party infrastructure that can be used during targeting. Infrastructure solutions include physical or cloud servers, domains, and third-party web and DNS services. Instead of buying, leasing, or renting infrastructure an adversary may compromise infrastructure and use it during other phases of the adversary lifecycle. Additionally, adversaries may compromise numerous machines to form a botnet they can leverage. Use of compromised infrastructure allows an adversary to stage, launch, and execute an operation. Compromised infrastructure can help adversary operations blend in with traffic that is seen as normal, such as contact with high reputation or trusted sites. For example, adversaries may leverage compromised infrastructure (potentially also in conjunction with [Digital Certificates](T1588.004)) to further blend in and support staged information gathering and/or [Phishing](T1566) campaigns. By using compromised infrastructure, adversaries may make it difficult to tie their actions back to them. Prior to targeting, adversaries may compromise the infrastructure of other adversaries.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Internet Scan: Response Metadata	80.0%	20.0%
Domain Name: Passive DNS	100.0%	100.0%

Domain Name: Domain Registration	100.0%	100.0%
Domain Name: Active DNS	100.0%	100.0%
Internet Scan: Response Content	80.0%	20.0%

Technique Analysis	
Overall Score	**43.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**24/100**
Overall Log Source Coverage	**92.0%**
Overall Log Collection Coverage	**68.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	Consider monitoring for anomalous changes to domain registrant information and/or domain resolution information that may indicate the compromise of a domain. Efforts may need to be tailored to specific domains of interest as benign registration and resolution changes are a common occurrence on the internet. Once adversaries have provisioned compromised infrastructure (ex: a server for use in command and

	control), internet scans may help proactively discover compromised infrastructure. Consider looking for identifiable patterns such as services listening, certificates in use, SSL/TLS negotiation features, or other response artifacts associated with adversary C2 software. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Command and Control.

4.2.2.3. *Establish Accounts (T1585)*

Technique Information	
Technique ID	T1585
Technique Name	Establish Accounts
Technique Description	Adversaries may create and cultivate accounts with services that can be used during targeting. Adversaries can create accounts that can be used to build a persona to further operations. Persona development consists of the development of public information, presence, history and appropriate affiliations. This development could be applied to social media, website, or other publicly available information that could be referenced and scrutinized for legitimacy over the course of an operation using that persona or identity. For operations incorporating social engineering, the utilization of an online persona may be important. These personas may be fictitious or impersonate real people. The persona may exist on a single site or across multiple sites (ex: Facebook, LinkedIn, Twitter, Google, GitHub, Docker Hub, etc.). Establishing a persona may require development of additional documentation to make them seem real. This could include filling out profile information, developing social networks, or incorporating photos. Establishing accounts can also include the creation of accounts with email providers, which may be directly leveraged for [Phishing for Information](T1598) or [Phishing](T1566).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Persona: Social Media	100.0%	100.0%

Technique Analysis	
Overall Score	**36.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**27/100**
Overall Log Source Coverage	**82.33%**
Overall Log Collection Coverage	**50.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	Consider monitoring social media activity related to your organization. Suspicious activity may include personas claiming to work for your organization or recently created/modified accounts making numerous connection requests to accounts affiliated with your organization. Much of this activity will take place outside the visibility of the target organization, making detection of this behavior difficult. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access (ex: [Phishing](T1566)).

4.2.2.4. *Compromise Accounts (T1586)*

Technique Information	
Technique ID	T1586
Technique Name	Compromise Accounts
Technique Description	Adversaries may compromise accounts with services that can be used during targeting. For operations incorporating social engineering, the utilization of an online persona may be important. Rather than creating and cultivating accounts (i.e. [Establish Accounts](T1585)), adversaries may compromise existing accounts. Utilizing an existing persona may engender a level of trust in a potential victim if they have a relationship, or knowledge of, the compromised persona. A variety of methods exist for compromising accounts, such as gathering credentials via [Phishing for Information](T1598), purchasing credentials from third-party sites, or by brute forcing credentials (ex: password reuse from breach credential dumps). Prior to compromising accounts, adversaries may conduct Reconnaissance to inform decisions about which accounts to compromise to further their operation. Personas may exist on a single site or across multiple sites (ex: Facebook, LinkedIn, Twitter, Google, etc.). Compromised accounts may require additional development, this could include filling out or modifying profile information, further developing social networks, or incorporating photos. Adversaries may directly leverage compromised email accounts for [Phishing for Information](T1598) or [Phishing](T1566).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%

Persona: Social Media	100.0%	100.0%

Technique Analysis	
Overall Score	**81.0%**
Status	**Good maturity**
Sector Specific Priority	**17/100**
Overall Log Source Coverage	**82.33%**
Overall Log Collection Coverage	**50.0%**
Detection Capability Present	**Yes**
Detection Sources	• Sentinel

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	Consider monitoring social media activity related to your organization. Suspicious activity may include personas claiming to work for your organization or recently modified accounts making numerous connection requests to accounts affiliated with your organization. Much of this activity will take place outside the visibility of the target organization, making detection of this behavior difficult. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Initial Access (ex: [Phishing](T1566)).

4.2.2.5. Develop Capabilities (T1587)

Technique Information	
Technique ID	T1587
Technique Name	Develop Capabilities
Technique Description	Adversaries may build capabilities that can be used during targeting. Rather than purchasing, freely downloading, or stealing capabilities, adversaries may develop their own capabilities in-house. This is the process of identifying development requirements and building solutions such as malware, exploits, and self-signed certificates. Adversaries may develop capabilities to support their operations throughout numerous phases of the adversary lifecycle. As with legitimate development efforts, different skill sets may be required for developing capabilities. The skills needed may be located in-house, or may need to be contracted out. Use of a contractor may be considered an extension of that adversary's development capabilities, provided the adversary plays a role in shaping requirements and maintains a degree of exclusivity to the capability.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Malware Repository: Malware Content	100.0%	0.0%
Malware Repository: Malware Metadata	100.0%	0.0%
Internet Scan: Response Content	80.0%	20.0%

Technique Analysis	
Overall Score	**25.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**93.33%**
Overall Log Collection Coverage	**6.67%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	Consider analyzing malware for features that may be associated with the adversary and/or their developers, such as compiler used, debugging artifacts, or code similarities. Malware repositories can also be used to identify additional samples associated with the adversary and identify development patterns over time. Consider use of services that may aid in the tracking of certificates in use on sites across the Internet. In some cases it may be possible to pivot on known pieces of certificate information to uncover other adversary infrastructure. Much of this activity will take place outside the visibility of the target organization, making detection of this behavior difficult. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Defense Evasion or Command and Control.

4.2.2.6. *Obtain Capabilities (T1588)*

Technique Information	
Technique ID	T1588
Technique Name	Obtain Capabilities
Technique Description	Adversaries may buy and/or steal capabilities that can be used during targeting. Rather than developing their own capabilities in-house, adversaries may purchase, freely download, or steal them. Activities may include the acquisition of malware, software (including licenses), exploits, certificates, and information relating to vulnerabilities. Adversaries may obtain capabilities to support their operations throughout numerous phases of the adversary lifecycle. In addition to downloading free malware, software, and exploits from the internet, adversaries may purchase these capabilities from third-party entities. Third-party entities can include technology companies that specialize in malware and exploits, criminal marketplaces, or from individuals. In addition to purchasing capabilities, adversaries may steal capabilities from third-party entities (including other adversaries). This can include stealing software licenses, malware, SSL/TLS and code-signing certificates, or raiding closed databases of vulnerabilities or exploits.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Malware Repository: Malware Content	100.0%	0.0%
Malware Repository: Malware Metadata	100.0%	0.0%
Certificate: Certificate Registration	100.0%	0.0%
Internet Scan: Response Content	80.0%	20.0%

Technique Analysis	
Overall Score	**25.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**95.0%**
Overall Log Collection Coverage	**5.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	Consider analyzing malware for features that may be associated with malware providers, such as compiler used, debugging artifacts, code similarities, or even group identifiers associated with specific Malware-as-a-Service (MaaS) offerings. Malware repositories can also be used to identify additional samples associated with the developers and the adversary utilizing their services. Identifying overlaps in malware use by different adversaries may indicate malware was obtained by the adversary rather than developed by them. In some cases, identifying overlapping characteristics in malware used by different adversaries may point to a shared quartermaster. Malware repositories can also be used to identify features of tool use associated with an adversary, such as watermarks in [Cobalt Strike](S0154) payloads.

	Consider use of services that may aid in the tracking of newly issued certificates and/or certificates in use on sites across the Internet. In some cases it may be possible to pivot on known pieces of certificate information to uncover other adversary infrastructure. Some server-side components of adversary tools may have default values set for SSL/TLS certificates. Much of this activity will take place outside the visibility of the target organization, making detection of this behavior difficult. Detection efforts may be focused on related stages of the adversary lifecycle, such as during Defense Evasion or Command and Control.

4.2.2.7. *Stage Capabilities (T1608)*

Technique Information	
Technique ID	T1608
Technique Name	Stage Capabilities
Technique Description	Adversaries may upload, install, or otherwise set up capabilities that can be used during targeting. To support their operations, an adversary may need to take capabilities they developed ([Develop Capabilities](T1587)) or obtained ([Obtain Capabilities](T1588)) and stage them on infrastructure under their control. These capabilities may be staged on infrastructure that was previously purchased/rented by the adversary ([Acquire Infrastructure](T1583)) or was otherwise compromised by them ([Compromise Infrastructure](T1584)). Capabilities can also be staged on web services, such as GitHub or Pastebin. Staging of capabilities can aid the adversary in a number of initial access and post-compromise behaviors, including (but not limited to): * Staging web resources necessary to conduct [Drive-by Compromise](T1189) when a user browses to a site. * Staging web resources for a link target to be used with spearphishing. * Uploading malware or tools to a location accessible to a victim network to enable [Ingress Tool Transfer](T1105). * Installing a previously acquired SSL/TLS certificate to use to encrypt command and control traffic (ex: [Asymmetric Cryptography](T1573.002) with [Web Protocols](T1071.001)).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Internet Scan: Response Content	80.0%	20.0%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**80.0%**
Overall Log Collection Coverage	**20.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls.
Implement Detection/Monitoring Capabilities	If infrastructure or patterns in malware, tooling, certificates, or malicious web content have been previously identified, internet scanning may uncover when an adversary has staged their capabilities. Much of this activity will take place outside the visibility of the target organization, making detection of this behavior difficult. Detection efforts may be focused on related stages of the adversary lifecycle, such as initial access and post-compromise behaviors.

4.2.3. Initial Access

4.2.3.1. Drive-by Compromise (T1189)

Technique Information	
Technique ID	T1189
Technique Name	Drive-by Compromise
Technique Description	Adversaries may gain access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behavior such as acquiring [Application Access Token](T1550.001). Multiple ways of delivering exploit code to a browser exist, including: * A legitimate website is compromised where adversaries have injected some form of malicious code such as JavaScript, iFrames, and cross-site scripting. * Malicious ads are paid for and served through legitimate ad providers. * Built-in web application interfaces are leveraged for the insertion of any other kind of object that can be used to display web content or contain a script that executes on the visiting client (e.g. forum posts, comments, and other user controllable web content). Often the website used by an adversary is one visited by a specific community, such as government, a particular industry, or region, where the goal is to compromise a specific user or set of users based on a shared interest. This kind of targeted campaign is often referred to a strategic web compromise or watering hole attack. There are several known examples of this occurring. Typical drive-by compromise process: 1. A user visits a website that is used to host the adversary controlled content. 2. Scripts automatically execute, typically searching versions of the browser and plugins for a potentially vulnerable version.

	* The user may be required to assist in this process by enabling scripting or active website components and ignoring warning dialog boxes. 3. Upon finding a vulnerable version, exploit code is delivered to the browser. 4. If exploitation is successful, then it will give the adversary code execution on the user's system unless other protections are in place. * In some cases a second visit to the website after the initial scan is required before exploit code is delivered. Unlike [Exploit Public-Facing Application](T1190), the focus of this technique is to exploit software on a client endpoint upon visiting a website. This will commonly give an adversary access to systems on the internal network instead of external systems that may be in a DMZ. Adversaries may also use compromised websites to deliver a user to a malicious application designed to [Steal Application Access Token](T1528)s, like OAuth tokens, to gain access to protected applications and information. These malicious applications have been delivered through popups on legitimate websites.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Creation	72.99%	57.47%
Network Traffic: Network Connection Creation	45.45%	45.45%
Process: Process Creation	39.25%	37.38%
Network Traffic: Network Traffic Content	64.66%	0.0%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**70.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**12/100**
Overall Log Source Coverage	**57.8%**
Overall Log Collection Coverage	**34.73%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Browser sandboxes can be used to mitigate some of the impact of exploitation, but sandbox escapes may still exist.
Exploit Protection	Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior.
Restrict Web-Based Content	For malicious code served up through ads, adblockers can help prevent that code from executing in the first place.
Update Software	Ensure all browsers and plugins kept updated can help prevent the exploit phase of this technique. Use modern browsers with security features turned on.
Implement Detection/Monitoring Capabilities	Firewalls and proxies can inspect URLs for potentially known-bad domains or parameters. They can also do reputation-based analytics on websites and their requested resources such as how old a domain is, who it's registered to, if it's on a known bad list, or how many other users have

	connected to it before. Network intrusion detection systems, sometimes with SSL/TLS inspection, can be used to look for known malicious scripts (recon, heap spray, and browser identification scripts have been frequently reused), common script obfuscation, and exploit code. Detecting compromise based on the drive-by exploit from a legitimate website may be difficult. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of browser processes. This could include suspicious files written to disk, evidence of [Process Injection](T1055) for attempts to hide execution, evidence of Discovery, or other unusual network traffic that may indicate additional tools transferred to the system.

4.2.3.2. *Exploit Public-Facing Application (T1190)*

Technique Information	
Technique ID	T1190
Technique Name	Exploit Public-Facing Application
Technique Description	Adversaries may attempt to take advantage of a weakness in an Internet-facing computer or program using software, data, or commands in order to cause unintended or unanticipated behavior. The weakness in the system can be a bug, a glitch, or a design vulnerability. These applications are often websites, but can include databases (like SQL), standard services (like SMB or SSH), network device administration and management protocols (like SNMP and Smart Install), and any other applications with Internet accessible open sockets, such as web servers and related services. Depending on the flaw being exploited this may include [Exploitation for Defense Evasion](T1211). If an application is hosted on cloud-based infrastructure and/or is containerized, then exploiting it may lead to compromise of the underlying instance or container. This can allow an adversary a path to access the cloud or container APIs, exploit container host access via [Escape to Host](T1611), or take advantage of weak identity and access management policies. For websites and databases, the OWASP top 10 and CWE top 25 highlight the most common web-based vulnerabilities.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Application Log: Application Log Content	66.67%	33.33%
Driver: Driver Load	45.98%	45.98%

Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Script: Script Execution	0.0%	0.0%
Module: Module Load	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**25.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**44.8%**
Overall Log Collection Coverage	**32.1%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Application isolation will limit what other processes and system features the exploited target can access.
Exploit Protection	Web Application Firewalls may be used to limit exposure of applications to prevent exploit traffic from reaching the application.
Network Segmentation	Segment externally facing servers and services from the rest of the network with a DMZ or on separate hosting infrastructure.

Privileged Account Management	Use least privilege for service accounts will limit what permissions the exploited process gets on the rest of the system.
Update Software	Update software regularly by employing patch management for externally exposed applications.
Vulnerability Scanning	Regularly scan externally facing systems for vulnerabilities and establish procedures to rapidly patch systems when critical vulnerabilities are discovered through scanning and through public disclosure.
Implement Detection/Monitoring Capabilities	Monitor application logs for abnormal behavior that may indicate attempted or successful exploitation. Use deep packet inspection to look for artifacts of common exploit traffic, such as SQL injection. Web Application Firewalls may detect improper inputs attempting exploitation.

4.2.3.3. Supply Chain Compromise (T1195)

Technique Information	
Technique ID	T1195
Technique Name	Supply Chain Compromise
Technique Description	Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise can take place at any stage of the supply chain including: * Manipulation of development tools * Manipulation of a development environment * Manipulation of source code repositories (public or private) * Manipulation of source code in open-source dependencies * Manipulation of software update/distribution mechanisms * Compromised/infected system images (multiple cases of removable media infected at the factory) * Replacement of legitimate software with modified versions * Sales of modified/counterfeit products to legitimate distributors * Shipment interdiction While supply chain compromise can impact any component of hardware or software, adversaries looking to gain execution have often focused on malicious additions to legitimate software in software distribution or update channels. Targeting may be specific to a desired victim set or malicious software may be distributed to a broad set of consumers but only move on to additional tactics on specific victims. Popular open source projects that are used as dependencies in many applications may also be targeted as a means to add malicious code to users of the dependency.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**

Email: Message Trace	100.0%	100.0%
Email: Threat Protection	100.0%	100.0%
CTI: Cyber Threat Data	100.0%	100.0%
CTI: ATO Information	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Update Software	A patch management process should be implemented to check unused dependencies, unmaintained and/or previously vulnerable dependencies, unnecessary features, components, files, and documentation.
Vulnerability Scanning	Continuous monitoring of vulnerability sources and the use of automatic and manual code review tools should also be implemented as well.
Implement Detection/Monitoring Capabilities	Use verification of distributed binaries through hash checking or other integrity checking mechanisms. Scan downloads for malicious signatures and attempt to test software and updates prior to deployment while taking note of potential

	suspicious activity. Perform physical inspection of hardware to look for potential tampering.

4.2.3.4. Trusted Relationship (T1199)

Technique Information	
Technique ID	T1199
Technique Name	Trusted Relationship
Technique Description	Adversaries may breach or otherwise leverage organizations who have access to intended victims. Access through trusted third party relationship exploits an existing connection that may not be protected or receives less scrutiny than standard mechanisms of gaining access to a network. Organizations often grant elevated access to second or third-party external providers in order to allow them to manage internal systems as well as cloud-based environments. Some examples of these relationships include IT services contractors, managed security providers, infrastructure contractors (e.g. HVAC, elevators, physical security). The third-party provider's access may be intended to be limited to the infrastructure being maintained, but may exist on the same network as the rest of the enterprise. As such, [Valid Accounts](T1078) used by the other party for access to internal network systems may be compromised and used.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Logon Session: Logon Session Metadata	100.0%	96.57%
Logon Session: Logon Session Creation	100.0%	96.57%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis

Overall Score	**45.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**23/100**
Overall Log Source Coverage	**88.89%**
Overall Log Collection Coverage	**75.49%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Multi-factor Authentication	Require MFA for all delegated administrator accounts.
Network Segmentation	Network segmentation can be used to isolate infrastructure components that do not require broad network access.
User Account Management	Properly manage accounts and permissions used by parties in trusted relationships to minimize potential abuse by the party and if the party is compromised by an adversary. In Office 365 environments, partner relationships and roles can be viewed under the "Partner Relationships" page.
Implement Detection/Monitoring Capabilities	Establish monitoring for activity conducted by second and third party providers and other trusted entities that may be leveraged as a means to gain access to the network. Depending on the type of relationship, an adversary may have access to significant amounts of information about the target before conducting an operation, especially if the trusted relationship is based on IT services. Adversaries may be able to act quickly towards an objective, so proper monitoring for behavior related

	to Credential Access, Lateral Movement, and Collection will be important to detect the intrusion.

4.2.3.5. Hardware Additions (T1200)

Technique Information	
Technique ID	T1200
Technique Name	Hardware Additions
Technique Description	Adversaries may introduce computer accessories, networking hardware, or other computing devices into a system or network that can be used as a vector to gain access. Rather than just connecting and distributing payloads via removable storage (i.e. [Replication Through Removable Media](T1091)), more robust hardware additions can be used to introduce new functionalities and/or features into a system that can then be abused. While public references of usage by threat actors are scarce, many red teams/penetration testers leverage hardware additions for initial access. Commercial and open source products can be leveraged with capabilities such as passive network tapping, network traffic modification (i.e. [Adversary-in-the-Middle](T1557)), keystroke injection, kernel memory reading via DMA, addition of new wireless access to an existing network, and others.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Flow	64.66%	0.0%
Drive: Drive Creation	45.45%	45.45%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**

Sector Specific Priority	**33/100**
Overall Log Source Coverage	**55.05%**
Overall Log Collection Coverage	**22.73%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Limit Access to Resource Over Network	Establish network access control policies, such as using device certificates and the 802.1x standard.
Limit Hardware Installation	Block unknown devices and accessories by endpoint security configuration and monitoring agent.
Implement Detection/Monitoring Capabilities	Asset management systems may help with the detection of computer systems or network devices that should not exist on a network. Endpoint sensors may be able to detect the addition of hardware via USB, Thunderbolt, and other external device communication ports.

4.2.3.6. *Phishing (T1566)*

Technique Information	
Technique ID	T1566
Technique Name	Phishing
Technique Description	Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns. Adversaries may send victims emails containing malicious attachments or links, typically to execute malicious code on victim systems. Phishing may also be conducted via third-party services, like social media platforms. Phishing may also involve social engineering techniques, such as posing as a trusted source.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%
File: File Creation	72.99%	57.47%
Application Log: Application Log Content	66.67%	33.33%
Email: Threat Protection	100.0%	100.0%

Technique Analysis

Overall Score	**75.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**25/100**
Overall Log Source Coverage	**73.79%**
Overall Log Collection Coverage	**38.16%**
Detection Capability Present	**Yes**
Detection Sources	• Microsoft Defender for Office

Mitigations	
Name	**Description**
Antivirus/Antimalware	Anti-virus can automatically quarantine suspicious files.
Network Intrusion Prevention	Network intrusion prevention systems and systems designed to scan and remove malicious email attachments or links can be used to block activity.
Restrict Web-Based Content	Determine if certain websites or attachment types (ex: .scr, .exe, .pif, .cpl, etc.) that can be used for phishing are necessary for business operations and consider blocking access if activity cannot be monitored well or if it poses a significant risk.
Software Configuration	Use anti-spoofing and email authentication mechanisms to filter messages based on validity checks of the sender domain (using SPF) and integrity of messages (using DKIM). Enabling these mechanisms within an organization (through policies such as DMARC) may enable recipients (intra-org and cross domain) to perform similar message filtering and validation.
User Training	Users can be trained to identify social engineering techniques and phishing emails.

Implement Detection/Monitoring Capabilities	Network intrusion detection systems and email gateways can be used to detect phishing with malicious attachments in transit. Detonation chambers may also be used to identify malicious attachments. Solutions can be signature and behavior based, but adversaries may construct attachments in a way to avoid these systems. Filtering based on DKIM+SPF or header analysis can help detect when the email sender is spoofed. URL inspection within email (including expanding shortened links) can help detect links leading to known malicious sites. Detonation chambers can be used to detect these links and either automatically go to these sites to determine if they're potentially malicious, or wait and capture the content if a user visits the link. Because most common third-party services used for phishing via service leverage TLS encryption, SSL/TLS inspection is generally required to detect the initial communication/delivery. With SSL/TLS inspection intrusion detection signatures or other security gateway appliances may be able to detect malware. Anti-virus can potentially detect malicious documents and files that are downloaded on the user's computer. Many possible detections of follow-on behavior may take place once [User Execution](T1204) occurs.

4.2.4. Execution

4.2.4.1. Windows Management Instrumentation (T1047)

Technique Information	
Technique ID	T1047
Technique Name	Windows Management Instrumentation
Technique Description	Adversaries may abuse Windows Management Instrumentation (WMI) to execute malicious commands and payloads. WMI is an administration feature that provides a uniform environment to access Windows system components. The WMI service enables both local and remote access, though the latter is facilitated by [Remote Services](T1021) such as [Distributed Component Object Model](T1021.003) (DCOM) and [Windows Remote Management](T1021.006) (WinRM). Remote WMI over DCOM operates using port 135, whereas WMI over WinRM operates over port 5985 when using HTTP and 5986 for HTTPS. An adversary can use WMI to interact with local and remote systems and use it as a means to execute various behaviors, such as gathering information for Discovery as well as remote Execution of files as part of Lateral Movement.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Connection Creation	45.45%	45.45%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**61.0%**

Status	**Needs future improvements**
Sector Specific Priority	**39/100**
Overall Log Source Coverage	**43.56%**
Overall Log Collection Coverage	**42.94%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to block processes created by WMI commands from running. Note: many legitimate tools and applications utilize WMI for command execution.
Execution Prevention	Use application control configured to block execution of
Privileged Account Management	Prevent credential overlap across systems of administrator and privileged accounts.
User Account Management	By default, only administrators are allowed to connect remotely using WMI. Restrict other users who are allowed to connect, or disallow all users to connect remotely to WMI.
Implement Detection/Monitoring Capabilities	Monitor network traffic for WMI connections; the use of WMI in environments that do not typically use WMI may be suspect. Perform process monitoring to capture command-line arguments of "wmic" and detect commands that are used to perform remote behavior.

4.2.4.2. Command and Scripting Interpreter (T1059)

Technique Information	
Technique ID	T1059
Technique Name	Command and Scripting Interpreter
Technique Description	Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](T1059.004) while Windows installations include the [Windows Command Shell](T1059.003) and [PowerShell](T1059.001). There are also cross-platform interpreters such as [Python](T1059.006), as well as those commonly associated with client applications such as [JavaScript](T1059.007) and [Visual Basic](T1059.005). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](TA0001) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](T1021) in order to achieve remote Execution.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Script: Script Execution	0.0%	0.0%
Module: Module Load	45.98%	45.98%

Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**57.0%**
Status	**Needs future improvements**
Sector Specific Priority	**39/100**
Overall Log Source Coverage	**36.22%**
Overall Log Collection Coverage	**35.5%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Antivirus/Antimalware	Anti-virus can be used to automatically quarantine suspicious files.
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent
Code Signing	Where possible, only permit execution of signed scripts.
Disable or Remove Feature or Program	Disable or remove any unnecessary or unused shells or interpreters.
Execution Prevention	Use application control where appropriate.
Privileged Account Management	When PowerShell is necessary, restrict PowerShell execution policy to administrators. Be aware that there are methods of bypassing the PowerShell execution policy, depending on environment configuration.

Restrict Web-Based Content	Script blocking extensions can help prevent the execution of scripts and HTA files that may commonly be used during the exploitation process. For malicious code served up through ads, adblockers can help prevent that code from executing in the first place.
Implement Detection/Monitoring Capabilities	Command-line and scripting activities can be captured through proper logging of process execution with command-line arguments. This information can be useful in gaining additional insight to adversaries' actions through how they use native processes or custom tools. Also monitor for loading of modules associated with specific languages. If scripting is restricted for normal users, then any attempt to enable scripts running on a system would be considered suspicious. If scripts are not commonly used on a system, but enabled, scripts running out of cycle from patching or other administrator functions are suspicious. Scripts should be captured from the file system when possible to determine their actions and intent. Scripts are likely to perform actions with various effects on a system that may generate events, depending on the types of monitoring used. Monitor processes and command-line arguments for script execution and subsequent behavior. Actions may be related to network and system information discovery, collection, or other scriptable post-compromise behaviors and could be used as indicators of detection leading back to the source script.

4.2.4.3. Native API (T1106)

Technique Information	
Technique ID	T1106
Technique Name	Native API
Technique Description	Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services within the kernel, such as those involving hardware/devices, memory, and processes. These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations. Native API functions (such as `NtCreateProcess`) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries. For example, functions such as the Windows API `CreateProcess()` or GNU `fork()` will allow programs and scripts to start other processes. This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API functions exist for various system operations. Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/portability of code. Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](T1059), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. While invoking API functions, adversaries may also attempt to bypass defensive tools (ex: unhooking monitored functions via [Disable or Modify Tools](T1562.001)).

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Process: OS API Execution	49.89%	48.17%
Module: Module Load	45.98%	45.98%

Technique Analysis	
Overall Score	**64.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**47.93%**
Overall Log Collection Coverage	**47.07%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent Office VBA macros from calling Win32 APIs.
Execution Prevention	Identify and block potentially malicious software executed that may be executed through this technique by using application control
Implement Detection/Monitoring Capabilities	Monitoring API calls may generate a significant amount of data and may not be useful for defense unless collected under specific circumstances, since benign use of API functions are common and may be difficult to distinguish from malicious behavior. Correlation of other events with behavior surrounding API function calls using API monitoring will provide additional context to an event that may

	assist in determining if it is due to malicious behavior. Correlation of activity by process lineage by process ID may be sufficient. Utilization of the Windows APIs may involve processes loading/accessing system DLLs associated with providing called functions (ex: ntdll.dll, kernel32.dll, advapi32.dll, user32.dll, and gdi32.dll). Monitoring for DLL loads, especially to abnormal/unusual or potentially malicious processes, may indicate abuse of the Windows API. Though noisy, this data can be combined with other indicators to identify adversary activity.

4.2.4.4. *Shared Modules (T1129)*

Technique Information	
Technique ID	T1129
Technique Name	Shared Modules
Technique Description	Adversaries may execute malicious payloads via loading shared modules. The Windows module loader can be instructed to load DLLs from arbitrary local paths and arbitrary Universal Naming Convention (UNC) network paths. This functionality resides in NTDLL.dll and is part of the Windows [Native API](T1106) which is called from functions like `CreateProcess`, `LoadLibrary`, etc. of the Win32 API. The module loader can load DLLs: * via specification of the (fully-qualified or relative) DLL pathname in the IMPORT directory; * via EXPORT forwarded to another DLL, specified with (fully-qualified or relative) pathname (but without extension); * via an NTFS junction or symlink program.exe.local with the fully-qualified or relative pathname of a directory containing the DLLs specified in the IMPORT directory or forwarded EXPORTs; * via `<file name="filename.extension" loadFrom="fully-qualified or relative pathname">` in an embedded or external "application manifest". The file name refers to an entry in the IMPORT directory or a forwarded EXPORT. Adversaries may use this functionality as a way to execute arbitrary payloads on a victim system. For example, malware may execute share modules to load additional components or features.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**

Process: OS API Execution	49.89%	48.17%
Module: Module Load	45.98%	45.98%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**47.93%**
Overall Log Collection Coverage	**47.07%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Execution Prevention	Identify and block potentially malicious software executed through this technique by using application control tools capable of preventing unknown DLLs from being loaded.
Implement Detection/Monitoring Capabilities	Monitoring DLL module loads may generate a significant amount of data and may not be directly useful for defense unless collected under specific circumstances, since benign use of Windows modules load functions are common and may be difficult to distinguish from malicious behavior. Legitimate software will likely only need to load routine, bundled DLL modules or Windows system DLLs such that deviation from known module loads may be suspicious. Limiting DLL module loads to `%SystemRoot%` and `%ProgramFiles%` directories will protect against module loads from unsafe paths.

	Correlation of other events with behavior surrounding module loads using API monitoring and suspicious DLLs written to disk will provide additional context to an event that may assist in determining if it is due to malicious behavior.

4.2.4.5. Exploitation for Client Execution (T1203)

Technique Information	
Technique ID	T1203
Technique Name	Exploitation for Client Execution
Technique Description	Adversaries may exploit software vulnerabilities in client applications to execute code. Vulnerabilities can exist in software due to unsecure coding practices that can lead to unanticipated behavior. Adversaries can take advantage of certain vulnerabilities through targeted exploitation for the purpose of arbitrary code execution. Oftentimes the most valuable exploits to an offensive toolkit are those that can be used to obtain code execution on a remote system because they can be used to gain access to that system. Users will expect to see files related to the applications they commonly used to do work, so they are a useful target for exploit research and development because of their high utility. Several types exist: # Browser-based Exploitation # Web browsers are a common target through [Drive-by Compromise](T1189) and [Spearphishing Link](T1566.002). Endpoint systems may be compromised through normal web browsing or from certain users being targeted by links in spearphishing emails to adversary controlled sites used to exploit the web browser. These often do not require an action by the user for the exploit to be executed. # Office Applications # Common office and productivity applications such as Microsoft Office are also targeted through [Phishing](T1566). Malicious files will be transmitted directly as attachments or through links to download them. These require the user to open the document or file for the exploit to run. # Common Third-party Applications # Other applications that are commonly seen or are part of the software deployed in a target network may also be used for

	exploitation. Applications such as Adobe Reader and Flash, which are common in enterprise environments, have been routinely targeted by adversaries attempting to gain access to systems. Depending on the software and nature of the vulnerability, some may be exploited in the browser or require the user to open a file. For instance, some Flash exploits have been delivered as objects within Microsoft Office documents.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Driver: Driver Load	45.98%	45.98%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Script: Script Execution	0.0%	0.0%
Module: Module Load	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**37.84%**
Overall Log Collection Coverage	**37.25%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Browser sandboxes can be used to mitigate some of the impact of exploitation, but sandbox escapes may still exist.
Exploit Protection	Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior.
Implement Detection/Monitoring Capabilities	Detecting software exploitation may be difficult depending on the tools available. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the browser or Office processes. This could include suspicious files written to disk, evidence of [Process Injection](T1055) for attempts to hide execution, evidence of Discovery, or other unusual network traffic that may indicate additional tools transferred to the system.

4.2.4.6. User Execution (T1204)

Technique Information	
Technique ID	T1204
Technique Name	User Execution
Technique Description	An adversary may rely upon specific actions by a user in order to gain execution. Users may be subjected to social engineering to get them to execute malicious code by, for example, opening a malicious document file or link. These user actions will typically be observed as follow-on behavior from forms of [Phishing](T1566). While [User Execution](T1204) frequently occurs shortly after Initial Access it may occur at other phases of an intrusion, such as when an adversary places a file in a shared directory or on a user's desktop hoping that a user will click on it. This activity may also be seen shortly after [Internal Spearphishing](T1534). Adversaries may also deceive users into performing actions such as enabling [Remote Access Software](T1219), allowing direct control of the system to the adversary, or downloading and executing malware for [User Execution](T1204). For example, tech support scams can be facilitated through [Phishing](T1566), vishing, or various forms of user interaction. Adversaries can use a combination of these methods, such as spoofing and promoting toll-free numbers or call centers that are used to direct victims to malicious websites, to deliver and execute payloads containing malware or [Remote Access Software](T1219).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
Instance: Instance Start	0.0%	0.0%

Network Traffic: Network Connection Creation	45.45%	45.45%
Instance: Instance Creation	100.0%	100.0%
Process: Process Creation	39.25%	37.38%
Container: Container Start	8.0%	0.0%
Network Traffic: Network Traffic Content	64.66%	0.0%
Container: Container Creation	70.0%	0.0%
Image: Image Creation	0.0%	0.0%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**34.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**66/100**
Overall Log Source Coverage	**46.64%**
Overall Log Collection Coverage	**29.06%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent executable files from running unless they meet a prevalence, age, or trusted list criteria and to prevent Office applications from

	creating potentially malicious executable content by blocking malicious code from being written to disk. Note: cloud-delivered protection must be enabled to use certain rules.
Execution Prevention	Application control may be able to prevent the running of executables masquerading as other files.
Network Intrusion Prevention	If a link is being visited by a user, network intrusion prevention systems and systems designed to scan and remove malicious downloads can be used to block activity.
Restrict Web-Based Content	If a link is being visited by a user, block unknown or unused files in transit by default that should not be downloaded or by policy from suspicious sites as a best practice to prevent some vectors, such as .scr, .exe, .pif, .cpl, etc. Some download scanning devices can open and analyze compressed and encrypted formats, such as zip and rar that may be used to conceal malicious files.
User Training	Use user training as a way to bring awareness to common phishing and spearphishing techniques and how to raise suspicion for potentially malicious events.
Implement Detection/Monitoring Capabilities	Monitor the execution of and command-line arguments for applications that may be used by an adversary to gain Initial Access that require user interaction. This includes compression applications, such as those for zip files, that can be used to [Deobfuscate/Decode Files or Information](T1140) in payloads. Anti-virus can potentially detect malicious documents and files that are downloaded and executed on the user's computer. Endpoint sensing or network sensing can potentially detect malicious events once the file is opened (such as a Microsoft Word document or PDF reaching out to the internet or spawning powershell.exe).

4.2.4.7. Inter-Process Communication (T1559)

Technique Information	
Technique ID	T1559
Technique Name	Inter-Process Communication
Technique Description	Adversaries may abuse inter-process communication (IPC) mechanisms for local code or command execution. IPC is typically used by processes to share data, communicate with each other, or synchronize execution. IPC is also commonly used to avoid situations such as deadlocks, which occurs when processes are stuck in a cyclic waiting pattern. Adversaries may abuse IPC to execute arbitrary code or commands. IPC mechanisms may differ depending on OS, but typically exists in a form accessible through programming languages/libraries or native interfaces such as Windows [Dynamic Data Exchange](T1559.002) or [Component Object Model](T1559.001). Linux environments support several different IPC mechanisms, two of which being sockets and pipes. Higher level execution mediums, such as those of [Command and Scripting Interpreter](T1059)s, may also leverage underlying IPC mechanisms. Adversaries may also use [Remote Services](T1021) such as [Distributed Component Object Model](T1021.003) to facilitate remote IPC execution.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Process: Process Access	45.98%	45.98%
Script: Script Execution	0.0%	0.0%
Process: Process Creation	39.25%	37.38%
Module: Module Load	45.98%	45.98%

Technique Analysis

Overall Score	**22.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**32.8%**
Overall Log Collection Coverage	**32.33%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Application Developer Guidance	Enable the Hardened Runtime capability when developing applications. Do not include the
Application Isolation and Sandboxing	Ensure all COM alerts and Protected View are enabled.
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent DDE attacks and spawning of child processes from Office programs.
Disable or Remove Feature or Program	Registry keys specific to Microsoft Office feature control security can be set to disable automatic DDE/OLE execution.
Privileged Account Management	Modify Registry settings (directly or using Dcomcnfg.exe) in
Software Configuration	Consider disabling embedded files in Office programs, such as OneNote, that do not work with Protected View.
Implement Detection/Monitoring Capabilities	Monitor for strings in files/commands, loaded DLLs/libraries, or spawned processes that are associated with abuse of IPC mechanisms.

4.2.4.8. System Services (T1569)

Technique Information	
Technique ID	T1569
Technique Name	System Services
Technique Description	Adversaries may abuse system services or daemons to execute commands or programs. Adversaries can execute malicious content by interacting with or creating services either locally or remotely. Many services are set to run at boot, which can aid in achieving persistence ([Create or Modify System Process](T1543)), but adversaries can also abuse services for one-time or temporary execution.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Modification	47.8%	34.23%
Service: Service Creation	0.23%	0.23%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**46.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**22/100**
Overall Log Source Coverage	**35.85%**
Overall Log Collection Coverage	**32.76%**

Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to block processes created by
Privileged Account Management	Ensure that permissions disallow services that run at a higher permissions level from being created or interacted with by a user with a lower permission level.
Restrict File and Directory Permissions	Ensure that high permission level service binaries cannot be replaced or modified by users with a lower permission level.
User Account Management	Prevent users from installing their own launch agents or launch daemons.
Implement Detection/Monitoring Capabilities	Monitor for command line invocations of tools capable of modifying services that doesn't correspond to normal usage patterns and known software, patch cycles, etc. Also monitor for changes to executables and other files associated with services. Changes to Windows services may also be reflected in the Registry.

4.2.5. Persistence

4.2.5.1. Account Manipulation (T1098)

Technique Information	
Technique ID	T1098
Technique Name	Account Manipulation
Technique Description	Adversaries may manipulate accounts to maintain access to victim systems. Account manipulation may consist of any action that preserves adversary access to a compromised account, such as modifying credentials or permission groups. These actions could also include account activity designed to subvert security policies, such as performing iterative password updates to bypass password duration policies and preserve the life of compromised credentials. In order to create or manipulate accounts, the adversary must already have sufficient permissions on systems or the domain. However, account manipulation may also lead to privilege escalation where modifications grant access to additional roles, permissions, or higher-privileged [Valid Accounts](T1078).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
User Account: User Account Modification	66.1%	63.61%
File: File Modification	47.8%	34.23%
Application Log: Application Log Content	66.67%	33.33%
Active Directory: Active Directory Object Creation	100.0%	100.0%
Active Directory: Active Directory Object Modification	100.0%	100.0%

Group: Group Modification	100.0%	98.28%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**82.0%**
Status	**Good maturity**
Sector Specific Priority	**8/100**
Overall Log Source Coverage	**70.72%**
Overall Log Collection Coverage	**64.1%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Multi-factor Authentication	Use multi-factor authentication for user and privileged accounts.
Network Segmentation	Configure access controls and firewalls to limit access to critical systems and domain controllers. Most cloud environments support separate virtual private cloud (VPC) instances that enable further segmentation of cloud systems.
Operating System Configuration	Protect domain controllers by ensuring proper security configuration for critical servers to limit access by potentially unnecessary protocols and services, such as SMB file sharing.
Privileged Account Management	Do not allow domain administrator accounts to be used for day-to-day operations that may expose them to potential adversaries on unprivileged systems.

User Account Management	Ensure that low-privileged user accounts do not have permissions to modify accounts or account-related policies.
Implement Detection/Monitoring Capabilities	Collect events that correlate with changes to account objects and/or permissions on systems and the domain, such as event IDs 4738, 4728 and 4670. Monitor for modification of accounts in correlation with other suspicious activity. Changes may occur at unusual times or from unusual systems. Especially flag events where the subject and target accounts differ or that include additional flags such as changing a password without knowledge of the old password. Monitor for use of credentials at unusual times or to unusual systems or services. This may also correlate with other suspicious activity. Monitor for unusual permissions changes that may indicate excessively broad permissions being granted to compromised accounts. However, account manipulation may also lead to privilege escalation where modifications grant access to additional roles, permissions, or higher-privileged [Valid Accounts](T1078)

4.2.5.2. *Create Account (T1136)*

Technique Information	
Technique ID	T1136
Technique Name	Create Account
Technique Description	Adversaries may create an account to maintain access to victim systems. With a sufficient level of access, creating such accounts may be used to establish secondary credentialed access that do not require persistent remote access tools to be deployed on the system. Accounts may be created on the local system or within a domain or cloud tenant. In cloud environments, adversaries may create accounts that only have access to specific services, which can reduce the chance of detection.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
User Account: User Account Creation	100.0%	98.28%
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**52.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**40/100**
Overall Log Source Coverage	**61.74%**
Overall Log Collection Coverage	**60.55%**

Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Multi-factor Authentication	Use multi-factor authentication for user and privileged accounts.
Network Segmentation	Configure access controls and firewalls to limit access to domain controllers and systems used to create and manage accounts.
Operating System Configuration	Protect domain controllers by ensuring proper security configuration for critical servers.
Privileged Account Management	Do not allow domain administrator accounts to be used for day-to-day operations that may expose them to potential adversaries on unprivileged systems.
Implement Detection/Monitoring Capabilities	Monitor for processes and command-line parameters associated with account creation, such as `net user` or `useradd`. Collect data on account creation within a network. Event ID 4720 is generated when a user account is created on a Windows system and domain controller. Perform regular audits of domain and local system accounts to detect suspicious accounts that may have been created by an adversary. Collect usage logs from cloud administrator accounts to identify unusual activity in the creation of new accounts and assignment of roles to those accounts. Monitor for accounts assigned to admin roles that go over a certain threshold of known admins.

4.2.5.3. Office Application Startup (T1137)

Technique Information	
Technique ID	T1137
Technique Name	Office Application Startup
Technique Description	Adversaries may leverage Microsoft Office-based applications for persistence between startups. Microsoft Office is a fairly common application suite on Windows-based operating systems within an enterprise network. There are multiple mechanisms that can be used with Office for persistence when an Office-based application is started; this can include the use of Office Template Macros and add-ins. A variety of features have been discovered in Outlook that can be abused to obtain persistence, such as Outlook rules, forms, and Home Page. These persistence mechanisms can work within Outlook or be used through Office 365.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
File: File Modification	47.8%	34.23%
Module: Module Load	45.98%	45.98%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Windows Registry: Windows Registry Key Creation	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis

Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**49.14%**
Overall Log Collection Coverage	**44.71%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent Office applications from creating child processes and from writing potentially malicious executable content to disk.
Disable or Remove Feature or Program	Follow Office macro security best practices suitable for your environment. Disable Office VBA macros from executing.
Software Configuration	For the Office Test method, create the Registry key used to execute it and set the permissions to "Read Control" to prevent easy access to the key without administrator permissions or requiring Privilege Escalation.
Update Software	For the Outlook methods, blocking macros may be ineffective as the Visual Basic engine used for these features is separate from the macro scripting engine.
Implement Detection/Monitoring Capabilities	Collect process execution information including process IDs (PID) and parent process IDs (PPID) and look for abnormal chains of activity resulting from Office processes. Non-standard process execution trees may also indicate suspicious or malicious behavior. If winword.exe is the parent process for

	suspicious processes and activity relating to other adversarial techniques, then it could indicate that the application was used maliciously. Many Office-related persistence mechanisms require changes to the Registry and for binaries, files, or scripts to be written to disk or existing files modified to include malicious scripts. Collect events related to Registry key creation and modification for keys that could be used for Office-based persistence. Microsoft has released a PowerShell script to safely gather mail forwarding rules and custom forms in your mail environment as well as steps to interpret the output. SensePost, whose tool [Ruler](S0358) can be used to carry out malicious rules, forms, and Home Page attacks, has released a tool to detect Ruler usage.

4.2.5.4. *Browser Extensions (T1176)*

Technique Information	
Technique ID	T1176
Technique Name	Browser Extensions
Technique Description	Adversaries may abuse Internet browser extensions to establish persistent access to victim systems. Browser extensions or plugins are small programs that can add functionality and customize aspects of Internet browsers. They can be installed directly or through a browser's app store and generally have access and permissions to everything that the browser can access. Malicious extensions can be installed into a browser through malicious app store downloads masquerading as legitimate extensions, through social engineering, or by an adversary that has already compromised a system. Security can be limited on browser app stores so it may not be difficult for malicious extensions to defeat automated scanners. Depending on the browser, adversaries may also manipulate an extension's update url to install updates from an adversary controlled server or manipulate the mobile configuration file to silently install additional extensions. Previous to macOS 11, adversaries could silently install browser extensions via the command line using the \`profiles\` tool to install malicious \`.mobileconfig\` files. In macOS 11+, the use of the \`profiles\` tool can no longer install configuration profiles, however \`.mobileconfig\` files can be planted and installed with user interaction. Once the extension is installed, it can browse to websites in the background, steal all information that a user enters into a browser (including credentials), and be used as an installer for a RAT for persistence. There have also been instances of botnets using a persistent backdoor through malicious Chrome extensions. There have also been similar examples of extensions being used for command & control.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
Network Traffic: Network Connection Creation	45.45%	45.45%
Windows Registry: Windows Registry Key Creation	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**49.93%**
Overall Log Collection Coverage	**46.45%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Audit	Ensure extensions that are installed are the intended ones as many malicious extensions will masquerade as legitimate ones.
Execution Prevention	Set a browser extension allow or deny list as appropriate for your security policy.

Limit Software Installation	Only install browser extensions from trusted sources that can be verified. Browser extensions for some browsers can be controlled through Group Policy. Change settings to prevent the browser from installing extensions without sufficient permissions.
Update Software	Ensure operating systems and browsers are using the most current version.
User Training	Close out all browser sessions when finished using them to prevent any potentially malicious extensions from continuing to run.
Implement Detection/Monitoring Capabilities	Inventory and monitor browser extension installations that deviate from normal, expected, and benign extensions. Process and network monitoring can be used to detect browsers communicating with a C2 server. However, this may prove to be a difficult way of initially detecting a malicious extension depending on the nature and volume of the traffic it generates. Monitor for any new items written to the Registry or PE files written to disk. That may correlate with browser extension installation. On macOS, monitor the command line for usage of the profiles tool, such as `profiles install -type=configuration`. Additionally, all installed extensions maintain a `plist` file in the `/Library/Managed Preferences/username/` directory. Ensure all listed files are in alignment with approved extensions.

4.2.5.5. *Server Software Component (T1505)*

Technique Information	
Technique ID	T1505
Technique Name	Server Software Component
Technique Description	Adversaries may abuse legitimate extensible development features of servers to establish persistent access to systems. Enterprise server applications may include features that allow developers to write and install software or scripts to extend the functionality of the main application. Adversaries may install malicious components to extend and abuse server applications.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
Network Traffic: Network Traffic Flow	64.66%	0.0%
File: File Modification	47.8%	34.23%
Application Log: Application Log Content	66.67%	33.33%
Module: Module Load	45.98%	45.98%
Network Traffic: Network Traffic Content	64.66%	0.0%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**28.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**30/100**
Overall Log Source Coverage	**54.88%**
Overall Log Collection Coverage	**33.37%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Audit	Regularly check component software on critical services that adversaries may target for persistence to verify the integrity of the systems and identify if unexpected changes have been made.
Code Signing	Ensure all application component binaries are signed by the correct application developers.
Disable or Remove Feature or Program	Consider disabling software components from servers when possible to prevent abuse by adversaries.
Privileged Account Management	Do not allow administrator accounts that have permissions to add component software on these services to be used for day-to-day operations that may expose them to potential adversaries on unprivileged systems.
Restrict Registry Permissions	Consider using Group Policy to configure and block modifications to service and other critical server parameters in the Registry.
User Account Management	Enforce the principle of least privilege by limiting privileges of user accounts so only authorized

	accounts can modify and/or add server software components.
Implement Detection/Monitoring Capabilities	Consider monitoring application logs for abnormal behavior that may indicate suspicious installation of application software components. Consider monitoring file locations associated with the installation of new application software components such as paths from which applications typically load such extensible components. Process monitoring may be used to detect servers components that perform suspicious actions such as running cmd.exe or accessing files. Log authentication attempts to the server and any unusual traffic patterns to or from the server and internal network.

4.2.5.6. Compromise Client Software Binary (T1554)

Technique Information	
Technique ID	T1554
Technique Name	Compromise Client Software Binary
Technique Description	Adversaries may modify client software binaries to establish persistent access to systems. Client software enables users to access services provided by a server. Common client software types are SSH clients, FTP clients, email clients, and web browsers. Adversaries may make modifications to client software binaries to carry out malicious tasks when those applications are in use. For example, an adversary may copy source code for the client software, add a backdoor, compile for the target, and replace the legitimate application binary (or support files) with the backdoored one. Since these applications may be routinely executed by the user, the adversary can leverage this for persistent access to the host.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Deletion	72.99%	72.99%
File: File Creation	72.99%	57.47%
File: File Metadata	67.37%	55.82%
File: File Modification	47.8%	34.23%

Technique Analysis	
Overall Score	**33.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**28/100**

Overall Log Source Coverage	**65.29%**
Overall Log Collection Coverage	**55.13%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Code Signing	Ensure all application component binaries are signed by the correct application developers.
Implement Detection/Monitoring Capabilities	Collect and analyze signing certificate metadata and check signature validity on software that executes within the environment. Look for changes to client software that do not correlate with known software or patch cycles. Consider monitoring for anomalous behavior from client applications, such as atypical module loads, file reads/writes, or network connections.

4.2.6. Privilege Escalation

4.2.6.1. Exploitation for Privilege Escalation (T1068)

Technique Information	
Technique ID	T1068
Technique Name	Exploitation for Privilege Escalation
Technique Description	Adversaries may exploit software vulnerabilities in an attempt to elevate privileges. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Security constructs such as permission levels will often hinder access to information and use of certain techniques, so adversaries will likely need to perform privilege escalation to include use of software exploitation to circumvent those restrictions. When initially gaining access to a system, an adversary may be operating within a lower privileged process which will prevent them from accessing certain resources on the system. Vulnerabilities may exist, usually in operating system components and software commonly running at higher permissions, that can be exploited to gain higher levels of access on the system. This could enable someone to move from unprivileged or user level permissions to SYSTEM or root permissions depending on the component that is vulnerable. This could also enable an adversary to move from a virtualized environment, such as within a virtual machine or container, onto the underlying host. This may be a necessary step for an adversary compromising an endpoint system that has been properly configured and limits other privilege escalation methods. Adversaries may bring a signed vulnerable driver onto a compromised machine so that they can exploit the vulnerability to execute code in kernel mode. This process is sometimes referred to as Bring Your Own Vulnerable Driver (BYOVD). Adversaries may include the vulnerable driver with files delivered during Initial Access or download it to a compromised system via [Ingress Tool Transfer](T1105) or [Lateral Tool Transfer](T1570).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Driver: Driver Load	45.98%	45.98%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Script: Script Execution	0.0%	0.0%
Module: Module Load	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**37.84%**
Overall Log Collection Coverage	**37.25%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing. Other types of virtualization and application microsegmentation may also mitigate the impact of

	some types of exploitation. Risks of additional exploits and weaknesses in these systems may still exist.
Execution Prevention	Consider blocking the execution of known vulnerable drivers that adversaries may exploit to execute code in kernel mode. Validate driver block rules in audit mode to ensure stability prior to production deployment.
Exploit Protection	Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior.
Threat Intelligence Program	Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization.
Update Software	Update software regularly by employing patch management for internal enterprise endpoints and servers.
Implement Detection/Monitoring Capabilities	Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the processes. This could include suspicious files written to disk, evidence of [Process Injection](T1055) for attempts to hide execution or evidence of Discovery. Consider monitoring for the presence or loading (ex: Sysmon Event ID 6) of known vulnerable drivers that adversaries may drop and exploit to execute code in kernel mode. Higher privileges are often necessary to perform additional actions such as some methods of [OS Credential Dumping](T1003). Look for additional activity that may indicate an adversary has gained higher privileges.

4.2.7. Defense Evasion

4.2.7.1. Direct Volume Access (T1006)

Technique Information	
Technique ID	T1006
Technique Name	Direct Volume Access
Technique Description	Adversaries may directly access a volume to bypass file access controls and file system monitoring. Windows allows programs to have direct access to logical volumes. Programs with direct access may read and write files directly from the drive by analyzing file system data structures. This technique bypasses Windows file access controls as well as file system monitoring tools. Utilities, such as NinjaCopy, exist to perform these actions in PowerShell.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Drive: Drive Access	45.45%	45.45%

Technique Analysis	
Overall Score	**25.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**45.72%**
Overall Log Collection Coverage	**45.72%**
Detection Capability Present	**No**

Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor handle opens on drive volumes that are made by processes to determine when they may directly access logical drives. Monitor processes and command-line arguments for actions that could be taken to copy files from the logical drive and evade common file system protections. Since this technique may also be used through [PowerShell](T1059.001), additional logging of PowerShell scripts is recommended.

4.2.7.2. *Rootkit (T1014)*

Technique Information	
Technique ID	T1014
Technique Name	Rootkit
Technique Description	Adversaries may use rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components. Rootkits are programs that hide the existence of malware by intercepting/hooking and modifying operating system API calls that supply system information. Rootkits or rootkit enabling functionality may reside at the user or kernel level in the operating system or lower, to include a hypervisor, Master Boot Record, or [System Firmware](T1542.001). Rootkits have been seen for Windows, Linux, and Mac OS X systems.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Drive: Drive Modification	45.45%	45.45%
Firmware: Firmware Modification	91.95%	91.95%

Technique Analysis	
Overall Score	**38.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**26/100**
Overall Log Source Coverage	**68.7%**
Overall Log Collection Coverage	**68.7%**
Detection Capability Present	**No**

Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Some rootkit protections may be built into anti-virus or operating system software. There are dedicated rootkit detection tools that look for specific types of rootkit behavior. Monitor for the existence of unrecognized DLLs, devices, services, and changes to the MBR.

4.2.7.3. Obfuscated Files or Information (T1027)

Technique Information	
Technique ID	T1027
Technique Name	Obfuscated Files or Information
Technique Description	Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](T1140) for [User Execution](T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. Adversaries may also used compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. Adversaries may also obfuscate commands executed from payloads or directly via a [Command and Scripting Interpreter](T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%

File: File Creation	72.99%	57.47%
File: File Metadata	67.37%	55.82%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**74.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**24/100**
Overall Log Source Coverage	**56.4%**
Overall Log Collection Coverage	**49.16%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • ESET Antivirus

Mitigations	
Name	**Description**
Antivirus/Antimalware	Anti-virus can be used to automatically detect and quarantine suspicious files. Consider utilizing the Antimalware Scan Interface (AMSI) on Windows 10 to analyze commands after being processed/interpreted.
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to prevent execution of potentially obfuscated payloads.
Implement Detection/Monitoring Capabilities	Detection of file obfuscation is difficult unless artifacts are left behind by the obfuscation process that are uniquely detectable with a signature. If detection of the obfuscation itself is not possible, it may be possible to detect the malicious activity that caused the obfuscated file (for example, the method

	that was used to write, read, or modify the file on the file system). Flag and analyze commands containing indicators of obfuscation and known suspicious syntax such as uninterpreted escape characters like '''^''' and '''"'''. Windows' Sysmon and Event ID 4688 displays command-line arguments for processes. Deobfuscation tools can be used to detect these indicators in files/payloads. Obfuscation used in payloads for Initial Access can be detected at the network. Use network intrusion detection systems and email gateway filtering to identify compressed and encrypted attachments and scripts. Some email attachment detonation systems can open compressed and encrypted attachments. Payloads delivered over an encrypted connection from a website require encrypted network traffic inspection. The first detection of a malicious tool may trigger an anti-virus or other security tool alert. Similar events may also occur at the boundary through network IDS, email scanning appliance, etc. The initial detection should be treated as an indication of a potentially more invasive intrusion. The alerting system should be thoroughly investigated beyond that initial alert for activity that was not detected. Adversaries may continue with an operation, assuming that individual events like an anti-virus detect will not be investigated or that an analyst will not be able to conclusively link that event to other activity occurring on the network.

4.2.7.4. Masquerading (T1036)

Technique Information	
Technique ID	T1036
Technique Name	Masquerading
Technique Description	Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusable system utilities to evade security monitoring is also a form of [Masquerading](T1036).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Image: Image Metadata	0.0%	0.0%
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
Scheduled Job: Scheduled Job Metadata	28.63%	13.57%
File: File Metadata	67.37%	55.82%
Process: Process Metadata	39.25%	37.38%
File: File Modification	47.8%	34.23%
Scheduled Job: Scheduled Job Modification	28.63%	13.57%
Service: Service Metadata	0.23%	0.23%

Service: Service Creation	0.23%	0.23%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**33.11%**
Overall Log Collection Coverage	**25.85%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Code Signing	Require signed binaries.
Execution Prevention	Use tools that restrict program execution via application control by attributes other than file name for common operating system utilities that are needed.
Restrict File and Directory Permissions	Use file system access controls to protect folders such as C:\\Windows\\System32.
Implement Detection/Monitoring Capabilities	Collect file hashes; file names that do not match their expected hash are suspect. Perform file monitoring; files with known names but in unusual locations are suspect. Likewise, files that are modified outside of an update or patch are suspect. If file names are mismatched between the file name on disk and that of the binary's PE metadata, this is a likely indicator that a binary was renamed after it was compiled. Collecting and comparing disk and

	resource filenames for binaries by looking to see if the InternalName, OriginalFilename, and/or ProductName match what is expected could provide useful leads, but may not always be indicative of malicious activity. Do not focus on the possible names a file could have, but instead on the command-line arguments that are known to be used and are distinct because it will have a better rate of detection. Look for indications of common characters that may indicate an attempt to trick users into misidentifying the file type, such as a space as the last character of a file name or the right-to-left override characters"\u202E", "[U+202E]", and "%E2%80%AE".

4.2.7.5. *Indicator Removal on Host (T1070)*

Technique Information	
Technique ID	T1070
Technique Name	Indicator Removal on Host
Technique Description	Adversaries may delete or modify artifacts generated on a host system to remove evidence of their presence or hinder defenses. Various artifacts may be created by an adversary or something that can be attributed to an adversary's actions. Typically these artifacts are used as defensive indicators related to monitored events, such as strings from downloaded files, logs that are generated from user actions, and other data analyzed by defenders. Location, format, and type of artifact (such as command or login history) are often specific to each platform. Removal of these indicators may interfere with event collection, reporting, or other processes used to detect intrusion activity. This may compromise the integrity of security solutions by causing notable events to go unreported. This activity may also impede forensic analysis and incident response, due to lack of sufficient data to determine what occurred.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
File: File Metadata	67.37%	55.82%
File: File Modification	47.8%	34.23%
File: File Deletion	72.99%	72.99%
User Account: User Account Authentication	100.0%	98.28%

Windows Registry: Windows Registry Key Deletion	45.98%	45.98%
Network Traffic: Network Traffic Content	64.66%	0.0%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**74.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**25/100**
Overall Log Source Coverage	**57.99%**
Overall Log Collection Coverage	**48.48%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Encrypt Sensitive Information	Obfuscate/encrypt event files locally and in transit to avoid giving feedback to an adversary.
Remote Data Storage	Automatically forward events to a log server or data repository to prevent conditions in which the adversary can locate and manipulate data on the local system. When possible, minimize time delay on event reporting to avoid prolonged storage on the local system.

Restrict File and Directory Permissions	Protect generated event files that are stored locally with proper permissions and authentication and limit opportunities for adversaries to increase privileges by preventing Privilege Escalation opportunities.
Implement Detection/Monitoring Capabilities	File system monitoring may be used to detect improper deletion or modification of indicator files. Events not stored on the file system may require different detection mechanisms.

4.2.7.6. Modify Registry (T1112)

Technique Information	
Technique ID	T1112
Technique Name	Modify Registry
Technique Description	Adversaries may interact with the Windows Registry to hide configuration information within Registry keys, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution. Access to specific areas of the Registry depends on account permissions, some requiring administrator-level access. The built-in Windows command-line utility [Reg](S0075) may be used for local or remote Registry modification. Other tools may also be used, such as a remote access tool, which may contain functionality to interact with the Registry through the Windows API. Registry modifications may also include actions to hide keys, such as prepending key names with a null character, which will cause an error and/or be ignored when read via [Reg](S0075) or other utilities using the Win32 API. Adversaries may abuse these pseudo-hidden keys to conceal payloads/commands used to maintain persistence. The Registry of a remote system may be modified to aid in execution of files as part of lateral movement. It requires the remote Registry service to be running on the target system. Often [Valid Accounts](T1078) are required, along with access to the remote system's [SMB/Windows Admin Shares](T1021.002) for RPC communication.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Windows Registry: Windows Registry Key Deletion	45.98%	45.98%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Windows Registry: Windows Registry Key Creation	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**62.0%**
Status	**Needs future improvements**
Sector Specific Priority	**38/100**
Overall Log Source Coverage	**45.51%**
Overall Log Collection Coverage	**44.91%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Restrict Registry Permissions	Ensure proper permissions are set for Registry hives to prevent users from modifying keys for system components that may lead to privilege escalation.
Implement Detection/Monitoring Capabilities	Modifications to the Registry are normal and occur throughout typical use of the Windows operating system. Consider enabling Registry Auditing on specific keys to produce an alertable event (Event ID 4657) whenever a value is changed (though this may not trigger when values are created with Reghide or other evasive methods). Changes to Registry entries that load software on Windows startup that do not

	correlate with known software, patch cycles, etc., are suspicious, as are additions or changes to files within the startup folder. Changes could also include new services and modification of existing binary paths to point to malicious files. If a change to a service-related entry occurs, then it will likely be followed by a local or remote service start or restart to execute the file. Monitor processes and command-line arguments for actions that could be taken to change or delete information in the Registry. Remote access tools with built-in features may interact directly with the Windows API to gather information. The Registry may also be modified through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001), which may require additional logging features to be configured in the operating system to collect necessary information for analysis. Monitor for processes, command-line arguments, and API calls associated with concealing Registry keys, such as Reghide. Inspect and cleanup malicious hidden Registry entries using Native Windows API calls and/or tools such as Autoruns and RegDelNull .

4.2.7.7. Trusted Developer Utilities Proxy Execution (T1127)

Technique Information	
Technique ID	T1127
Technique Name	Trusted Developer Utilities Proxy Execution
Technique Description	Adversaries may take advantage of trusted developer utilities to proxy execution of malicious payloads. There are many utilities used for software development related tasks that can be used to execute code in various forms to assist in development, debugging, and reverse engineering. These utilities may often be signed with legitimate certificates that allow them to execute on a system and proxy execution of malicious code through a trusted process that effectively bypasses application control solutions.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**23.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**42.61%**
Overall Log Collection Coverage	**41.68%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Disable or Remove Feature or Program	Specific developer utilities may not be necessary within a given environment and should be removed if not used.
Execution Prevention	Certain developer utilities should be blocked or restricted if not required.
Implement Detection/Monitoring Capabilities	Monitor for abnormal presence of these or other utilities that enable proxy execution that are typically used for development, debugging, and reverse engineering on a system that is not used for these purposes may be suspicious. Use process monitoring to monitor the execution and arguments of from developer utilities that may be abused. Compare recent invocations of those binaries with prior history of known good arguments and executed binaries to determine anomalous and potentially adversarial activity. It is likely that these utilities will be used by software developers or for other software development related tasks, so if it exists and is used outside of that context, then the event may be suspicious. Command arguments used before and after invocation of the utilities may also be useful in determining the origin and purpose of the binary being executed.

4.2.7.8. *Deobfuscate/Decode Files or Information (T1140)*

Technique Information	
Technique ID	T1140
Technique Name	Deobfuscate/Decode Files or Information
Technique Description	Adversaries may use [Obfuscated Files or Information](T1027) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is use of [certutil](S0160) to decode a remote access tool portable executable file that has been hidden inside a certificate file. Another example is using the Windows \`copy /b\` command to reassemble binary fragments into a malicious payload. Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [User Execution](T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Process: Process Creation	39.25%	37.38%
File: File Modification	47.8%	34.23%

Technique Analysis	
Overall Score	**28.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**72/100**
Overall Log Source Coverage	**29.02%**
Overall Log Collection Coverage	**23.87%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Detecting the action of deobfuscating or decoding files or information may be difficult depending on the implementation. If the functionality is contained within malware and uses the Windows API, then attempting to detect malicious behavior before or after the action may yield better results than attempting to perform analysis on loaded libraries or API calls. If scripts are used, then collecting the scripts for analysis may be necessary. Perform process and command-line monitoring to detect potentially malicious behavior related to scripts and system utilities such as [certutil](S0160). Monitor the execution file paths and command-line arguments for common archive file applications and extensions, such as those for Zip and RAR archive tools, and correlate with other suspicious behavior to reduce false positives from normal user and administrator behavior.

4.2.7.9. Indirect Command Execution (T1202)

Technique Information	
Technique ID	T1202
Technique Name	Indirect Command Execution
Technique Description	Adversaries may abuse utilities that allow for command execution to bypass security restrictions that limit the use of command-line interpreters. Various Windows utilities may be used to execute commands, possibly without invoking [cmd](S0106). For example, [Forfiles](S0193), the Program Compatibility Assistant (pcalua.exe), components of the Windows Subsystem for Linux (WSL), as well as other utilities may invoke the execution of programs and commands from a [Command and Scripting Interpreter](T1059), Run window, or via scripts. Adversaries may abuse these features for [Defense Evasion](TA0005), specifically to perform arbitrary execution while subverting detections and/or mitigation controls (such as Group Policy) that limit/prevent the usage of [cmd](S0106) or file extensions more commonly associated with malicious payloads.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**23.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**

Overall Log Source Coverage	**42.61%**
Overall Log Collection Coverage	**41.68%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor and analyze logs from host-based detection mechanisms, such as Sysmon, for events such as process creations that include or are resulting from parameters associated with invoking programs/commands/files and/or spawning child processes/network connections.

4.2.7.10. Rogue Domain Controller (T1207)

Technique Information	
Technique ID	T1207
Technique Name	Rogue Domain Controller
Technique Description	Adversaries may register a rogue Domain Controller to enable manipulation of Active Directory data. DCShadow may be used to create a rogue Domain Controller (DC). DCShadow is a method of manipulating Active Directory (AD) data, including objects and schemas, by registering (or reusing an inactive registration) and simulating the behavior of a DC. Once registered, a rogue DC may be able to inject and replicate changes into AD infrastructure for any domain object, including credentials and keys. Registering a rogue DC involves creating a new server and nTDSDSA objects in the Configuration partition of the AD schema, which requires Administrator privileges (either Domain or local to the DC) or the KRBTGT hash. This technique may bypass system logging and security monitors such as security information and event management (SIEM) products (since actions taken on a rogue DC may not be reported to these sensors). The technique may also be used to alter and delete replication and other associated metadata to obstruct forensic analysis. Adversaries may also utilize this technique to perform [SID-History Injection](T1134.005) and/or manipulate AD objects (such as accounts, access control lists, schemas) to establish backdoors for Persistence.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Active Directory: Active Directory Object Modification	100.0%	100.0%
User Account: User Account Authentication	100.0%	98.28%

Active Directory: Active Directory Object Creation	100.0%	100.0%
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**45.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**23/100**
Overall Log Source Coverage	**91.16%**
Overall Log Collection Coverage	**74.57%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor and analyze network traffic associated with data replication (such as calls to DrsAddEntry, DrsReplicaAdd, and especially GetNCChanges) between DCs as well as to/from non DC hosts. DC replication will naturally take place every 15 minutes but can be triggered by an adversary or by legitimate urgent changes (ex: passwords). Also consider monitoring and alerting on the replication of AD objects (Audit Detailed Directory Service Replication Events 4928 and 4929). Leverage AD directory synchronization (DirSync) to monitor changes to directory state using AD replication cookies. Baseline and periodically analyze the Configuration

	partition of the AD schema and alert on creation of nTDSDSA objects. Investigate usage of Kerberos Service Principal Names (SPNs), especially those associated with services (beginning with “GC/”) by computers not present in the DC organizational unit (OU). The SPN associated with the Directory Replication Service (DRS) Remote Protocol interface (GUID E3514235–4B06–11D1-AB04–00C04FC2DCD2) can be set without logging. A rogue DC must authenticate as a service using these two SPNs for the replication process to successfully complete.

4.2.7.11. *Exploitation for Defense Evasion (T1211)*

Technique Information	
Technique ID	T1211
Technique Name	Exploitation for Defense Evasion
Technique Description	Adversaries may exploit a system or application vulnerability to bypass security features. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Vulnerabilities may exist in defensive security software that can be used to disable or circumvent them. Adversaries may have prior knowledge through reconnaissance that security software exists within an environment or they may perform checks during or shortly after the system is compromised for [Security Software Discovery](T1518.001). The security software will likely be targeted directly for exploitation. There are examples of antivirus software being targeted by persistent threat groups to avoid detection.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Driver: Driver Load	45.98%	45.98%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Script: Script Execution	0.0%	0.0%
Module: Module Load	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**37.84%**
Overall Log Collection Coverage	**37.25%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of exploitation. Risks of additional exploits and weaknesses in these systems may still exist.
Exploit Protection	Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior.
Threat Intelligence Program	Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization.
Update Software	Update software regularly by employing patch management for internal enterprise endpoints and servers.

Implement Detection/Monitoring Capabilities	Exploitation for defense evasion may happen shortly after the system has been compromised to prevent detection during later actions for for additional tools that may be brought in and used. Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the system that might indicate successful compromise, such as abnormal behavior of processes. This could include suspicious files written to disk, evidence of [Process Injection](T1055) for attempts to hide execution or evidence of Discovery.

4.2.7.12. System Script Proxy Execution (T1216)

Technique Information	
Technique ID	T1216
Technique Name	System Script Proxy Execution
Technique Description	Adversaries may use trusted scripts, often signed with certificates, to proxy the execution of malicious files. Several Microsoft signed scripts that have been downloaded from Microsoft or are default on Windows installations can be used to proxy execution of other files. This behavior may be abused by adversaries to execute malicious files that could bypass application control and signature validation on systems.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**15.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**35/100**
Overall Log Source Coverage	**28.41%**
Overall Log Collection Coverage	**27.79%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Execution Prevention	Certain signed scripts that can be used to execute other programs may not be necessary within a given environment. Use application control configured to block execution of these scripts if they are not required for a given system or network to prevent potential misuse by adversaries.
Implement Detection/Monitoring Capabilities	Monitor script processes, such as \`cscript\`, and command-line parameters for scripts like PubPrn.vbs that may be used to proxy execution of malicious files.

4.2.7.13. *System Binary Proxy Execution (T1218)*

Technique Information	
Technique ID	T1218
Technique Name	System Binary Proxy Execution
Technique Description	Adversaries may bypass process and/or signature-based defenses by proxying execution of malicious content with signed, or otherwise trusted, binaries. Binaries used in this technique are often Microsoft-signed files, indicating that they have been either downloaded from Microsoft or are already native in the operating system. Binaries signed with trusted digital certificates can typically execute on Windows systems protected by digital signature validation. Several Microsoft signed binaries that are default on Windows installations can be used to proxy execution of other files or commands. Similarly, on Linux systems adversaries may abuse trusted binaries such as \`split\` to proxy execution of malicious commands.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
Process: OS API Execution	49.89%	48.17%
File: File Metadata	67.37%	55.82%
Network Traffic: Network Connection Creation	45.45%	45.45%
Module: Module Load	45.98%	45.98%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**72.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**12/100**
Overall Log Source Coverage	**51.61%**
Overall Log Collection Coverage	**47.78%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Disable or Remove Feature or Program	Many native binaries may not be necessary within a given environment.
Execution Prevention	Consider using application control to prevent execution of binaries that are susceptible to abuse and not required for a given system or network.
Exploit Protection	Microsoft's Enhanced Mitigation Experience Toolkit (EMET) Attack Surface Reduction (ASR) feature can be used to block methods of using using trusted binaries to bypass application control.
Privileged Account Management	Restrict execution of particularly vulnerable binaries to privileged accounts or groups that need to use it to lessen the opportunities for malicious usage.
Implement Detection/Monitoring Capabilities	Monitor processes and command-line parameters for signed binaries that may be used to proxy execution of malicious files. Compare recent invocations of signed binaries that may be used to proxy execution with prior history of known good arguments and loaded files to determine anomalous and potentially adversarial activity.

	Legitimate programs used in suspicious ways, like msiexec.exe downloading an MSI file from the Internet, may be indicative of an intrusion. Correlate activity with other suspicious behavior to reduce false positives that may be due to normal benign use by users and administrators. Monitor for file activity (creations, downloads, modifications, etc.), especially for file types that are not typical within an environment and may be indicative of adversary activity.

4.2.7.14. *XSL Script Processing (T1220)*

Technique Information	
Technique ID	T1220
Technique Name	XSL Script Processing
Technique Description	Adversaries may bypass application control and obscure execution of code by embedding scripts inside XSL files. Extensible Stylesheet Language (XSL) files are commonly used to describe the processing and rendering of data within XML files. To support complex operations, the XSL standard includes support for embedded scripting in various languages. Adversaries may abuse this functionality to execute arbitrary files while potentially bypassing application control. Similar to [Trusted Developer Utilities Proxy Execution](T1127), the Microsoft common line transformation utility binary (msxsl.exe) can be installed and used to execute malicious JavaScript embedded within local or remote (URL referenced) XSL files. Since msxsl.exe is not installed by default, an adversary will likely need to package it with dropped files. Msxsl.exe takes two main arguments, an XML source file and an XSL stylesheet. Since the XSL file is valid XML, the adversary may call the same XSL file twice. When using msxsl.exe adversaries may also give the XML/XSL files an arbitrary file extension. Command-line examples: * `msxsl.exe customers[.]xml script[.]xsl` * `msxsl.exe script[.]xsl script[.]xsl` * `msxsl.exe script[.]jpeg script[.]jpeg` Another variation of this technique, dubbed “Squiblytwo”, involves using [Windows Management Instrumentation](T1047) to invoke JScript or VBScript within an XSL file. This technique can also execute local/remote scripts and, similar to its [Regsvr32](T1218.010)/ "Squiblydoo" counterpart, leverages a trusted, built-in Windows tool. Adversaries may abuse any alias in [Windows Management Instrumentation](T1047) provided they utilize the /FORMAT switch. Command-line examples:

	* Local File: `wmic process list /FORMAT:evil[.]xsl` * Remote File: `wmic os get /FORMAT:"https[:]//example[.]com/evil[.]xsl"`

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Process: Process Creation	39.25%	37.38%
Module: Module Load	45.98%	45.98%

Technique Analysis	
Overall Score	**23.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**42.61%**
Overall Log Collection Coverage	**41.68%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Execution Prevention	If msxsl.exe is unnecessary, then block its execution to prevent abuse by adversaries.
Implement Detection/Monitoring Capabilities	Use process monitoring to monitor the execution and arguments of msxsl.exe and wmic.exe. Compare recent invocations of these utilities with prior history of known good arguments and loaded files to

	determine anomalous and potentially adversarial activity (ex: URL command line arguments, creation of external network connections, loading of DLLs associated with scripting). Command arguments used before and after the script invocation may also be useful in determining the origin and purpose of the payload being loaded. The presence of msxsl.exe or other utilities that enable proxy execution that are typically used for development, debugging, and reverse engineering on a system that is not used for these purposes may be suspicious.

4.2.7.15. *Template Injection (T1221)*

Technique Information	
Technique ID	T1221
Technique Name	Template Injection
Technique Description	Adversaries may create or modify references in user document templates to conceal malicious code or force authentication attempts. For example, Microsoft's Office Open XML (OOXML) specification defines an XML-based format for Office documents (.docx, xlsx, .pptx) to replace older binary formats (.doc, .xls, .ppt). OOXML files are packed together ZIP archives compromised of various XML files, referred to as parts, containing properties that collectively define how a document is rendered. Properties within parts may reference shared public resources accessed via online URLs. For example, template properties may reference a file, serving as a pre-formatted document blueprint, that is fetched when the document is loaded. Adversaries may abuse these templates to initially conceal malicious code to be executed via user documents. Template references injected into a document may enable malicious payloads to be fetched and executed when the document is loaded. These documents can be delivered via other techniques such as [Phishing](T1566) and/or [Taint Shared Content](T1080) and may evade static detections since no typical indicators (VBA macro, script, etc.) are present until after the malicious payload is fetched. Examples have been seen in the wild where template injection was used to load malicious code containing an exploit. Adversaries may also modify the `*\template` control word within an .rtf file to similarly conceal then download malicious code. This legitimate control word value is intended to be a file destination of a template file resource that is retrieved and loaded when an .rtf file is opened. However, adversaries may alter the bytes of an existing .rtf file to insert a template control word field to include a URL resource of a malicious payload. This technique may also enable [Forced Authentication](T1187)

	by injecting a SMB/HTTPS (or other credential prompting) URL and triggering an authentication attempt.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**49.79%**
Overall Log Collection Coverage	**27.61%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Antivirus/Antimalware	Network/Host intrusion prevention systems, antivirus, and detonation chambers can be employed to prevent documents from fetching and/or executing malicious payloads.

Disable or Remove Feature or Program	Consider disabling Microsoft Office macros/active content to prevent the execution of malicious payloads in documents
Network Intrusion Prevention	Network/Host intrusion prevention systems, antivirus, and detonation chambers can be employed to prevent documents from fetching and/or executing malicious payloads.
User Training	Train users to identify social engineering techniques and spearphishing emails that could be used to deliver malicious documents.
Implement Detection/Monitoring Capabilities	Analyze process behavior to determine if user document applications (such as Office) are performing actions, such as opening network connections, reading files, spawning abnormal child processes (ex: [PowerShell](T1059.001)), or other suspicious actions that could relate to post-compromise behavior. Monitor .rtf files for strings indicating the \`*\template\` control word has been modified to retrieve a URL resource, such as \`*\template http\` or \`*\template \u-\`.

4.2.7.16. File and Directory Permissions Modification (T1222)

Technique Information	
Technique ID	T1222
Technique Name	File and Directory Permissions Modification
Technique Description	Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files. File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.). Modifications may include changing specific access rights, which may require taking ownership of a file or directory and/or elevated permissions depending on the file or directory's existing permissions. This may enable malicious activity such as modifying, replacing, or deleting specific files or directories. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Accessibility Features](T1546.008), [Boot or Logon Initialization Scripts](T1037), [Unix Shell Configuration Modification](T1546.004), or tainting/hijacking other instrumental binary/configuration files via [Hijack Execution Flow](T1574).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Active Directory: Active Directory Object Modification	100.0%	100.0%
Command: Command Execution	45.98%	45.98%
File: File Metadata	67.37%	55.82%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**79.0%**
Status	**Good maturity**
Sector Specific Priority	**9/100**
Overall Log Source Coverage	**63.15%**
Overall Log Collection Coverage	**59.8%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Privileged Account Management	Ensure critical system files as well as those known to be abused by adversaries have restrictive permissions and are owned by an appropriately privileged account, especially if access is not required by users nor will inhibit system functionality.
Restrict File and Directory Permissions	Applying more restrictive permissions to files and directories could prevent adversaries from modifying their access control lists. Additionally, ensure that user settings regarding local and remote symbolic links are properly set or disabled where unneeded.
Implement Detection/Monitoring Capabilities	Monitor and investigate attempts to modify ACLs and file/directory ownership. Many of the commands used to modify ACLs and file/directory ownership are built-in system utilities and may generate a high false positive alert rate, so compare against baseline knowledge for how systems are typically used and correlate modification events with other indications of malicious activity where

	possible. Consider enabling file/directory permission change auditing on folders containing key binary/configuration files. For example, Windows Security Log events (Event ID 4670) are created when DACLs are modified.

4.2.7.17. Execution Guardrails (T1480)

Technique Information	
Technique ID	T1480
Technique Name	Execution Guardrails
Technique Description	Adversaries may use execution guardrails to constrain execution or actions based on adversary supplied and environment specific conditions that are expected to be present on the target. Guardrails ensure that a payload only executes against an intended target and reduces collateral damage from an adversary's campaign. Values an adversary can provide about a target system or environment to use as guardrails may include specific network share names, attached physical devices, files, joined Active Directory (AD) domains, and local/external IP addresses. Guardrails can be used to prevent exposure of capabilities in environments that are not intended to be compromised or operated within. This use of guardrails is distinct from typical [Virtualization/Sandbox Evasion](T1497). While use of [Virtualization/Sandbox Evasion](T1497) may involve checking for known sandbox values and continuing with execution only if there is no match, the use of guardrails will involve checking for an expected target-specific value and only continuing with execution if there is such a match.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**23.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**42.61%**
Overall Log Collection Coverage	**41.68%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Do Not Mitigate	likely should not be mitigated with preventative controls because it may protect unintended targets from being compromised. If targeted, efforts should be focused on preventing adversary tools from running earlier in the chain of activity and on identifying subsequent malicious behavior if compromised.
Implement Detection/Monitoring Capabilities	Detecting the use of guardrails may be difficult depending on the implementation. Monitoring for suspicious processes being spawned that gather a variety of system information or perform other forms of [Discovery](TA0007), especially in a short period of time, may aid in detection.

4.2.7.18. *Unused/Unsupported Cloud Regions (T1535)*

Technique Information	
Technique ID	T1535
Technique Name	Unused/Unsupported Cloud Regions
Technique Description	Adversaries may create cloud instances in unused geographic service regions in order to evade detection. Access is usually obtained through compromising accounts used to manage cloud infrastructure. Cloud service providers often provide infrastructure throughout the world in order to improve performance, provide redundancy, and allow customers to meet compliance requirements. Oftentimes, a customer will only use a subset of the available regions and may not actively monitor other regions. If an adversary creates resources in an unused region, they may be able to operate undetected. A variation on this behavior takes advantage of differences in functionality across cloud regions. An adversary could utilize regions which do not support advanced detection services in order to avoid detection of their activity. An example of adversary use of unused AWS regions is to mine cryptocurrency through [Resource Hijacking](T1496), which can cost organizations substantial amounts of money over time depending on the processing power used.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Instance: Instance Creation	100.0%	100.0%

Technique Analysis	
Overall Score	**55.0%**

Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**100.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Software Configuration	Cloud service providers may allow customers to deactivate unused regions.
Implement Detection/Monitoring Capabilities	Monitor system logs to review activities occurring across all cloud environments and regions. Configure alerting to notify of activity in normally unused regions or if the number of instances active in a region goes above a certain threshold.

4.2.7.19. *Subvert Trust Controls (T1553)*

Technique Information	
Technique ID	T1553
Technique Name	Subvert Trust Controls
Technique Description	Adversaries may undermine security controls that will either warn users of untrusted activity or prevent execution of untrusted programs. Operating systems and security products may contain mechanisms to identify programs or websites as possessing some level of trust. Examples of such features would include a program being allowed to run because it is signed by a valid code signing certificate, a program prompting the user with a warning because it has an attribute set from being downloaded from the Internet, or getting an indication that you are about to connect to an untrusted site. Adversaries may attempt to subvert these trust mechanisms. The method adversaries use will depend on the specific mechanism they seek to subvert. Adversaries may conduct [File and Directory Permissions Modification](T1222) or [Modify Registry](T1112) in support of subverting these controls. Adversaries may also create or steal code signing certificates to acquire trust on target systems.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
File: File Metadata	67.37%	55.82%
File: File Modification	47.8%	34.23%
Module: Module Load	45.98%	45.98%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%

Windows Registry: Windows Registry Key Creation	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**27.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**73/100**
Overall Log Source Coverage	**51.41%**
Overall Log Collection Coverage	**46.1%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Execution Prevention	System settings can prevent applications from running that haven't been downloaded through the Apple Store (or other legitimate repositories) which can help mitigate some of these issues. Also enable application control solutions such as AppLocker and/or Device Guard to block the loading of malicious content.
Operating System Configuration	Windows Group Policy can be used to manage root certificates and the
Restrict Registry Permissions	Ensure proper permissions are set for Registry hives to prevent users from modifying keys related to SIP and trust provider components. Components may still be able to be hijacked to suitable functions already present on disk if malicious modifications to Registry keys are not prevented.

Software Configuration	HTTP Public Key Pinning (HPKP) is one method to mitigate potential
Implement Detection/Monitoring Capabilities	Collect and analyze signing certificate metadata on software that executes within the environment to look for unusual certificate characteristics and outliers. Periodically baseline registered SIPs and trust providers (Registry entries and files on disk), specifically looking for new, modified, or non-Microsoft entries. A system's root certificates are unlikely to change frequently. Monitor new certificates installed on a system that could be due to malicious activity. Analyze Autoruns data for oddities and anomalies, specifically malicious files attempting persistent execution by hiding within auto-starting locations. Autoruns will hide entries signed by Microsoft or Windows by default, so ensure "Hide Microsoft Entries" and "Hide Windows Entries" are both deselected. Monitor and investigate attempts to modify extended file attributes with utilities such as \`xattr\`. Built-in system utilities may generate high false positive alerts, so compare against baseline knowledge for how systems are typically used and correlate modification events with other indications of malicious activity where possible.

4.2.7.20. *Impair Defenses (T1562)*

Technique Information	
Technique ID	T1562
Technique Name	Impair Defenses
Technique Description	Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Windows Registry: Windows Registry Key Creation	45.98%	45.98%
Process: Process Metadata	39.25%	37.38%
Firewall: Firewall Rule Modification	100.0%	0.0%
Script: Script Execution	0.0%	0.0%
Sensor Health: Host Status	36.78%	31.06%
Firewall: Firewall Disable	100.0%	0.0%
Service: Service Metadata	0.23%	0.23%

Application Log: Application Log Content	66.67%	33.33%
Process: Process Termination	39.25%	37.38%
Cloud Service: Cloud Service Modification	100.0%	0.0%
Windows Registry: Windows Registry Key Deletion	45.98%	45.98%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Cloud Service: Cloud Service Disable	100.0%	0.0%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**39.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**56/100**
Overall Log Source Coverage	**53.69%**
Overall Log Collection Coverage	**24.05%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Execution Prevention	Use application control where appropriate, especially regarding the execution of tools outside of the organization's security policies (such as rootkit removal tools) that have been abused to

	impair system defenses. Ensure that only approved security applications are used and running on enterprise systems.
Restrict File and Directory Permissions	Ensure proper process and file permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
Restrict Registry Permissions	Ensure proper Registry permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
User Account Management	Ensure proper user permissions are in place to prevent adversaries from disabling or interfering with security/logging services.
Implement Detection/Monitoring Capabilities	Monitor processes and command-line arguments to see if security tools or logging services are killed or stop running. Monitor Registry edits for modifications to services and startup programs that correspond to security tools. Lack of log events may be suspicious. Monitor environment variables and APIs that can be leveraged to disable security measures.

4.2.7.21. Hide Artifacts (T1564)

Technique Information	
Technique ID	T1564
Technique Name	Hide Artifacts
Technique Description	Adversaries may attempt to hide artifacts associated with their behaviors to evade detection. Operating systems may have features to hide various artifacts, such as important system files and administrative task execution, to avoid disrupting user work environments and prevent users from changing files or features on the system. Adversaries may abuse these features to hide artifacts such as files, directories, user accounts, or other system activity to evade detection. Adversaries may also attempt to hide artifacts associated with malicious behavior by creating computing regions that are isolated from common security instrumentation, such as through the use of virtualization technology.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
User Account: User Account Creation	100.0%	98.28%
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
File: File Metadata	67.37%	55.82%
Process: OS API Execution	49.89%	48.17%
File: File Modification	47.8%	34.23%
Script: Script Execution	0.0%	0.0%
Process: Process Creation	39.25%	37.38%
Service: Service Creation	0.23%	0.23%

User Account: User Account Metadata	100.0%	98.28%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Firmware: Firmware Modification	91.95%	91.95%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**69.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**13/100**
Overall Log Source Coverage	**56.01%**
Overall Log Collection Coverage	**49.78%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor files, processes, and command-line arguments for actions indicative of hidden artifacts. Monitor event and authentication logs for records of hidden artifacts being used. Monitor the file system and shell commands for hidden attribute usage.

4.2.7.22. Network Boundary Bridging (T1599)

Technique Information	
Technique ID	T1599
Technique Name	Network Boundary Bridging
Technique Description	Adversaries may bridge network boundaries by compromising perimeter network devices or internal devices responsible for network segmentation. Breaching these devices may enable an adversary to bypass restrictions on traffic routing that otherwise separate trusted and untrusted networks. Devices such as routers and firewalls can be used to create boundaries between trusted and untrusted networks. They achieve this by restricting traffic types to enforce organizational policy in an attempt to reduce the risk inherent in such connections. Restriction of traffic can be achieved by prohibiting IP addresses, layer 4 protocol ports, or through deep packet inspection to identify applications. To participate with the rest of the network, these devices can be directly addressable or transparent, but their mode of operation has no bearing on how the adversary can bypass them when compromised. When an adversary takes control of such a boundary device, they can bypass its policy enforcement to pass normally prohibited traffic across the trust boundary between the two separated networks without hinderance. By achieving sufficient rights on the device, an adversary can reconfigure the device to allow the traffic they want, allowing them to then further achieve goals such as command and control via [Multi-hop Proxy](T1090.003) or exfiltration of data via [Traffic Duplication](T1020.001). Adversaries may also target internal devices responsible for network segmentation and abuse these in conjunction with [Internal Proxy](T1090.001) to achieve the same goals. In the cases where a border device separates two separate organizations, the adversary can also facilitate lateral movement into new victim environments.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**39.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**25/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Credential Access Protection	Some embedded network devices are capable of storing passwords for local accounts in either plain-text or encrypted formats. Ensure that, where available, local passwords are always encrypted, per vendor recommendations.
Filter Network Traffic	Upon identifying a compromised network device being used to bridge a network boundary, block the malicious packets using an unaffected network device in path, such as a firewall or a router that has not been compromised. Continue to monitor for additional activity and to ensure that the blocks are indeed effective.

Multi-factor Authentication	Use multi-factor authentication for user and privileged accounts. Most embedded network devices support TACACS+ and/or RADIUS. Follow vendor prescribed best practices for hardening access control.
Password Policies	Refer to NIST guidelines when creating password policies.
Privileged Account Management	Restrict administrator accounts to as few individuals as possible, following least privilege principles. Prevent credential overlap across systems of administrator and privileged accounts, particularly between network and non-network platforms, such as servers or endpoints.
Implement Detection/Monitoring Capabilities	Consider monitoring network traffic on both interfaces of border network devices with out-of-band packet capture or network flow data, using a different device than the one in question. Look for traffic that should be prohibited by the intended network traffic policy enforcement for the border network device. Monitor the border network device's configuration to validate that the policy enforcement sections are what was intended. Look for rules that are less restrictive, or that allow specific traffic types that were not previously authorized.

4.2.7.23. *Weaken Encryption (T1600)*

Technique Information	
Technique ID	T1600
Technique Name	Weaken Encryption
Technique Description	Adversaries may compromise a network device's encryption capability in order to bypass encryption that would otherwise protect data communications. Encryption can be used to protect transmitted network traffic to maintain its confidentiality (protect against unauthorized disclosure) and integrity (protect against unauthorized changes). Encryption ciphers are used to convert a plaintext message to ciphertext and can be computationally intensive to decipher without the associated decryption key. Typically, longer keys increase the cost of cryptanalysis, or decryption without the key. Adversaries can compromise and manipulate devices that perform encryption of network traffic. For example, through behaviors such as [Modify System Image](T1601), [Reduce Key Space](T1600.001), and [Disable Crypto Hardware](T1600.002), an adversary can negatively effect and/or eliminate a device's ability to securely encrypt network traffic. This poses a greater risk of unauthorized disclosure and may help facilitate data manipulation, Credential Access, or Collection efforts.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Modification	47.8%	34.23%

Technique Analysis	
Overall Score	**22.0%**
Status	**Needs immediate remediation**

Sector Specific Priority	**32/100**
Overall Log Source Coverage	**47.8%**
Overall Log Collection Coverage	**34.23%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	There is no documented method for defenders to directly identify behaviors that weaken encryption. Detection efforts may be focused on closely related adversary behaviors, such as [Modify System Image](T1601). Some detection methods require vendor support to aid in investigation.

4.2.7.24. Modify System Image (T1601)

Technique Information	
Technique ID	T1601
Technique Name	Modify System Image
Technique Description	Adversaries may make changes to the operating system of embedded network devices to weaken defenses and provide new capabilities for themselves. On such devices, the operating systems are typically monolithic and most of the device functionality and capabilities are contained within a single file. To change the operating system, the adversary typically only needs to affect this one file, replacing or modifying it. This can either be done live in memory during system runtime for immediate effect, or in storage to implement the change on the next boot of the network device.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Modification	47.8%	34.23%

Technique Analysis	
Overall Score	**22.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**47.8%**
Overall Log Collection Coverage	**34.23%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Boot Integrity	Some vendors of embedded network devices provide cryptographic signing to ensure the integrity of operating system images at boot time. Implement where available, following vendor guidelines.
Code Signing	Many vendors provide digitally signed operating system images to validate the integrity of the software used on their platform. Make use of this feature where possible in order to prevent and/or detect attempts by adversaries to compromise the system image.
Credential Access Protection	Some embedded network devices are capable of storing passwords for local accounts in either plain-text or encrypted formats. Ensure that, where available, local passwords are always encrypted, per vendor recommendations.
Multi-factor Authentication	Use multi-factor authentication for user and privileged accounts. Most embedded network devices support TACACS+ and/or RADIUS. Follow vendor prescribed best practices for hardening access control.
Password Policies	Refer to NIST guidelines when creating password policies.
Privileged Account Management	Restrict administrator accounts to as few individuals as possible, following least privilege principles. Prevent credential overlap across systems of administrator and privileged accounts, particularly between network and non-network platforms, such as servers or endpoints.
Implement Detection/Monitoring Capabilities	Most embedded network devices provide a command to print the version of the currently running operating system. Use this command to query the operating system for its version number

	and compare it to what is expected for the device in question. Because this method may be used in conjunction with [Patch System Image](T1601.001), it may be appropriate to also verify the integrity of the vendor provided operating system image file. Compare the checksum of the operating system file with the checksum of a known good copy from a trusted source. Some embedded network device platforms may have the capability to calculate the checksum of the file, while others may not. Even for those platforms that have the capability, it is recommended to download a copy of the file to a trusted computer to calculate the checksum with software that is not compromised. Many vendors of embedded network devices can provide advanced debugging support that will allow them to work with device owners to validate the integrity of the operating system running in memory. If a compromise of the operating system is suspected, contact the vendor technical support and seek such services for a more thorough inspection of the current running system.

4.2.7.25. *Reflective Code Loading (T1620)*

Technique Information	
Technique ID	T1620
Technique Name	Reflective Code Loading
Technique Description	Adversaries may reflectively load code into a process in order to conceal the execution of malicious payloads. Reflective loading involves allocating then executing payloads directly within the memory of the process, vice creating a thread or process backed by a file path on disk. Reflectively loaded payloads may be compiled binaries, anonymous files (only present in RAM), or just snubs of fileless executable code (ex: position-independent shellcode). Reflective code injection is very similar to [Process Injection](T1055) except that the "injection" loads code into the processes' own memory instead of that of a separate process. Reflective loading may evade process-based detections since the execution of the arbitrary code may be masked within a legitimate or otherwise benign process. Reflectively loading payloads directly into memory may also avoid creating files or other artifacts on disk, while also enabling malware to keep these payloads encrypted (or otherwise obfuscated) until execution.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Process: OS API Execution	49.89%	48.17%
Module: Module Load	45.98%	45.98%

Technique Analysis	
Overall Score	**17.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**35/100**
Overall Log Source Coverage	**31.95%**
Overall Log Collection Coverage	**31.38%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor for code artifacts associated with reflectively loading code, such as the abuse of .NET functions such as `Assembly.Load()` and [Native API](T1106) functions such as `CreateThread()`, `memfd_create()`, `execve()`, and/or `execveat()`. Monitor for artifacts of abnormal process execution. For example, a common signature related to reflective code loading on Windows is mechanisms related to the .NET Common Language Runtime (CLR) -- such as mscor.dll, mscoree.dll, and clr.dll -- loading into abnormal processes (such as notepad.exe). Similarly, AMSI / ETW traces can be used to identify signs of arbitrary code execution from within the memory of potentially compromised processes. Analyze process behavior to determine if a process is performing actions it usually does not, such as opening network connections, reading files, or other suspicious actions that could relate to post-compromise behavior.

4.2.8. Credential Access

4.2.8.1. OS Credential Dumping (T1003)

Technique Information	
Technique ID	T1003
Technique Name	OS Credential Dumping
Technique Description	Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform [Lateral Movement](TA0008) and access restricted information. Several of the tools mentioned in associated sub-techniques may be used by both adversaries and professional security testers. Additional custom tools likely exist as well.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Active Directory: Active Directory Object Access	100.0%	100.0%
Process: OS API Execution	49.89%	48.17%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Traffic: Network Traffic Content	64.66%	0.0%
Windows Registry: Windows Registry Key Access	100.0%	0.0%
File: File Access	37.14%	29.99%
Process: Process Access	45.98%	45.98%

Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**47.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**22/100**
Overall Log Source Coverage	**60.84%**
Overall Log Collection Coverage	**34.17%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Active Directory Configuration	Manage the access control list for "Replicating Directory Changes" and other permissions associated with domain controller replication.
Behavior Prevention on Endpoint	On Windows 10, enable Attack Surface Reduction (ASR) rules to secure LSASS and prevent credential stealing.
Credential Access Protection	With Windows 10, Microsoft implemented new protections called Credential Guard to protect the LSA secrets that can be used to obtain credentials through forms of credential dumping. It is not configured by default and has hardware and firmware system requirements.
Encrypt Sensitive Information	Ensure Domain Controller backups are properly secured.
Operating System Configuration	Consider disabling or restricting NTLM.

Password Policies	Ensure that local administrator accounts have complex, unique passwords across all systems on the network.
Privileged Account Management	Windows:Do not put user or admin domain accounts in the local administrator groups across systems unless they are tightly controlled, as this is often equivalent to having a local administrator account with the same password on all systems. Follow best practices for design and administration of an enterprise network to limit privileged account use across administrative tiers.
Privileged Process Integrity	On Windows 8.1 and Windows Server 2012 R2, enable Protected Process Light for LSA.
User Training	Limit credential overlap across accounts and systems by training users and administrators not to use the same password for multiple accounts.
Implement Detection/Monitoring Capabilities	On Windows devices you can monitor for unexpected processes interacting with lsass.exe. Common credential dumpers such as [Mimikatz](S0002) access the LSA Subsystem Service (LSASS) process by opening the process, locating the LSA secrets key, and decrypting the sections in memory where credential details are stored. Credential dumpers may also use methods for reflective [Process Injection](T1055) to reduce potential indicators of malicious activity. Hash dumpers open the Security Accounts Manager (SAM) on the local file system (%SystemRoot%/system32/config/SAM) or create a dump of the Registry SAM key to access stored account password hashes. Some hash dumpers will open the local file system as a device and parse to the SAM table to avoid file access defenses. Others will make an in-memory copy of the SAM table before reading hashes. Detection of compromised [Valid Accounts](T1078) in-use by adversaries may help as well. On Windows 8.1 and Windows Server 2012 R2,

	monitor Windows Logs for LSASS.exe creation to verify that LSASS started as a protected process. Monitor processes and command-line arguments for program execution that may be indicative of credential dumping. Remote access tools may contain built-in features or incorporate existing tools like [Mimikatz](S0002). [PowerShell](T1059.001) scripts also exist that contain credential dumping functionality, such as PowerSploit's Invoke-Mimikatz module, which may require additional logging features to be configured in the operating system to collect necessary information for analysis. Monitor domain controller logs for replication requests and other unscheduled activity possibly associated with DCSync. Note: Domain controllers may not log replication requests originating from the default domain controller account. . Also monitor for network protocols and other replication requests from IPs not associated with known domain controllers. On Linux devices, in order to obtain the passwords and hashes stored in memory, processes must open a maps file in the /proc filesystem for the process being analyzed. This file is stored under the path `/proc/<pid>/maps`, where the `<pid>` directory is the unique pid of the program being interrogated for such authentication data. The AuditD monitoring tool, which ships stock in many Linux distributions, can be used to watch for hostile processes opening this file in the proc file system, alerting on the pid, process name, and arguments of such programs.

4.2.8.2. *Brute Force (T1110)*

Technique Information	
Technique ID	T1110
Technique Name	Brute Force
Technique Description	Adversaries may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained. Without knowledge of the password for an account or set of accounts, an adversary may systematically guess the password using a repetitive or iterative mechanism. Brute forcing passwords can take place via interaction with a service that will check the validity of those credentials or offline against previously acquired credential data, such as password hashes. Brute forcing credentials may take place at various points during a breach. For example, adversaries may attempt to brute force access to [Valid Accounts](T1078) within a victim environment leveraging knowledge gathered from other post-compromise behaviors such as [OS Credential Dumping](T1003), [Account Discovery](T1087), or [Password Policy Discovery](T1201). Adversaries may also combine brute forcing activity with behaviors such as [External Remote Services](T1133) as part of Initial Access.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
User Account: User Account Authentication	100.0%	98.28%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis

Overall Score	**80.0%**
Status	**Good maturity**
Sector Specific Priority	**8/100**
Overall Log Source Coverage	**70.88%**
Overall Log Collection Coverage	**59.2%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Account Use Policies	Set account lockout policies after a certain number of failed login attempts to prevent passwords from being guessed. Too strict a policy may create a denial of service condition and render environments un-usable, with all accounts used in the brute force being locked-out.
Multi-factor Authentication	Use multi-factor authentication. Where possible, also enable multi-factor authentication on externally facing services.
Password Policies	Refer to NIST guidelines when creating password policies.
User Account Management	Proactively reset accounts that are known to be part of breached credentials either immediately, or after detecting bruteforce attempts.
Implement Detection/Monitoring Capabilities	Monitor authentication logs for system and application login failures of [Valid Accounts](T1078). If authentication failures are high, then there may be a brute force attempt to gain access to a system using legitimate credentials. Also monitor for many failed authentication attempts across various accounts that may result from password spraying attempts. It is difficult to detect when hashes are

	cracked, since this is generally done outside the scope of the target network.

4.2.8.3. Multi-Factor Authentication Interception (T1111)

Technique Information	
Technique ID	T1111
Technique Name	Multi-Factor Authentication Interception
Technique Description	Adversaries may target multi-factor authentication (MFA) mechanisms, (I.e., smart cards, token generators, etc.) to gain access to credentials that can be used to access systems, services, and network resources. Use of MFA is recommended and provides a higher level of security than user names and passwords alone, but organizations should be aware of techniques that could be used to intercept and bypass these security mechanisms. If a smart card is used for multi-factor authentication, then a keylogger will need to be used to obtain the password associated with a smart card during normal use. With both an inserted card and access to the smart card password, an adversary can connect to a network resource using the infected system to proxy the authentication with the inserted hardware token. Adversaries may also employ a keylogger to similarly target other hardware tokens, such as RSA SecurID. Capturing token input (including a user's personal identification code) may provide temporary access (i.e. replay the one-time passcode until the next value rollover) as well as possibly enabling adversaries to reliably predict future authentication values (given access to both the algorithm and any seed values used to generate appended temporary codes). Other methods of MFA may be intercepted and used by an adversary to authenticate. It is common for one-time codes to be sent via out-of-band communications (email, SMS). If the device and/or service is not secured, then it may be vulnerable to interception. Although primarily focused on by cyber criminals, these authentication mechanisms have been targeted by advanced actors.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Driver: Driver Load	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**47.28%**
Overall Log Collection Coverage	**46.71%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
User Training	Remove smart cards when not in use.
Implement Detection/Monitoring Capabilities	Detecting use of proxied smart card connections by an adversary may be difficult because it requires the token to be inserted into a system; thus it is more likely to be in use by a legitimate user and blend in with other network behavior. Similar to [Input Capture](T1056), keylogging activity can take various forms but can may be detected via installation of a driver, setting a hook,

	or usage of particular API calls associated with polling to intercept keystrokes.

4.2.8.4. *Forced Authentication (T1187)*

Technique Information	
Technique ID	T1187
Technique Name	Forced Authentication
Technique Description	Adversaries may gather credential material by invoking or forcing a user to automatically provide authentication information through a mechanism in which they can intercept. The Server Message Block (SMB) protocol is commonly used in Windows networks for authentication and communication between systems for access to resources and file sharing. When a Windows system attempts to connect to an SMB resource it will automatically attempt to authenticate and send credential information for the current user to the remote system. This behavior is typical in enterprise environments so that users do not need to enter credentials to access network resources. Web Distributed Authoring and Versioning (WebDAV) is also typically used by Windows systems as a backup protocol when SMB is blocked or fails. WebDAV is an extension of HTTP and will typically operate over TCP ports 80 and 443. Adversaries may take advantage of this behavior to gain access to user account hashes through forced SMB/WebDAV authentication. An adversary can send an attachment to a user through spearphishing that contains a resource link to an external server controlled by the adversary (i.e. [Template Injection](T1221)), or place a specially crafted file on navigation path for privileged accounts (e.g. .SCF file placed on desktop) or on a publicly accessible share to be accessed by victim(s). When the user's system accesses the untrusted resource it will attempt authentication and send information, including the user's hashed credentials, over SMB to the adversary controlled server. With access to the credential hash, an adversary can perform off-line [Brute Force](T1110) cracking to gain access to plaintext credentials. There are several different ways this can occur. Some specifics from in-the-wild use include: * A spearphishing attachment containing a document with a

	resource that is automatically loaded when the document is opened (i.e. [Template Injection](T1221)). The document can include, for example, a request similar to `file[:]//[remote address]/Normal.dotm` to trigger the SMB request. * A modified .LNK or .SCF file with the icon filename pointing to an external reference such as `\\[remote address]\pic.png` that will force the system to load the resource when the icon is rendered to repeatedly gather credentials.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Creation	72.99%	57.47%
Network Traffic: Network Traffic Flow	64.66%	0.0%
File: File Modification	47.8%	34.23%
File: File Access	37.14%	29.99%
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**22.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**57.45%**
Overall Log Collection Coverage	**24.34%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Filter Network Traffic	Block SMB traffic from exiting an enterprise network with egress filtering or by blocking TCP ports 139, 445 and UDP port 137. Filter or block WebDAV protocol traffic from exiting the network. If access to external resources over SMB and WebDAV is necessary, then traffic should be tightly limited with allowlisting.
Password Policies	Use strong passwords to increase the difficulty of credential hashes from being cracked if they are obtained.
Implement Detection/Monitoring Capabilities	Monitor for SMB traffic on TCP ports 139, 445 and UDP port 137 and WebDAV traffic attempting to exit the network to unknown external systems. If attempts are detected, then investigate endpoint data sources to find the root cause. For internal traffic, monitor the workstation-to-workstation unusual (vs. baseline) SMB traffic. For many networks there should not be any, but it depends on how systems on the network are configured and where resources are located. Monitor creation and modification of .LNK, .SCF, or any other files on systems and within virtual environments that contain resources that point to external network resources as these could be used to gather credentials when the files are rendered.

4.2.8.5. *Exploitation for Credential Access (T1212)*

Technique Information	
Technique ID	T1212
Technique Name	Exploitation for Credential Access
Technique Description	Adversaries may exploit software vulnerabilities in an attempt to collect credentials. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. Credentialing and authentication mechanisms may be targeted for exploitation by adversaries as a means to gain access to useful credentials or circumvent the process to gain access to systems. One example of this is MS14-068, which targets Kerberos and can be used to forge Kerberos tickets using domain user permissions. Exploitation for credential access may also result in Privilege Escalation depending on the process targeted or credentials obtained.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Driver: Driver Load	45.98%	45.98%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Script: Script Execution	0.0%	0.0%
Module: Module Load	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**21.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**37.84%**
Overall Log Collection Coverage	**37.25%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of exploitation. Risks of additional exploits and weaknesses in these systems may still exist.
Exploit Protection	Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation Experience Toolkit (EMET) can be used to mitigate some exploitation behavior.
Threat Intelligence Program	Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization.
Update Software	Update software regularly by employing patch management for internal enterprise endpoints and servers.
Implement Detection/Monitoring Capabilities	Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for

	behavior on the system that might indicate successful compromise, such as abnormal behavior of processes. Credential resources obtained through exploitation may be detectable in use if they are not normally used or seen.

4.2.8.6. Steal Application Access Token (T1528)

Technique Information	
Technique ID	T1528
Technique Name	Steal Application Access Token
Technique Description	Adversaries can steal application access tokens as a means of acquiring credentials to access remote systems and resources. Application access tokens are used to make authorized API requests on behalf of a user or service and are commonly used as a way to access resources in cloud and container-based applications and software-as-a-service (SaaS). OAuth is one commonly implemented framework that issues tokens to users for access to systems. Adversaries who steal account API tokens in cloud and containerized environments may be able to access data and perform actions with the permissions of these accounts, which can lead to privilege escalation and further compromise of the environment. In Kubernetes environments, processes running inside a container communicate with the Kubernetes API server using service account tokens. If a container is compromised, an attacker may be able to steal the container's token and thereby gain access to Kubernetes API commands. Token theft can also occur through social engineering, in which case user action may be required to grant access. An application desiring access to cloud-based services or protected APIs can gain entry using OAuth 2.0 through a variety of authorization protocols. An example commonly-used sequence is Microsoft's Authorization Code Grant flow. An OAuth access token enables a third-party application to interact with resources containing user data in the ways requested by the application without obtaining user credentials. Adversaries can leverage OAuth authorization by constructing a malicious application designed to be granted access to resources with the target user's OAuth token. The adversary will need to complete registration of their application with the authorization server, for example Microsoft Identity Platform using Azure Portal, the Visual Studio IDE, the command-line

	interface, PowerShell, or REST API calls. Then, they can send a [Spearphishing Link](T1566.002) to the target user to entice them to grant access to the application. Once the OAuth access token is granted, the application can gain potentially long-term access to features of the user account through [Application Access Token](T1550.001). Application access tokens may function within a limited lifetime, limiting how long an adversary can utilize the stolen token. However, in some cases, adversaries can also steal application refresh tokens, allowing them to obtain new access tokens without prompting the user.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
User Account: User Account Modification	66.1%	63.61%

Technique Analysis	
Overall Score	**36.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**27/100**
Overall Log Source Coverage	**66.1%**
Overall Log Collection Coverage	**63.61%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations

Name	Description
Audit	Administrators should audit all cloud and container accounts to ensure that they are necessary and that the permissions granted to them are appropriate. Additionally, administrators should perform an audit of all OAuth applications and the permissions they have been granted to access organizational data. This should be done extensively on all applications in order to establish a baseline, followed up on with periodic audits of new or updated applications. Suspicious applications should be investigated and removed.
Restrict Web-Based Content	Administrators can block end-user consent to OAuth applications, disabling users from authorizing third-party apps through OAuth 2.0 and forcing administrative consent for all requests. They can also block end-user registration of applications by their users, to reduce risk. A Cloud Access Security Broker can also be used to ban applications.
User Account Management	Enforce role-based access control to limit accounts to the least privileges they require. A Cloud Access Security Broker (CASB) can be used to set usage policies and manage user permissions on cloud applications to prevent access to application access tokens. In Kubernetes applications, set "automountServiceAccountToken: false" in the YAML specification of pods that do not require access to service account tokens.
User Training	Users need to be trained to not authorize third-party applications they don\u2019t recognize. The user should pay particular attention to the redirect URL: if the URL is a misspelled or convoluted sequence of words related to an expected service or SaaS application, the website is likely trying to spoof a legitimate service. Users should also be cautious about the permissions they are granting to apps. For example, offline access and access to read emails should excite higher suspicions because adversaries can utilize SaaS APIs to discover credentials and other sensitive communications.

Implement Detection/Monitoring Capabilities	Administrators should set up monitoring to trigger automatic alerts when policy criteria are met. For example, using a Cloud Access Security Broker (CASB), admins can create a "High severity app permissions" policy that generates alerts if apps request high severity permissions or send permissions requests for too many users. Security analysts can hunt for malicious apps using the tools available in their CASB, identity provider, or resource provider (depending on platform.) For example, they can filter for apps that are authorized by a small number of users, apps requesting high risk permissions, permissions incongruous with the app's purpose, or apps with old "Last authorized" fields. A specific app can be investigated using an activity log displaying activities the app has performed, although some activities may be mis-logged as being performed by the user. App stores can be useful resources to further investigate suspicious apps. Administrators can set up a variety of logs and leverage audit tools to monitor actions that can be conducted as a result of OAuth 2.0 access. For instance, audit reports enable admins to identify privilege escalation actions such as role creations or policy modifications, which could be actions performed after initial access.

4.2.8.7. *Steal Web Session Cookie (T1539)*

Technique Information	
Technique ID	T1539
Technique Name	Steal Web Session Cookie
Technique Description	An adversary may steal web application or service session cookies and use them to gain access to web applications or Internet services as an authenticated user without needing credentials. Web applications and services often use session cookies as an authentication token after a user has authenticated to a website. Cookies are often valid for an extended period of time, even if the web application is not actively used. Cookies can be found on disk, in the process memory of the browser, and in network traffic to remote systems. Additionally, other applications on the targets machine might store sensitive authentication cookies in memory (e.g. apps which authenticate to cloud services). Session cookies can be used to bypasses some multi-factor authentication protocols. There are several examples of malware targeting cookies from web browsers on the local system. There are also open source frameworks such as Evilginx 2 and Muraena that can gather session cookies through a malicious proxy (ex: [Adversary-in-the-Middle](T1557)) that can be set up by an adversary and used in phishing campaigns. After an adversary acquires a valid cookie, they can then perform a [Web Session Cookie](T1550.004) technique to login to the corresponding web application.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Process: Process Access	45.98%	45.98%
File: File Access	37.14%	29.99%

Technique Analysis	
Overall Score	**22.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**41.56%**
Overall Log Collection Coverage	**37.98%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Multi-factor Authentication	A physical second factor key that uses the target login domain as part of the negotiation protocol will prevent session cookie theft through proxy methods.
Software Configuration	Configure browsers or tasks to regularly delete persistent cookies.
User Training	Train users to identify aspects of phishing attempts where they're asked to enter credentials into a site that has the incorrect domain for the application they are logging into.
Implement Detection/Monitoring Capabilities	Monitor for attempts to access files and repositories on a local system that are used to store browser session cookies. Monitor for attempts by programs to inject into or dump browser process memory.

4.2.8.8. *Unsecured Credentials (T1552)*

Technique Information	
Technique ID	T1552
Technique Name	Unsecured Credentials
Technique Description	Adversaries may search compromised systems to find and obtain insecurely stored credentials. These credentials can be stored and/or misplaced in many locations on a system, including plaintext files (e.g. [Bash History](T1552.003)), operating system or application-specific repositories (e.g. [Credentials in Registry](T1552.002)), or other specialized files/artifacts (e.g. [Private Keys](T1552.004)).

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Command: Command Execution	45.98%	45.98%
Windows Registry: Windows Registry Key Access	100.0%	0.0%
File: File Access	37.14%	29.99%
User Account: User Account Authentication	100.0%	98.28%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**42.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**58/100**
Overall Log Source Coverage	**64.47%**

Overall Log Collection Coverage	**42.33%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Active Directory Configuration	Remove vulnerable Group Policy Preferences.
Audit	Preemptively search for files containing passwords or other credentials and take actions to reduce the exposure risk when found.
Encrypt Sensitive Information	When possible, store keys on separate cryptographic hardware instead of on the local system.
Filter Network Traffic	Limit access to the Instance Metadata API using a host-based firewall such as iptables. A properly configured Web Application Firewall (WAF) may help prevent external adversaries from exploiting Server-side Request Forgery (SSRF) attacks that allow access to the Cloud Instance Metadata API.
Operating System Configuration	There are multiple methods of preventing a user's command history from being flushed to their .bash_history file, including use of the following commands:
Password Policies	Use strong passphrases for private keys to make cracking difficult. Do not store credentials within the Registry. Establish an organizational policy that prohibits password storage in files.
Privileged Account Management	If it is necessary that software must store credentials in the Registry, then ensure the associated accounts have limited permissions so they cannot be abused if obtained by an adversary.

Restrict File and Directory Permissions	Restrict file shares to specific directories with access only to necessary users.
Update Software	Apply patch KB2962486 which prevents credentials from being stored in GPPs.
User Training	Ensure that developers and system administrators are aware of the risk associated with having plaintext passwords in software configuration files that may be left on endpoint systems or servers.
Implement Detection/Monitoring Capabilities	While detecting adversaries accessing credentials may be difficult without knowing they exist in the environment, it may be possible to detect adversary use of credentials they have obtained. Monitor the command-line arguments of executing processes for suspicious words or regular expressions that may indicate searching for a password (for example: password, pwd, login, secure, or credentials). See [Valid Accounts](T1078) for more information. Monitor for suspicious file access activity, specifically indications that a process is reading multiple files in a short amount of time and/or using command-line arguments indicative of searching for credential material (ex: regex patterns). These may be indicators of automated/scripted credential access behavior. Monitoring when the user's `.bash_history` is read can help alert to suspicious activity. While users do typically rely on their history of commands, they often access this history through other utilities like "history" instead of commands like `cat ~/.bash_history`. Additionally, monitor processes for applications that can be used to query the Registry, such as [Reg](S0075), and collect command parameters that may indicate credentials are being searched. Correlate activity with related suspicious behavior that may indicate an active intrusion to reduce false positives.

4.2.8.9. Credentials from Password Stores (T1555)

Technique Information	
Technique ID	T1555
Technique Name	Credentials from Password Stores
Technique Description	Adversaries may search for common password storage locations to obtain user credentials. Passwords are stored in several places on a system, depending on the operating system or application holding the credentials. There are also specific applications that store passwords to make it easier for users manage and maintain. Once credentials are obtained, they can be used to perform lateral movement and access restricted information.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
File: File Access	37.14%	29.99%
Process: Process Access	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**64.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**43.65%**
Overall Log Collection Coverage	**41.5%**

Detection Capability Present	Yes
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Password Policies	The password for the user's login keychain can be changed from the user's login password. This increases the complexity for an adversary because they need to know an additional password.
Implement Detection/Monitoring Capabilities	Monitor system calls, file read events, and processes for suspicious activity that could indicate searching for a password or other activity related to performing keyword searches (e.g. password, pwd, login, store, secure, credentials, etc.) in process memory for credentials. File read events should be monitored surrounding known password storage applications.

4.2.8.10. *Steal or Forge Kerberos Tickets (T1558)*

Technique Information	
Technique ID	T1558
Technique Name	Steal or Forge Kerberos Tickets
Technique Description	Adversaries may attempt to subvert Kerberos authentication by stealing or forging Kerberos tickets to enable [Pass the Ticket](T1550.003). Kerberos is an authentication protocol widely used in modern Windows domain environments. In Kerberos environments, referred to as “realms”, there are three basic participants: client, service, and Key Distribution Center (KDC). Clients request access to a service and through the exchange of Kerberos tickets, originating from KDC, they are granted access after having successfully authenticated. The KDC is responsible for both authentication and ticket granting. Adversaries may attempt to abuse Kerberos by stealing tickets or forging tickets to enable unauthorized access. On Windows, the built-in `klist` utility can be used to list and analyze cached Kerberos tickets. Linux systems on Active Directory domains store Kerberos credentials locally in the credential cache file referred to as the "ccache". The credentials are stored in the ccache file while they remain valid and generally while a user's session lasts. On modern Redhat Enterprise Linux systems, and derivative distributions, the System Security Services Daemon (SSSD) handles Kerberos tickets. By default SSSD maintains a copy of the ticket database that can be found in `/var/lib/sss/secrets/secrets.ldb` as well as the corresponding key located in `/var/lib/sss/secrets/.secrets.mkey`. Both files require root access to read. If an adversary is able to access the database and key, the credential cache Kerberos blob can be extracted and converted into a usable Kerberos ccache file that adversaries may use for [Pass the Ticket](T1550.003). The ccache file may also be converted into a Windows format using tools such as Kekeo. Kerberos tickets on macOS are stored in a standard ccache format, similar to Linux. By default, access to these ccache entries is federated through the KCM daemon process via the

	Mach RPC protocol, which uses the caller's environment to determine access. The storage location for these ccache entries is influenced by the `/etc/krb5.conf` configuration file and the `KRB5CCNAME` environment variable which can specify to save them to disk or keep them protected via the KCM daemon. Users can interact with ticket storage using `kinit`, `klist`, `ktutil`, and `kcc` built-in binaries or via Apple's native Kerberos framework. Adversaries can use open source tools to interact with the ccache files directly or to use the Kerberos framework to call lower-level APIs for extracting the user's TGT or Service Tickets.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
Active Directory: Active Directory Credential Request	100.0%	100.0%
Logon Session: Logon Session Metadata	100.0%	96.57%

Technique Analysis	
Overall Score	**61.0%**
Status	**Needs future improvements**
Sector Specific Priority	**16/100**
Overall Log Source Coverage	**70.78%**
Overall Log Collection Coverage	**68.13%**
Detection Capability Present	**Yes**

Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Active Directory Configuration	For containing the impact of a previously generated golden ticket, reset the built-in KRBTGT account password twice, which will invalidate any existing golden tickets that have been created with the KRBTGT hash and other Kerberos tickets derived from it. For each domain, change the KRBTGT account password once, force replication, and then change the password a second time. Consider rotating the KRBTGT account password every 180 days.
Encrypt Sensitive Information	Enable AES Kerberos encryption (or another stronger encryption algorithm), rather than RC4, where possible.
Password Policies	Ensure strong password length (ideally 25+ characters) and complexity for service accounts and that these passwords periodically expire.
Privileged Account Management	Limit domain admin account permissions to domain controllers and limited servers. Delegate other admin functions to separate accounts.
Implement Detection/Monitoring Capabilities	Monitor for anomalous Kerberos activity, such as malformed or blank fields in Windows logon/logoff events (Event ID 4624, 4672, 4634), RC4 encryption within ticket granting tickets (TGTs), and ticket granting service (TGS) requests without preceding TGT requests. Monitor the lifetime of TGT tickets for values that differ from the default domain duration. Monitor for indications of [Pass the Ticket](T1550.003) being used to move laterally. Enable Audit Kerberos Service Ticket Operations to

	log Kerberos TGS service ticket requests. Particularly investigate irregular patterns of activity (ex: accounts making numerous requests, Event ID 4769, within a small time frame, especially if they also request RC4 encryption [Type 0x17]). Monitor for unexpected processes interacting with lsass.exe. Common credential dumpers such as [Mimikatz](S0002) access the LSA Subsystem Service (LSASS) process by opening the process, locating the LSA secrets key, and decrypting the sections in memory where credential details, including Kerberos tickets, are stored. Monitor for unusual processes accessing \`secrets.ldb\` and \`.secrets.mkey\` located in \`/var/lib/sss/secrets/\`.

4.2.8.11. Forge Web Credentials (T1606)

Technique Information	
Technique ID	T1606
Technique Name	Forge Web Credentials
Technique Description	Adversaries may forge credential materials that can be used to gain access to web applications or Internet services. Web applications and services (hosted in cloud SaaS environments or on-premise servers) often use session cookies, tokens, or other materials to authenticate and authorize user access. Adversaries may generate these credential materials in order to gain access to web resources. This differs from [Steal Web Session Cookie](T1539), [Steal Application Access Token](T1528), and other similar behaviors in that the credentials are new and forged by the adversary, rather than stolen or intercepted from legitimate users. The generation of web credentials often requires secret values, such as passwords, [Private Keys](T1552.004), or other cryptographic seed values. Once forged, adversaries may use these web credentials to access resources (ex: [Use Alternate Authentication Material](T1550)), which may bypass multi-factor and other authentication protection mechanisms.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Web Credential: Web Credential Usage	0.0%	0.0%
Logon Session: Logon Session Creation	100.0%	96.57%
Web Credential: Web Credential Creation	0.0%	0.0%

Technique Analysis	
Overall Score	**18.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**34/100**
Overall Log Source Coverage	**33.33%**
Overall Log Collection Coverage	**32.19%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Audit	Administrators should perform an audit of all access lists and the permissions they have been granted to access web applications and services. This should be done extensively on all resources in order to establish a baseline, followed up on with periodic audits of new or updated resources. Suspicious accounts/credentials should be investigated and removed.
Privileged Account Management	Restrict permissions and access to the AD FS server to only originate from privileged access workstations.
Software Configuration	Configure browsers/applications to regularly delete persistent web credentials (such as cookies).
User Account Management	Ensure that user accounts with administrative rights follow best practices, including use of privileged access workstations, Just in Time/Just Enough Administration (JIT/JEA), and strong authentication. Reduce the number of users that are members of highly privileged Directory Roles.

Implement Detection/Monitoring Capabilities	Monitor for anomalous authentication activity, such as logons or other user session activity associated with unknown accounts. Monitor for unexpected and abnormal access to resources, including access of websites and cloud-based applications by the same user in different locations or by different systems that do not match expected configurations.

4.2.8.12. Multi-Factor Authentication Request Generation (T1621)

Technique Information	
Technique ID	T1621
Technique Name	Multi-Factor Authentication Request Generation
Technique Description	Adversaries may attempt to bypass multi-factor authentication (MFA) mechanisms and gain access to accounts by generating MFA requests sent to users. Adversaries in possession credentials to [Valid Accounts](T1078) may be unable to complete the login process if they lack access to the 2FA or MFA mechanisms required as an additional credential and security control. To circumvent this, adversaries may abuse the automatic generation of push notifications to MFA services such as Duo Push, Microsoft Authenticator, Okta, or similar services to have the user grant access to their account. In some cases, adversaries may continuously repeat login attempts in order to bombard users with MFA push notifications, SMS messages, and phone calls, potentially resulting in the user finally accepting the authentication request in response to "MFA fatigue."

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
User Account: User Account Authentication	100.0%	98.28%
Logon Session: Logon Session Metadata	100.0%	96.57%
Logon Session: Logon Session Creation	100.0%	96.57%

Technique Analysis

Overall Score	**88.0%**
Status	**Good maturity**
Sector Specific Priority	**5/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**97.14%**
Detection Capability Present	**Yes**
Detection Sources	• Sentinel

Mitigations	
Name	**Description**
Account Use Policies	Enable account restrictions to prevent login attempts, and the subsequent 2FA/MFA service requests, from being initiated from suspicious locations or when the source of the login attempts do not match the location of the 2FA/MFA smart device.
Multi-factor Authentication	Implement more secure 2FA/MFA mechanisms in replacement of simple push or one-click 2FA/MFA options. For example, having users enter a one-time code provided by the login screen into the 2FA/MFA application or utilizing other out-of-band 2FA/MFA mechanisms (such as rotating code-based hardware tokens providing rotating codes that need an accompanying user pin) may be more secure. Furthermore, change default configurations and implement limits upon the maximum number of 2FA/MFA request prompts that can be sent to users in period of time.
User Training	Train users to only accept 2FA/MFA requests from login attempts they initiated, to review source location of the login attempt prompting the 2FA/MFA requests, and to report suspicious/unsolicited prompts.

Implement Detection/Monitoring Capabilities	Monitor user account logs as well as 2FA/MFA application logs for suspicious events: unusual login attempt source location, mismatch in location of login attempt and smart device receiving 2FA/MFA request prompts, and high volume of repeated login attempts, all of which may indicate user's primary credentials have been compromised minus 2FA/MFA mechanism.

4.2.9. Discovery

4.2.9.1. System Service Discovery (T1007)

Technique Information	
Technique ID	T1007
Technique Name	System Service Discovery
Technique Description	Adversaries may try to gather information about registered local system services. Adversaries may obtain information about services using tools as well as OS utility commands such as `sc query`, `tasklist /svc`, `systemctl --type=service`, and `net start`. Adversaries may use the information from [System Service Discovery](T1007) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**61.0%**
Status	**Needs future improvements**
Sector Specific Priority	**16/100**
Overall Log Source Coverage	**42.61%**
Overall Log Collection Coverage	**41.68%**
Detection Capability Present	**Yes**

Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system information related to services. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.2. *Application Window Discovery (T1010)*

Technique Information	
Technique ID	T1010
Technique Name	Application Window Discovery
Technique Description	Adversaries may attempt to get a listing of open application windows. Window listings could convey information about how the system is used or give context to information collected by a keylogger.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**24.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**45.04%**
Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations

Name	Description
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.3. *Query Registry (T1012)*

Technique Information	
Technique ID	T1012
Technique Name	Query Registry
Technique Description	Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software. The Registry contains a significant amount of information about the operating system, configuration, software, and security. Information can easily be queried using the [Reg](S0075) utility, though other means to access the Registry exist. Some of the information may help adversaries to further their operation within a network. Adversaries may use the information from [Query Registry](T1012) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Windows Registry: Windows Registry Key Access	100.0%	0.0%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**62.0%**
Status	**Needs future improvements**

Sector Specific Priority	**16/100**
Overall Log Source Coverage	**58.78%**
Overall Log Collection Coverage	**32.88%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Interaction with the Windows Registry may come from the command line using utilities such as [Reg](S0075) or through running malware that may interact with the Registry through an API. Command-line invocation of utilities used to query the Registry may be detected through process and command-line monitoring. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.4. System Network Configuration Discovery (T1016)

Technique Information	
Technique ID	T1016
Technique Name	System Network Configuration Discovery
Technique Description	Adversaries may look for details about the network configuration and settings, such as IP and/or MAC addresses, of systems they access or through information discovery of remote systems. Several operating system administration utilities exist that can be used to gather this information. Examples include [Arp](S0099), [ipconfig](S0100)/[ifconfig](S0101), [nbtstat](S0102), and [route](S0103). Adversaries may also leverage a [Network Device CLI](T1059.008) on network devices to gather information about configurations and settings, such as IP addresses of configured interfaces and static/dynamic routes. Adversaries may use the information from [System Network Configuration Discovery](T1016) during automated discovery to shape follow-on behaviors, including determining certain access within the target network and what actions to do next.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis

Overall Score	**56.0%**
Status	**Needs future improvements**
Sector Specific Priority	**37/100**
Overall Log Source Coverage	**33.78%**
Overall Log Collection Coverage	**32.88%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Further, {{LinkById|T1059.008} commands may also be used to gather system and network information with built-in features native to the network device platform. Monitor CLI activity for unexpected or unauthorized use commands being run by non-standard users from non-standard locations. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.5. Remote System Discovery (T1018)

Technique Information	
Technique ID	T1018
Technique Name	Remote System Discovery
Technique Description	Adversaries may attempt to get a listing of other systems by IP address, hostname, or other logical identifier on a network that may be used for Lateral Movement from the current system. Functionality could exist within remote access tools to enable this, but utilities available on the operating system could also be used such as [Ping](S0097) or `net view` using [Net](S0039). Adversaries may also analyze data from local host files (ex: `C:\Windows\System32\Drivers\etc\hosts` or `/etc/hosts`) or other passive means (such as local [Arp](S0099) cache entries) in order to discover the presence of remote systems in an environment. Adversaries may also target discovery of network infrastructure as well as leverage [Network Device CLI](T1059.008) commands on network devices to gather detailed information about systems within a network.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Connection Creation	45.45%	45.45%
Process: Process Creation	39.25%	37.38%

Technique Analysis

Overall Score	**60.0%**
Status	**Needs future improvements**
Sector Specific Priority	**40/100**
Overall Log Source Coverage	**41.96%**
Overall Log Collection Coverage	**39.7%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events related to legitimate remote system discovery may be uncommon, depending on the environment and how they are used. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). Monitor for processes that can be used to discover remote systems, such as `ping.exe` and `tracert.exe`, especially when executed in quick succession.

4.2.9.6. System Owner/User Discovery (T1033)

Technique Information	
Technique ID	T1033
Technique Name	System Owner/User Discovery
Technique Description	Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using [OS Credential Dumping](T1003). The information may be collected in a number of different ways using other Discovery techniques, because user and username details are prevalent throughout a system and include running process ownership, file/directory ownership, session information, and system logs. Adversaries may use the information from [System Owner/User Discovery](T1033) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Various utilities and commands may acquire this information, including `whoami`. In macOS and Linux, the currently logged in user can be identified with `w` and `who`. On macOS the `dscl . list /Users | grep -v '_'` command can also be used to enumerate user accounts. Environment variables, such as `%USERNAME%` and `$USER`, may also be used to access this information.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**61.0%**

Status	**Needs future improvements**
Sector Specific Priority	**16/100**
Overall Log Source Coverage	**42.61%**
Overall Log Collection Coverage	**41.68%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.7. Network Service Discovery (T1046)

Technique Information	
Technique ID	T1046
Technique Name	Network Service Discovery
Technique Description	Adversaries may attempt to get a listing of services running on remote hosts and local network infrastructure devices, including those that may be vulnerable to remote software exploitation. Common methods to acquire this information include port and/or vulnerability scans using tools that are brought onto a system. Within cloud environments, adversaries may attempt to discover services running on other cloud hosts. Additionally, if the cloud environment is connected to a on-premises environment, adversaries may be able to identify services running on non-cloud systems as well. Within macOS environments, adversaries may use the native Bonjour application to discover services running on other macOS hosts within a network. The Bonjour mDNSResponder daemon automatically registers and advertises a host's registered services on the network. For example, adversaries can use a mDNS query (such as \`dns-sd -B _ssh._tcp .\`) to find other systems broadcasting the ssh service.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Cloud Service: Cloud Service Enumeration	100.0%	100.0%
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**45/100**
Overall Log Source Coverage	**70.21%**
Overall Log Collection Coverage	**48.66%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Disable or Remove Feature or Program	Ensure that unnecessary ports and services are closed to prevent risk of discovery and potential exploitation.
Network Intrusion Prevention	Use network intrusion detection/prevention systems to detect and prevent remote service scans.
Network Segmentation	Ensure proper network segmentation is followed to protect critical servers and devices.
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events from legitimate remote service scanning may be uncommon, depending on the environment and how they are used. Legitimate open port and vulnerability scanning may be conducted within the environment and will need to be deconflicted with

	any detection capabilities developed. Network intrusion detection systems can also be used to identify scanning activity. Monitor for process use of the networks and inspect intra-network flows to detect port scans.

4.2.9.8. System Network Connections Discovery (T1049)

Technique Information	
Technique ID	T1049
Technique Name	System Network Connections Discovery
Technique Description	Adversaries may attempt to get a listing of network connections to or from the compromised system they are currently accessing or from remote systems by querying for information over the network. An adversary who gains access to a system that is part of a cloud-based environment may map out Virtual Private Clouds or Virtual Networks in order to determine what systems and services are connected. The actions performed are likely the same types of discovery techniques depending on the operating system, but the resulting information may include details about the networked cloud environment relevant to the adversary's goals. Cloud providers may have different ways in which their virtual networks operate. Similarly, adversaries who gain access to network devices may also perform similar discovery activities to gather information about connected systems and services. Utilities and commands that acquire this information include [netstat](S0104), "net use," and "net session" with [Net](S0039). In Mac and Linux, [netstat](S0104) and \`lsof\` can be used to list current connections. \`who -a\` and \`w\` can be used to show which users are currently logged in, similar to "net session". Additionally, built-in features native to network devices and [Network Device CLI](T1059.008) may be used.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**62.0%**
Status	**Needs future improvements**
Sector Specific Priority	**35/100**
Overall Log Source Coverage	**45.04%**
Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Further, [Network Device CLI](T1059.008) commands may also be used to gather system and network information with built-in features native to the network device platform. Monitor CLI activity for unexpected or unauthorized use commands being run by non-standard users from non-standard locations. Information may also be acquired

	through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.9. Process Discovery (T1057)

Technique Information	
Technique ID	T1057
Technique Name	Process Discovery
Technique Description	Adversaries may attempt to get information about running processes on a system. Information obtained could be used to gain an understanding of common software/applications running on systems within the network. Adversaries may use the information from [Process Discovery](T1057) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. In Windows environments, adversaries could obtain details on running processes using the [Tasklist](S0057) utility via [cmd](S0106) or `Get-Process` via [PowerShell](T1059.001). Information about processes can also be extracted from the output of [Native API](T1106) calls such as `CreateToolhelp32Snapshot`. In Mac and Linux, this is accomplished with the `ps` command. Adversaries may also opt to enumerate processes via /proc.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**69.0%**
Status	**Could benefit from improvments**

Sector Specific Priority	**26/100**
Overall Log Source Coverage	**45.04%**
Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • ESET Antivirus

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events that look like process discovery may be uncommon, depending on the environment and how they are used. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.10. *Permission Groups Discovery (T1069)*

Technique Information	
Technique ID	T1069
Technique Name	Permission Groups Discovery
Technique Description	Adversaries may attempt to find group and permission settings. This information can help adversaries determine which user accounts and groups are available, the membership of users in particular groups, and which users and groups have elevated permissions.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Group: Group Enumeration	100.0%	98.28%
Application Log: Application Log Content	66.67%	33.33%
Pod: Pod Metadata	44.0%	22.0%
Group: Group Metadata	100.0%	98.28%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**71.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**12/100**
Overall Log Source Coverage	**65.98%**
Overall Log Collection Coverage	**55.87%**

Detection Capability Present	Yes
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). Monitor container logs for commands and/or API calls related to listing permissions for pods and nodes, such as `kubectl auth can-i`.

4.2.9.11. System Information Discovery (T1082)

Technique Information	
Technique ID	T1082
Technique Name	System Information Discovery
Technique Description	An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](T1082) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Tools such as [Systeminfo](S0096) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](T1059.008) on network devices to gather detailed system information. [System Information Discovery](T1082) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment. Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Process: Process Creation	39.25%	37.38%
Instance: Instance Metadata	95.0%	30.0%

Technique Analysis	
Overall Score	**64.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**57.53%**
Overall Log Collection Coverage	**40.38%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Further, [Network Device CLI](T1059.008) commands may also be used to gather detailed system information with built-in features native to the network device platform. Monitor CLI activity for unexpected or unauthorized use commands being run by non-standard users from non-standard locations.

	Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). In cloud-based systems, native logging can be used to identify access to certain APIs and dashboards that may contain system information. Depending on how the environment is used, that data alone may not be useful due to benign use during normal operations.

4.2.9.12. File and Directory Discovery (T1083)

Technique Information	
Technique ID	T1083
Technique Name	File and Directory Discovery
Technique Description	Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery](T1083) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Many command shell utilities can be used to obtain this information. Examples include `dir`, `tree`, `ls`, `find`, and `locate`. Custom tools may also be used to gather file and directory information and interact with the [Native API](T1106). Adversaries may also leverage a [Network Device CLI](T1059.008) on network devices to gather file and directory information.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**62.0%**
Status	**Needs future improvements**
Sector Specific Priority	**32/100**

Overall Log Source Coverage	**45.04%**
Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Collection and Exfiltration, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). Further, [Network Device CLI](T1059.008) commands may also be used to gather file and directory information with built-in features native to the network device platform. Monitor CLI activity for unexpected or unauthorized use of commands being run by non-standard users from non-standard locations.

4.2.9.13. Account Discovery (T1087)

Technique Information	
Technique ID	T1087
Technique Name	Account Discovery
Technique Description	Adversaries may attempt to get a listing of accounts on a system or within an environment. This information can help adversaries determine which accounts exist to aid in follow-on behavior.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
User Account: User Account Metadata	100.0%	98.28%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**67.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**14/100**
Overall Log Source Coverage	**55.59%**
Overall Log Collection Coverage	**52.91%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Operating System Configuration	Prevent administrator accounts from being enumerated when an application is elevating through UAC since it can lead to the disclosure of account names. The Registry key is located
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). Monitor for processes that can be used to enumerate user accounts, such as `net.exe` and `net1.exe`, especially when executed in quick succession.

4.2.9.14. *Peripheral Device Discovery (T1120)*

Technique Information	
Technique ID	T1120
Technique Name	Peripheral Device Discovery
Technique Description	Adversaries may attempt to gather information about attached peripheral devices and components connected to a computer system. Peripheral devices could include auxiliary resources that support a variety of functionalities such as keyboards, printers, cameras, smart card readers, or removable storage. The information may be used to enhance their awareness of the system and network environment or may be used for further actions.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**62.0%**
Status	**Needs future improvements**
Sector Specific Priority	**38/100**
Overall Log Source Coverage	**45.04%**
Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.15. *System Time Discovery (T1124)*

Technique Information	
Technique ID	T1124
Technique Name	System Time Discovery
Technique Description	An adversary may gather the system time and/or time zone from a local or remote system. The system time is set and stored by the Windows Time Service within a domain to maintain time synchronization between systems and services in an enterprise network. System time information may be gathered in a number of ways, such as with [Net](S0039) on Windows by performing `net time \\hostname` to gather the system time on a remote system. The victim's time zone may also be inferred from the current system time or gathered by using `w32tm /tz`. This information could be useful for performing other techniques, such as executing a file with a [Scheduled Task/Job](T1053) , or to discover locality information based on time zone to assist in victim targeting (i.e. [System Location Discovery](T1614)). Adversaries may also use knowledge of system time as part of a time bomb, or delaying execution until a specified date/time.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**24.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**45.04%**
Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Command-line interface monitoring may be useful to detect instances of net.exe or other command-line utilities being used to gather system time or time zone. Methods of detecting API use for gathering this information are likely less useful due to how often they may be used by legitimate software.

4.2.9.16. Network Share Discovery (T1135)

Technique Information	
Technique ID	T1135
Technique Name	Network Share Discovery
Technique Description	Adversaries may look for folders and drives shared on remote systems as a means of identifying sources of information to gather as a precursor for Collection and to identify potential systems of interest for Lateral Movement. Networks often contain shared network drives and folders that enable users to access file directories on various systems across a network. File sharing over a Windows network occurs over the SMB protocol. [Net](S0039) can be used to query a remote system for available shared drives using the `net view \\\\remotesystem` command. It can also be used to query shared drives on the local system using `net share`. For macOS, the `sharing -l` command lists all shared points used for smb services.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**42.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**58/100**
Overall Log Source Coverage	**45.04%**

Overall Log Collection Coverage	**43.84%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Operating System Configuration	Enable Windows Group Policy "Do Not Allow Anonymous Enumeration of SAM Accounts and Shares" security setting to limit users who can enumerate network shares.
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Lateral Movement, based on the information obtained. Normal, benign system and network events related to legitimate remote system discovery may be uncommon, depending on the environment and how they are used. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.17. Password Policy Discovery (T1201)

Technique Information	
Technique ID	T1201
Technique Name	Password Policy Discovery
Technique Description	Adversaries may attempt to access detailed information about the password policy used within an enterprise network or cloud environment. Password policies are a way to enforce complex passwords that are difficult to guess or crack through [Brute Force](T1110). This information may help the adversary to create a list of common passwords and launch dictionary and/or brute force attacks which adheres to the policy (e.g. if the minimum password length should be 8, then not trying passwords such as 'pass123'; not checking for more than 3-4 passwords per account if the lockout is set to 6 as to not lock out accounts). Password policies can be set and discovered on Windows, Linux, and macOS systems via various command shell utilities such as `net accounts (/domain)`, `Get-ADDefaultDomainPasswordPolicy`, `chage -l <username>`, `cat /etc/pam.d/common-password`, and `pwpolicy getaccountpolicies` . Adversaries may also leverage a [Network Device CLI](T1059.008) on network devices to discover password policy information. Password policies can be discovered in cloud environments using available APIs such as `GetAccountPasswordPolicy` in AWS .

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
User Account: User Account Metadata	100.0%	98.28%

Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**79.0%**
Status	**Good maturity**
Sector Specific Priority	**9/100**
Overall Log Source Coverage	**61.74%**
Overall Log Collection Coverage	**60.55%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Password Policies	Ensure only valid password filters are registered. Filter DLLs must be present in Windows installation directory (
Implement Detection/Monitoring Capabilities	Monitor logs and processes for tools and command line arguments that may indicate they're being used for password policy discovery. Correlate that activity with other suspicious activity from the originating system to reduce potential false positives from valid user or administrator activity. Adversaries will likely attempt to find the password policy early in an operation and the activity is likely to happen with other Discovery activity.

4.2.9.18. *Browser Bookmark Discovery (T1217)*

Technique Information	
Technique ID	T1217
Technique Name	Browser Bookmark Discovery
Technique Description	Adversaries may enumerate browser bookmarks to learn more about compromised hosts. Browser bookmarks may reveal personal information about users (ex: banking sites, interests, social media, etc.) as well as details about internal network resources such as servers, tools/dashboards, or other related infrastructure. Browser bookmarks may also highlight additional targets after an adversary has access to valid credentials, especially [Credentials In Files](T1552.001) associated with logins cached by a browser. Specific storage locations vary based on platform and/or application, but browser bookmarks are typically stored in local files/databases.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**22.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**

Overall Log Source Coverage	**40.79%**
Overall Log Collection Coverage	**37.78%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor processes and command-line arguments for actions that could be taken to gather browser bookmark information. Remote access tools with built-in features may interact directly using APIs to gather information. Information may also be acquired through system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Collection and Exfiltration, based on the information obtained.

4.2.9.19. Domain Trust Discovery (T1482)

Technique Information	
Technique ID	T1482
Technique Name	Domain Trust Discovery
Technique Description	Adversaries may attempt to gather information on domain trust relationships that may be used to identify lateral movement opportunities in Windows multi-domain/forest environments. Domain trusts provide a mechanism for a domain to allow access to resources based on the authentication procedures of another domain. Domain trusts allow the users of the trusted domain to access resources in the trusting domain. The information discovered may help the adversary conduct [SID-History Injection](T1134.005), [Pass the Ticket](T1550.003), and [Kerberoasting](T1558.003). Domain trusts can be enumerated using the `DSEnumerateDomainTrusts()` Win32 API call, .NET methods, and LDAP. The Windows utility [Nltest](S0359) is known to be used by adversaries to enumerate domain trusts.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**63.0%**
Status	**Needs future improvements**
Sector Specific Priority	**15/100**

Overall Log Source Coverage	**33.78%**
Overall Log Collection Coverage	**32.88%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Audit	Map the trusts within existing domains/forests and keep trust relationships to a minimum.
Network Segmentation	Employ network segmentation for sensitive domains.
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information, such as `nltest /domain_trusts`. Remote access tools with built-in features may interact directly with the Windows API to gather information. Look for the `DSEnumerateDomainTrusts()` Win32 API call to spot activity associated with [Domain Trust Discovery](T1482). Information may also be acquired through Windows system management tools such as [PowerShell](T1059.001). The .NET method `GetAllTrustRelationships()` can be an indicator of [Domain Trust Discovery](T1482).

4.2.9.20. Software Discovery (T1518)

Technique Information	
Technique ID	T1518
Technique Name	Software Discovery
Technique Description	Adversaries may attempt to get a listing of software and software versions that are installed on a system or in a cloud environment. Adversaries may use the information from [Software Discovery](T1518) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Adversaries may attempt to enumerate software for a variety of reasons, such as figuring out what security measures are present or if the compromised system has a version of software that is vulnerable to [Exploitation for Privilege Escalation](T1068).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Firewall: Firewall Metadata	100.0%	0.0%
Process: OS API Execution	49.89%	48.17%
Firewall: Firewall Enumeration	100.0%	0.0%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**34.0%**
Status	**Needs immediate remediation**

Sector Specific Priority	**66/100**
Overall Log Source Coverage	**67.02%**
Overall Log Collection Coverage	**26.31%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as lateral movement, based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system and network information. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.9.21. Cloud Service Discovery (T1526)

Technique Information	
Technique ID	T1526
Technique Name	Cloud Service Discovery
Technique Description	An adversary may attempt to enumerate the cloud services running on a system after gaining access. These methods can differ from platform-as-a-service (PaaS), to infrastructure-as-a-service (IaaS), or software-as-a-service (SaaS). Many services exist throughout the various cloud providers and can include Continuous Integration and Continuous Delivery (CI/CD), Lambda Functions, Azure AD, etc. Adversaries may attempt to discover information about the services enabled throughout the environment. Azure tools and APIs, such as the Azure AD Graph API and Azure Resource Manager API, can enumerate resources and services, including applications, management groups, resources and policy definitions, and their relationships that are accessible by an identity. Stormspotter is an open source tool for enumerating and constructing a graph for Azure resources and services, and Pacu is an open source AWS exploitation framework that supports several methods for discovering cloud services.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Cloud Service: Cloud Service Enumeration	100.0%	100.0%
Cloud Service: Cloud Service Metadata	100.0%	0.0%

Technique Analysis

Overall Score	**40.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**25/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**50.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Cloud service discovery techniques will likely occur throughout an operation where an adversary is targeting cloud-based systems and services. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Normal, benign system and network events that look like cloud service discovery may be uncommon, depending on the environment and how they are used. Monitor cloud service usage for anomalous behavior that may indicate adversarial presence within the environment.

4.2.9.22. Cloud Service Dashboard (T1538)

Technique Information	
Technique ID	T1538
Technique Name	Cloud Service Dashboard
Technique Description	An adversary may use a cloud service dashboard GUI with stolen credentials to gain useful information from an operational cloud environment, such as specific services, resources, and features. For example, the GCP Command Center can be used to view all assets, findings of potential security risks, and to run additional queries, such as finding public IP addresses and open ports. Depending on the configuration of the environment, an adversary may be able to enumerate more information via the graphical dashboard than an API. This allows the adversary to gain information without making any API requests.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
User Account: User Account Authentication	100.0%	98.28%
Logon Session: Logon Session Creation	100.0%	96.57%

Technique Analysis	
Overall Score	**54.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**100.0%**
Overall Log Collection Coverage	**97.42%**

Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
User Account Management	Enforce the principle of least-privilege by limiting dashboard visibility to only the resources required. This may limit the discovery value of the dashboard in the event of a compromised account.
Implement Detection/Monitoring Capabilities	Monitor account activity logs to see actions performed and activity associated with the cloud service management console. Some cloud providers, such as AWS, provide distinct log events for login attempts to the management console.

4.2.9.23. *Cloud Infrastructure Discovery (T1580)*

Technique Information	
Technique ID	T1580
Technique Name	Cloud Infrastructure Discovery
Technique Description	An adversary may attempt to discover infrastructure and resources that are available within an infrastructure-as-a-service (IaaS) environment. This includes compute service resources such as instances, virtual machines, and snapshots as well as resources of other services including the storage and database services. Cloud providers offer methods such as APIs and commands issued through CLIs to serve information about infrastructure. For example, AWS provides a `DescribeInstances` API within the Amazon EC2 API that can return information about one or more instances within an account, the `ListBuckets` API that returns a list of all buckets owned by the authenticated sender of the request, the `HeadBucket` API to determine a bucket's existence along with access permissions of the request sender, or the `GetPublicAccessBlock` API to retrieve access block configuration for a bucket. Similarly, GCP's Cloud SDK CLI provides the `gcloud compute instances list` command to list all Google Compute Engine instances in a project , and Azure's CLI command `az vm list` lists details of virtual machines. In addition to API commands, adversaries can utilize open source tools to discover cloud storage infrastructure through [Wordlist Scanning](T1595.003). An adversary may enumerate resources using a compromised user's access keys to determine which are available to that user. The discovery of these available resources may help adversaries determine their next steps in the Cloud environment, such as establishing Persistence.An adversary may also use this information to change the configuration to make the bucket publicly accessible, allowing data to be accessed without authentication. Adversaries have also may use infrastructure discovery APIs such as `DescribeDBInstances` to determine size, owner, permissions, and network ACLs of database resources. Adversaries can use this information to determine the potential value of databases and discover the requirements to access them. Unlike in [Cloud Service Discovery](T1526), this

	technique focuses on the discovery of components of the provided services rather than the services themselves.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Snapshot: Snapshot Metadata	42.3%	40.05%
Volume: Volume Enumeration	100.0%	100.0%
Cloud Storage: Cloud Storage Metadata	14.29%	9.52%
Instance: Instance Enumeration	95.0%	30.0%
Instance: Instance Metadata	95.0%	30.0%
Volume: Volume Metadata	100.0%	100.0%
Cloud Storage: Cloud Storage Enumeration	14.29%	11.9%
Snapshot: Snapshot Enumeration	42.3%	40.05%

Technique Analysis	
Overall Score	**29.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**30/100**
Overall Log Source Coverage	**62.9%**
Overall Log Collection Coverage	**45.19%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
User Account Management	Limit permissions to discover cloud infrastructure in accordance with least privilege. Organizations should limit the number of users within the organization with an IAM role that has administrative privileges, strive to reduce all permanent privileged role assignments, and conduct periodic entitlement reviews on IAM users, roles and policies.
Implement Detection/Monitoring Capabilities	Establish centralized logging for the activity of cloud infrastructure components. Monitor logs for actions that could be taken to gather information about cloud infrastructure, including the use of discovery API calls by new or unexpected users and enumerations from unknown or malicious IP addresses. To reduce false positives, valid change management procedures could introduce a known identifier that is logged with the change (e.g., tag or header) if supported by the cloud provider, to help distinguish valid, expected actions from malicious ones.

4.2.9.24. Container and Resource Discovery (T1613)

Technique Information	
Technique ID	T1613
Technique Name	Container and Resource Discovery
Technique Description	Adversaries may attempt to discover containers and other resources that are available within a containers environment. Other resources may include images, deployments, pods, nodes, and other information such as the status of a cluster. These resources can be viewed within web applications such as the Kubernetes dashboard or can be queried via the Docker and Kubernetes APIs. In Docker, logs may leak information about the environment, such as the environment's configuration, which services are available, and what cloud provider the victim may be utilizing. The discovery of these resources may inform an adversary's next steps in the environment, such as how to perform lateral movement and which methods to utilize for execution.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Container: Container Enumeration	64.0%	18.0%
Pod: Pod Enumeration	66.0%	4.0%
Container: Container Metadata	96.49%	59.65%
Pod: Pod Metadata	44.0%	22.0%
Cluster: Cluster Metadata	0.0%	0.0%

Technique Analysis	
Overall Score	**20.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**54.1%**
Overall Log Collection Coverage	**20.73%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Limit Access to Resource Over Network	Limit communications with the container service to local Unix sockets or remote access via SSH. Require secure port access to communicate with the APIs over TLS by disabling unauthenticated access to the Docker API and Kubernetes API Server.
Network Segmentation	Deny direct remote access to internal systems through the use of network proxies, gateways, and firewalls.
User Account Management	Enforce the principle of least privilege by limiting dashboard visibility to only the required users.
Implement Detection/Monitoring Capabilities	Establish centralized logging for the activity of container and Kubernetes cluster components. This can be done by deploying logging agents on Kubernetes nodes and retrieving logs from sidecar proxies for application pods to detect malicious activity at the cluster level. Monitor logs for actions that could be taken to gather information about container infrastructure, including the use of discovery API calls by new or unexpected users. Monitor account activity logs to see actions performed and activity associated with the Kubernetes dashboard and other web applications.

4.2.9.25. System Location Discovery (T1614)

Technique Information	
Technique ID	T1614
Technique Name	System Location Discovery
Technique Description	Adversaries may gather information in an attempt to calculate the geographical location of a victim host. Adversaries may use the information from [System Location Discovery](T1614) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Adversaries may attempt to infer the location of a system using various system checks, such as time zone, keyboard layout, and/or language settings. Windows API functions such as \`GetLocaleInfoW\` can also be used to determine the locale of the host. In cloud environments, an instance's availability zone may also be discovered by accessing the instance metadata service from the instance. Adversaries may also attempt to infer the location of a victim host using IP addressing, such as via online geolocation IP-lookup services.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
Instance: Instance Metadata	95.0%	30.0%
Windows Registry: Windows Registry Key Access	100.0%	0.0%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**66.02%**
Overall Log Collection Coverage	**32.31%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor processes and command-line arguments for actions that could be taken to gather system location information. Remote access tools with built-in features may interact directly with the Windows API, such as calling ` GetLocaleInfoW` to gather information. Monitor traffic flows to geo-location service provider sites, such as ip-api and ipinfo.

4.2.9.26. Group Policy Discovery (T1615)

Technique Information	
Technique ID	T1615
Technique Name	Group Policy Discovery
Technique Description	Adversaries may gather information on Group Policy settings to identify paths for privilege escalation, security measures applied within a domain, and to discover patterns in domain objects that can be manipulated or used to blend in the environment. Group Policy allows for centralized management of user and computer settings in Active Directory (AD). Group policy objects (GPOs) are containers for group policy settings made up of files stored within a predicable network path `\<DOMAIN>\SYSVOL\<DOMAIN>\Policies\`. Adversaries may use commands such as `gpresult` or various publicly available PowerShell functions, such as `Get-DomainGPO` and `Get-DomainGPOLocalGroup`, to gather information on Group Policy settings. Adversaries may use this information to shape follow-on behaviors, including determining potential attack paths within the target network as well as opportunities to manipulate Group Policy settings (i.e. [Domain Policy Modification](T1484)) for their benefit.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Command: Command Execution	45.98%	45.98%
Active Directory: Active Directory Object Access	100.0%	100.0%
Script: Script Execution	0.0%	0.0%
Network Traffic: Network Traffic Content	64.66%	0.0%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**23.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**49.98%**
Overall Log Collection Coverage	**36.67%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities based on the information obtained. Monitor for suspicious use of \`gpresult\`. Monitor for the use of PowerShell functions such as \`Get-DomainGPO\` and \`Get-DomainGPOLocalGroup\` and processes spawning with command-line arguments containing \`GPOLocalGroup\`. Monitor for abnormal LDAP queries with filters for \`groupPolicyContainer\` and high volumes of LDAP traffic to domain controllers. Windows Event ID 4661 can also be used to detect when a directory service has been accessed.

4.2.9.27. *Cloud Storage Object Discovery (T1619)*

Technique Information	
Technique ID	T1619
Technique Name	Cloud Storage Object Discovery
Technique Description	Adversaries may enumerate objects in cloud storage infrastructure. Adversaries may use this information during automated discovery to shape follow-on behaviors, including requesting all or specific objects from cloud storage. Similar to [File and Directory Discovery](T1083) on a local host, after identifying available storage services (i.e. [Cloud Infrastructure Discovery](T1580)) adversaries may access the contents/objects stored in cloud infrastructure. Cloud service providers offer APIs allowing users to enumerate objects stored within cloud storage. Examples include ListObjectsV2 in AWS and List Blobs in Azure .

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Cloud Storage: Cloud Storage Access	14.29%	14.29%
Cloud Storage: Cloud Storage Enumeration	14.29%	11.9%

Technique Analysis	
Overall Score	**7.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**39/100**
Overall Log Source Coverage	**14.29%**
Overall Log Collection Coverage	**13.1%**

Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
User Account Management	Restrict granting of permissions related to listing objects in cloud storage to necessary accounts.
Implement Detection/Monitoring Capabilities	System and network discovery techniques normally occur throughout an operation as an adversary learns the environment. Data and events should not be viewed in isolation, but as part of a chain of behavior that could lead to other activities, such as Collection and Exfiltration, based on the information obtained. Monitor cloud logs for API calls used for file or object enumeration for unusual activity.

4.2.10. Lateral Movement

4.2.10.1. Remote Services (T1021)

Technique Information	
Technique ID	T1021
Technique Name	Remote Services
Technique Description	Adversaries may use [Valid Accounts](T1078) to log into a service specifically designed to accept remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user. In an enterprise environment, servers and workstations can be organized into domains. Domains provide centralized identity management, allowing users to login using one set of credentials across the entire network. If an adversary is able to obtain a set of valid domain credentials, they could login to many different machines using remote access protocols such as secure shell (SSH) or remote desktop protocol (RDP). Legitimate applications (such as [Software Deployment Tools](T1072) and other administrative programs) may utilize [Remote Services](T1021) to access remote hosts. For example, Apple Remote Desktop (ARD) on macOS is native software used for remote management. ARD leverages a blend of protocols, including [VNC](T1021.005) to send the screen and control buffers and [SSH](T1021.004) for secure file transfer. Adversaries can abuse applications such as ARD to gain remote code execution and perform lateral movement. In versions of macOS prior to 10.14, an adversary can escalate an SSH session to an ARD session which enables an adversary to accept TCC (Transparency, Consent, and Control) prompts without user interaction and gain access to data.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%

Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Share: Network Share Access	22.81%	14.03%
Network Traffic: Network Connection Creation	45.45%	45.45%
Service: Service Metadata	0.23%	0.23%
Module: Module Load	45.98%	45.98%
Logon Session: Logon Session Creation	100.0%	96.57%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**60.0%**
Status	**Needs future improvements**
Sector Specific Priority	**37/100**
Overall Log Source Coverage	**45.54%**
Overall Log Collection Coverage	**35.7%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Multi-factor Authentication	Use multi-factor authentication on remote service logons where possible.
User Account Management	Limit the accounts that may use remote services. Limit the permissions for accounts that are at higher

	risk of compromise; for example, configure SSH so users can only run specific programs.
Implement Detection/Monitoring Capabilities	Correlate use of login activity related to remote services with unusual behavior or other malicious or suspicious activity. Adversaries will likely need to learn about an environment and the relationships between systems through Discovery techniques prior to attempting Lateral Movement. Use of applications such as ARD may be legitimate depending on the environment and how it's used. Other factors, such as access patterns and activity that occurs after a remote login, may indicate suspicious or malicious behavior using these applications. Monitor for user accounts logged into systems they would not normally access or access patterns to multiple systems over a relatively short period of time. In macOS, you can review logs for "screensharingd" and "Authentication" event messages. Monitor network connections regarding remote management (ports tcp:3283 and tcp:5900) and for remote login (port tcp:22).

4.2.10.2. *Taint Shared Content (T1080)*

Technique Information	
Technique ID	T1080
Technique Name	Taint Shared Content
Technique Description	Adversaries may deliver payloads to remote systems by adding content to shared storage locations, such as network drives or internal code repositories. Content stored on network drives or in other shared locations may be tainted by adding malicious programs, scripts, or exploit code to otherwise valid files. Once a user opens the shared tainted content, the malicious portion can be executed to run the adversary's code on a remote system. Adversaries may use tainted shared content to move laterally. A directory share pivot is a variation on this technique that uses several other techniques to propagate malware when users access a shared network directory. It uses [Shortcut Modification](T1547.009) of directory .LNK files that use [Masquerading](T1036) to look like the real directories, which are hidden through [Hidden Files and Directories](T1564.001). The malicious .LNK-based directories have an embedded command that executes the hidden malware file in the directory and then opens the real intended directory so that the user's expected action still occurs. When used with frequently used network directories, the technique may result in frequent reinfections and broad access to systems and potentially to new and higher privileged accounts. Adversaries may also compromise shared network directories through binary infections by appending or prepending its code to the healthy binary on the shared network directory. The malware may modify the original entry point (OEP) of the healthy binary to ensure that it is executed before the legitimate code. The infection could continue to spread via the newly infected file when it is executed by a remote system. These infections may target both binary and non-binary formats that end with extensions including, but not limited to, .EXE, .DLL, .SCR, .BAT, and/or .VBS.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Share: Network Share Access	22.81%	14.03%
File: File Creation	72.99%	57.47%
Process: Process Creation	39.25%	37.38%
File: File Modification	47.8%	34.23%

Technique Analysis	
Overall Score	**22.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**78/100**
Overall Log Source Coverage	**45.71%**
Overall Log Collection Coverage	**35.78%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Execution Prevention	Identify potentially malicious software that may be used to taint content or may result from it and audit and/or block the unknown programs by using application control
Exploit Protection	Use utilities that detect or mitigate common features used in exploitation, such as the Microsoft Enhanced Mitigation Experience Toolkit (EMET).

Restrict File and Directory Permissions	Protect shared folders by minimizing users who have write access.
Implement Detection/Monitoring Capabilities	Processes that write or overwrite many files to a network shared directory may be suspicious. Monitor processes that are executed from removable media for malicious or abnormal activity such as network connections due to Command and Control and possible network Discovery techniques. Frequently scan shared network directories for malicious files, hidden files, .LNK files, and other file types that may not typical exist in directories used to share specific types of content.

4.2.10.3. Exploitation of Remote Services (T1210)

Technique Information	
Technique ID	T1210
Technique Name	Exploitation of Remote Services
Technique Description	Adversaries may exploit remote services to gain unauthorized access to internal systems once inside of a network. Exploitation of a software vulnerability occurs when an adversary takes advantage of a programming error in a program, service, or within the operating system software or kernel itself to execute adversary-controlled code. A common goal for post-compromise exploitation of remote services is for lateral movement to enable access to a remote system. An adversary may need to determine if the remote system is in a vulnerable state, which may be done through [Network Service Discovery](T1046) or other Discovery methods looking for common, vulnerable software that may be deployed in the network, the lack of certain patches that may indicate vulnerabilities, or security software that may be used to detect or contain remote exploitation. Servers are likely a high value target for lateral movement exploitation, but endpoint systems may also be at risk if they provide an advantage or access to additional resources. There are several well-known vulnerabilities that exist in common services such as SMB and RDP as well as applications that may be used within internal networks such as MySQL and web server services. Depending on the permissions level of the vulnerable remote service an adversary may achieve [Exploitation for Privilege Escalation](T1068) as a result of lateral movement exploitation as well.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**

Network Traffic: Network Traffic Content	64.66%	0.0%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**65.66%**
Overall Log Collection Coverage	**16.67%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Application Isolation and Sandboxing	Make it difficult for adversaries to advance their operation through exploitation of undiscovered or unpatched vulnerabilities by using sandboxing. Other types of virtualization and application microsegmentation may also mitigate the impact of some types of exploitation. Risks of additional exploits and weaknesses in these systems may still exist.
Disable or Remove Feature or Program	Minimize available services to only those that are necessary.
Exploit Protection	Security applications that look for behavior used during exploitation such as Windows Defender Exploit Guard (WDEG) and the Enhanced Mitigation

	Experience Toolkit (EMET) can be used to mitigate some exploitation behavior.
Network Segmentation	Segment networks and systems appropriately to reduce access to critical systems and services to controlled methods.
Privileged Account Management	Minimize permissions and access for service accounts to limit impact of exploitation.
Threat Intelligence Program	Develop a robust cyber threat intelligence capability to determine what types and levels of threat may use software exploits and 0-days against a particular organization.
Update Software	Update software regularly by employing patch management for internal enterprise endpoints and servers.
Vulnerability Scanning	Regularly scan the internal network for available services to identify new and potentially vulnerable services.
Implement Detection/Monitoring Capabilities	Detecting software exploitation may be difficult depending on the tools available. Software exploits may not always succeed or may cause the exploited process to become unstable or crash. Also look for behavior on the endpoint system that might indicate successful compromise, such as abnormal behavior of the processes. This could include suspicious files written to disk, evidence of [Process Injection](T1055) for attempts to hide execution, evidence of [Discovery](TA0007), or other unusual network traffic that may indicate additional tools transferred to the system.

4.2.10.4. *Internal Spearphishing (T1534)*

Technique Information	
Technique ID	T1534
Technique Name	Internal Spearphishing
Technique Description	Adversaries may use internal spearphishing to gain access to additional information or exploit other users within the same organization after they already have access to accounts or systems within the environment. Internal spearphishing is multi-staged campaign where an email account is owned either by controlling the user's device with previously installed malware or by compromising the account credentials of the user. Adversaries attempt to take advantage of a trusted internal account to increase the likelihood of tricking the target into falling for the phish attempt. Adversaries may leverage [Spearphishing Attachment](T1566.001) or [Spearphishing Link](T1566.002) as part of internal spearphishing to deliver a payload or redirect to an external site to capture credentials through [Input Capture](T1056) on sites that mimic email login interfaces. There have been notable incidents where internal spearphishing has been used. The Eye Pyramid campaign used phishing emails with malicious attachments for lateral movement between victims, compromising nearly 18,000 email accounts in the process. The Syrian Electronic Army (SEA) compromised email accounts at the Financial Times (FT) to steal additional account credentials. Once FT learned of the campaign and began warning employees of the threat, the SEA sent phishing emails mimicking the Financial Times IT department and were able to compromise even more users.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%

Network Traffic: Network Traffic Flow	64.66%	0.0%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**20.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**65.33%**
Overall Log Collection Coverage	**11.11%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Network intrusion detection systems and email gateways usually do not scan internal email, but an organization can leverage the journaling-based solution which sends a copy of emails to a security service for offline analysis or incorporate service-integrated solutions using on-premise or API-based integrations to help detect internal spearphishing campaigns.

4.2.10.5. Remote Service Session Hijacking (T1563)

Technique Information	
Technique ID	T1563
Technique Name	Remote Service Session Hijacking
Technique Description	Adversaries may take control of preexisting sessions with remote services to move laterally in an environment. Users may use valid credentials to log into a service specifically designed to accept remote connections, such as telnet, SSH, and RDP. When a user logs into a service, a session will be established that will allow them to maintain a continuous interaction with that service. Adversaries may commandeer these sessions to carry out actions on remote systems. [Remote Service Session Hijacking](T1563) differs from use of [Remote Services](T1021) because it hijacks an existing session rather than creating a new session using [Valid Accounts](T1078).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Traffic: Network Traffic Content	64.66%	0.0%
Logon Session: Logon Session Creation	100.0%	96.57%
Process: Process Creation	39.25%	37.38%

Technique Analysis

Overall Score	**27.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**30/100**
Overall Log Source Coverage	**62.91%**
Overall Log Collection Coverage	**35.99%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Disable or Remove Feature or Program	Disable the remote service (ex: SSH, RDP, etc.) if it is unnecessary.
Network Segmentation	Enable firewall rules to block unnecessary traffic between network security zones within a network.
Privileged Account Management	Do not allow remote access to services as a privileged account unless necessary.
User Account Management	Limit remote user permissions if remote access is necessary.
Implement Detection/Monitoring Capabilities	Use of these services may be legitimate, depending upon the network environment and how it is used. Other factors, such as access patterns and activity that occurs after a remote login, may indicate suspicious or malicious behavior with that service. Monitor for user accounts logged into systems they would not normally access or access patterns to multiple systems over a relatively short period of time. Monitor for processes and command-line arguments associated with hijacking service sessions.

4.2.10.6. Lateral Tool Transfer (T1570)

Technique Information	
Technique ID	T1570
Technique Name	Lateral Tool Transfer
Technique Description	Adversaries may transfer tools or other files between systems in a compromised environment. Once brought into the victim environment (i.e. [Ingress Tool Transfer](T1105)) files may then be copied from one system to another to stage adversary tools or other files over the course of an operation. Adversaries may copy files between internal victim systems to support lateral movement using inherent file sharing protocols such as file sharing over [SMB/Windows Admin Shares](T1021.002) to connected network shares or with authenticated connections via [Remote Desktop Protocol](T1021.001). Files can also be transferred using native or otherwise present tools on the victim system, such as scp, rsync, curl, sftp, and [ftp](S0095).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
File: File Metadata	67.37%	55.82%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Share: Network Share Access	22.81%	14.03%
Named Pipe: Named Pipe Metadata	100.0%	100.0%

Network Traffic: Network Traffic Content	64.66%	0.0%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**49.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**51/100**
Overall Log Source Coverage	**59.71%**
Overall Log Collection Coverage	**38.83%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Filter Network Traffic	Consider using the host firewall to restrict file sharing communications such as SMB.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware or unusual data transfer over known tools and protocols like FTP can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions.
Implement Detection/Monitoring Capabilities	Monitor for file creation and files transferred within a network using protocols such as SMB or FTP. Unusual processes with internal network

	connections creating files on-system may be suspicious. Consider monitoring for abnormal usage of utilities and command-line arguments that may be used in support of remote transfer of files. Considering monitoring for alike file hashes or characteristics (ex: filename) that are created on multiple hosts.

4.2.11. Collection

4.2.11.1. Data from Local System (T1005)

Technique Information	
Technique ID	T1005
Technique Name	Data from Local System
Technique Description	Adversaries may search local system sources, such as file systems and configuration files or local databases, to find files of interest and sensitive data prior to Exfiltration. Adversaries may do this using a [Command and Scripting Interpreter](T1059), such as [cmd](S0106) as well as a [Network Device CLI](T1059.008), which have functionality to interact with the file system to gather information. Adversaries may also use [Automated Collection](T1119) on the local system.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Command: Command Execution	45.98%	45.98%
File: File Access	37.14%	29.99%

Technique Analysis	
Overall Score	**45.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**55/100**
Overall Log Source Coverage	**27.71%**
Overall Log Collection Coverage	**25.32%**

Detection Capability Present	**Yes**
Detection Sources	• ESET Antivirus

Mitigations	
Name	**Description**
Data Loss Prevention	Data loss prevention can restrict access to sensitive data and detect sensitive data that is unencrypted.
Implement Detection/Monitoring Capabilities	Monitor processes and command-line arguments for actions that could be taken to collect files from a system. Remote access tools with built-in features may interact directly with the Windows API to gather data. Further, [Network Device CLI](T1059.008) commands may also be used to collect files such as configuration files with built-in features native to the network device platform. Monitor CLI activity for unexpected or unauthorized use commands being run by non-standard users from non-standard locations. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.11.2. Data from Removable Media (T1025)

Technique Information	
Technique ID	T1025
Technique Name	Data from Removable Media
Technique Description	Adversaries may search connected removable media on computers they have compromised to find files of interest. Sensitive data can be collected from any removable media (optical disk drive, USB memory, etc.) connected to the compromised system prior to Exfiltration. Interactive command shells may be in use, and common functionality within [cmd](S0106) may be used to gather information. Some adversaries may also use [Automated Collection](T1119) on removable media.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%

Technique Analysis	
Overall Score	**52.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**20/100**
Overall Log Source Coverage	**41.56%**
Overall Log Collection Coverage	**37.98%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • Sentinel

Mitigations	
Name	**Description**
Data Loss Prevention	Data loss prevention can restrict access to sensitive data and detect sensitive data that is unencrypted.
Implement Detection/Monitoring Capabilities	Monitor processes and command-line arguments for actions that could be taken to collect files from a system's connected removable media. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.11.3. Data from Network Shared Drive (T1039)

Technique Information	
Technique ID	T1039
Technique Name	Data from Network Shared Drive
Technique Description	Adversaries may search network shares on computers they have compromised to find files of interest. Sensitive data can be collected from remote systems via shared network drives (host shared directory, network file server, etc.) that are accessible from the current system prior to Exfiltration. Interactive command shells may be in use, and common functionality within [cmd](S0106) may be used to gather information.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Share: Network Share Access	22.81%	14.03%
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%

Technique Analysis	
Overall Score	**18.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**34/100**
Overall Log Source Coverage	**35.31%**
Overall Log Collection Coverage	**30.0%**
Detection Capability Present	**No**

Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor processes and command-line arguments for actions that could be taken to collect files from a network share. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001).

4.2.11.4. Data Staged (T1074)

Technique Information	
Technique ID	T1074
Technique Name	Data Staged
Technique Description	Adversaries may stage collected data in a central location or directory prior to Exfiltration. Data may be kept in separate files or combined into one file through techniques such as [Archive Collected Data](T1560). Interactive command shells may be used, and common functionality within [cmd](S0106) and bash may be used to copy data into a staging location. In cloud environments, adversaries may stage data within a particular instance or virtual machine before exfiltration. An adversary may [Create Cloud Instance](T1578.002) and stage data in that instance. Adversaries may choose to stage data from a victim network in a centralized location prior to Exfiltration to minimize the number of connections made to their C2 server and better evade detection.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%

Technique Analysis	
Overall Score	**64.0%**
Status	**Needs future improvements**

Sector Specific Priority	**15/100**
Overall Log Source Coverage	**52.04%**
Overall Log Collection Coverage	**44.48%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Processes that appear to be reading files from disparate locations and writing them to the same directory or file may be an indication of data being staged, especially if they are suspected of performing encryption or compression on the files, such as 7zip, RAR, ZIP, or zlib. Monitor publicly writeable directories, central locations, and commonly used staging directories (recycle bin, temp folders, etc.) to regularly check for compressed or encrypted data that may be indicative of staging. Monitor processes and command-line arguments for actions that could be taken to collect and combine files. Remote access tools with built-in features may interact directly with the Windows API to gather and copy to a location. Data may also be acquired and staged through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). Consider monitoring accesses and modifications to storage repositories (such as the Windows Registry), especially from suspicious processes that could be related to malicious data collection.

4.2.11.5. *Screen Capture (T1113)*

Technique Information	
Technique ID	T1113
Technique Name	Screen Capture
Technique Description	Adversaries may attempt to take screen captures of the desktop to gather information over the course of an operation. Screen capturing functionality may be included as a feature of a remote access tool used in post-compromise operations. Taking a screenshot is also typically possible through native utilities or API calls, such as \`CopyFromScreen\`, \`xwd\`, or \`screencapture\`.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Technique Analysis	
Overall Score	**64.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**47.93%**
Overall Log Collection Coverage	**47.07%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitoring for screen capture behavior will depend on the method used to obtain data from the operating system and write output files. Detection methods could include collecting information from unusual processes using API calls used to obtain image data, and monitoring for image files written to disk. The sensor data may need to be correlated with other events to identify malicious activity, depending on the legitimacy of this behavior within a given network environment.

4.2.11.6. Email Collection (T1114)

Technique Information	
Technique ID	T1114
Technique Name	Email Collection
Technique Description	Adversaries may target user email to collect sensitive information. Emails may contain sensitive data, including trade secrets or personal information, that can prove valuable to adversaries. Adversaries can collect or forward email from mail servers or clients.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Connection Creation	45.45%	45.45%
File: File Access	37.14%	29.99%
Logon Session: Logon Session Creation	100.0%	96.57%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**67.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**14/100**
Overall Log Source Coverage	**59.05%**
Overall Log Collection Coverage	**50.26%**

Detection Capability Present	Yes
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Audit	Enterprise email solutions have monitoring mechanisms that may include the ability to audit auto-forwarding rules on a regular basis.
Encrypt Sensitive Information	Use of encryption provides an added layer of security to sensitive information sent over email. Encryption using public key cryptography requires the adversary to obtain the private certificate along with an encryption key to decrypt messages.
Multi-factor Authentication	Use of multi-factor authentication for public-facing webmail servers is a recommended best practice to minimize the usefulness of usernames and passwords to adversaries.
Implement Detection/Monitoring Capabilities	There are likely a variety of ways an adversary could collect email from a target, each with a different mechanism for detection. File access of local system email files for Exfiltration, unusual processes connecting to an email server within a network, or unusual access patterns or authentication attempts on a public-facing webmail server may all be indicators of malicious activity. Monitor processes and command-line arguments for actions that could be taken to gather local email files. Remote access tools with built-in features may interact directly with the Windows API to gather information. Information may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001). Detection is challenging because all messages

	forwarded because of an auto-forwarding rule have the same presentation as a manually forwarded message. It is also possible for the user to not be aware of the addition of such an auto-forwarding rule and not suspect that their account has been compromised; email-forwarding rules alone will not affect the normal usage patterns or operations of the email account. Auto-forwarded messages generally contain specific detectable artifacts that may be present in the header; such artifacts would be platform-specific. Examples include `X-MS-Exchange-Organization-AutoForwarded` set to true, `X-MailFwdBy` and `X-Forwarded-To`. The `forwardingSMTPAddress` parameter used in a forwarding process that is managed by administrators and not by user actions. All messages for the mailbox are forwarded to the specified SMTP address. However, unlike typical client-side rules, the message does not appear as forwarded in the mailbox; it appears as if it were sent directly to the specified destination mailbox. High volumes of emails that bear the `X-MS-Exchange-Organization-AutoForwarded` header (indicating auto-forwarding) without a corresponding number of emails that match the appearance of a forwarded message may indicate that further investigation is needed at the administrator level rather than user-level.

4.2.11.7. Clipboard Data (T1115)

Technique Information	
Technique ID	T1115
Technique Name	Clipboard Data
Technique Description	Adversaries may collect data stored in the clipboard from users copying information within or between applications. In Windows, Applications can access clipboard data by using the Windows API. OSX provides a native command, `pbpaste`, to grab clipboard contents.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Technique Analysis	
Overall Score	**64.0%**
Status	**Needs future improvements**
Sector Specific Priority	**15/100**
Overall Log Source Coverage	**47.93%**
Overall Log Collection Coverage	**47.07%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations

Name	Description
Implement Detection/Monitoring Capabilities	Access to the clipboard is a legitimate function of many applications on an operating system. If an organization chooses to monitor for this behavior, then the data will likely need to be correlated against other suspicious or non-user-driven activity.

4.2.11.8. Automated Collection (T1119)

Technique Information	
Technique ID	T1119
Technique Name	Automated Collection
Technique Description	Once established within a system or network, an adversary may use automated techniques for collecting internal data. Methods for performing this technique could include use of a [Command and Scripting Interpreter](T1059) to search for and copy information fitting set criteria such as file type, location, or name at specific time intervals. In cloud-based environments, adversaries may also use cloud APIs, command line interfaces, or extract, transform, and load (ETL) services to automatically collect data. This functionality could also be built into remote access tools. This technique may incorporate use of other techniques such as [File and Directory Discovery](T1083) and [Lateral Tool Transfer](T1570) to identify and move files, as well as [Cloud Service Dashboard](T1538) and [Cloud Storage Object Discovery](T1619) to identify resources in cloud environments.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
Script: Script Execution	0.0%	0.0%

Technique Analysis	
Overall Score	**52.0%**
Status	**Needs imminent remediation**

Sector Specific Priority	**48/100**
Overall Log Source Coverage	**27.71%**
Overall Log Collection Coverage	**25.32%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Encrypt Sensitive Information	Encryption and off-system storage of sensitive information may be one way to mitigate collection of files, but may not stop an adversary from acquiring the information if an intrusion persists over a long period of time and the adversary is able to discover and access the data through other means. Strong passwords should be used on certain encrypted documents that use them to prevent offline cracking through
Remote Data Storage	Encryption and off-system storage of sensitive information may be one way to mitigate collection of files, but may not stop an adversary from acquiring the information if an intrusion persists over a long period of time and the adversary is able to discover and access the data through other means.
Implement Detection/Monitoring Capabilities	Depending on the method used, actions could include common file system commands and parameters on the command-line interface within batch files or scripts. A sequence of actions like this may be unusual, depending on the system and network environment. Automated collection may occur along with other techniques such as [Data Staged](T1074). As such, file access monitoring that shows an unusual process performing sequential file opens and potentially copy actions to another location on the file system for many files at once

	may indicate automated collection behavior. Remote access tools with built-in features may interact directly with the Windows API to gather data. Data may also be acquired through Windows system management tools such as [Windows Management Instrumentation](T1047) and [PowerShell](T1059.001), as well as through cloud APIs and command line interfaces.

4.2.11.9. Audio Capture (T1123)

Technique Information	
Technique ID	T1123
Technique Name	Audio Capture
Technique Description	An adversary can leverage a computer's peripheral devices (e.g., microphones and webcams) or applications (e.g., voice and video call services) to capture audio recordings for the purpose of listening into sensitive conversations to gather information. Malware or scripts may be used to interact with the devices through an available API provided by the operating system or an application to capture audio. Audio files may be written to disk and exfiltrated later.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**74/100**
Overall Log Source Coverage	**47.93%**
Overall Log Collection Coverage	**47.07%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Detection of this technique may be difficult due to the various APIs that may be used. Telemetry data regarding API use may not be useful depending on how a system is normally used, but may provide context to other potentially malicious activity occurring on a system. Behavior that could indicate technique use include an unknown or unusual process accessing APIs associated with devices or software that interact with the microphone, recording devices, or recording software, and a process periodically writing files to disk that contain audio data.

4.2.11.10. Video Capture (T1125)

Technique Information	
Technique ID	T1125
Technique Name	Video Capture
Technique Description	An adversary can leverage a computer's peripheral devices (e.g., integrated cameras or webcams) or applications (e.g., video call services) to capture video recordings for the purpose of gathering information. Images may also be captured from devices or applications, potentially in specified intervals, in lieu of video files. Malware or scripts may be used to interact with the devices through an available API provided by the operating system or an application to capture video or images. Video or image files may be written to disk and exfiltrated later. This technique differs from [Screen Capture](T1113) due to use of specific devices or applications for video recording rather than capturing the victim's screen. In macOS, there are a few different malware samples that record the user's webcam such as FruitFly and Proton.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%

Technique Analysis	
Overall Score	**26.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**74/100**

Overall Log Source Coverage	**47.93%**
Overall Log Collection Coverage	**47.07%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Detection of this technique may be difficult due to the various APIs that may be used. Telemetry data regarding API use may not be useful depending on how a system is normally used, but may provide context to other potentially malicious activity occurring on a system. Behavior that could indicate technique use include an unknown or unusual process accessing APIs associated with devices or software that interact with the video camera, recording devices, or recording software, and a process periodically writing files to disk that contain video or camera image data.

4.2.11.11. Browser Session Hijacking (T1185)

Technique Information	
Technique ID	T1185
Technique Name	Browser Session Hijacking
Technique Description	Adversaries may take advantage of security vulnerabilities and inherent functionality in browser software to change content, modify user-behaviors, and intercept information as part of various browser session hijacking techniques. A specific example is when an adversary injects software into a browser that allows them to inherit cookies, HTTP sessions, and SSL client certificates of a user then use the browser as a way to pivot into an authenticated intranet. Executing browser-based behaviors such as pivoting may require specific process permissions, such as \`SeDebugPrivilege\` and/or high-integrity/administrator rights. Another example involves pivoting browser traffic from the adversary's browser through the user's browser by setting up a proxy which will redirect web traffic. This does not alter the user's traffic in any way, and the proxy connection can be severed as soon as the browser is closed. The adversary assumes the security context of whichever browser process the proxy is injected into. Browsers typically create a new process for each tab that is opened and permissions and certificates are separated accordingly. With these permissions, an adversary could potentially browse to any resource on an intranet, such as [Sharepoint](T1213.002) or webmail, that is accessible through the browser and which the browser has sufficient permissions. Browser pivoting may also bypass security provided by 2-factor authentication.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Process: Process Access	45.98%	45.98%

Process: Process Modification	42.27%	41.24%
Logon Session: Logon Session Creation	100.0%	96.57%

Technique Analysis	
Overall Score	**34.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**28/100**
Overall Log Source Coverage	**62.75%**
Overall Log Collection Coverage	**61.26%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
User Account Management	Since browser pivoting requires a high integrity process to launch from, restricting user permissions and addressing Privilege Escalation and
User Training	Close all browser sessions regularly and when they are no longer needed.
Implement Detection/Monitoring Capabilities	This may be a difficult technique to detect because adversary traffic may be masked by normal user traffic. New processes may not be created and no additional software dropped to disk. Authentication logs can be used to audit logins to specific web applications, but determining malicious logins versus benign logins may be difficult if activity matches typical user behavior. Monitor for [Process Injection](T1055) against browser applications.

4.2.11.12. Data from Information Repositories (T1213)

Technique Information	
Technique ID	T1213
Technique Name	Data from Information Repositories
Technique Description	Adversaries may leverage information repositories to mine valuable information. Information repositories are tools that allow for storage of information, typically to facilitate collaboration or information sharing between users, and can store a wide variety of data that may aid adversaries in further objectives, or direct access to the target information. Adversaries may also abuse external sharing features to share sensitive documents with recipients outside of the organization. The following is a brief list of example information that may hold potential value to an adversary and may also be found on an information repository: * Policies, procedures, and standards * Physical / logical network diagrams * System architecture diagrams * Technical system documentation * Testing / development credentials * Work / project schedules * Source code snippets * Links to network shares and other internal resources Information stored in a repository may vary based on the specific instance or environment. Specific common information repositories include web-based platforms such as [Sharepoint](T1213.002) and [Confluence](T1213.001), specific services such as Code Repositories, IaaS databases, enterprise databases, and other storage infrastructure such as SQL Server.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**

Logon Session: Logon Session Creation	100.0%	96.57%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**40.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**25/100**
Overall Log Source Coverage	**83.33%**
Overall Log Collection Coverage	**64.95%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Audit	Consider periodic review of accounts and privileges for critical and sensitive repositories.
User Account Management	Enforce the principle of least-privilege. Consider implementing access control mechanisms that include both authentication and authorization.
User Training	Develop and publish policies that define acceptable information to be stored in repositories.
Implement Detection/Monitoring Capabilities	As information repositories generally have a considerably large user base, detection of malicious use can be non-trivial. At minimum, access to information repositories performed by privileged users (for example, Active Directory Domain, Enterprise, or Schema Administrators) should be closely monitored and alerted upon, as these types

	of accounts should generally not be used to access information repositories. If the capability exists, it may be of value to monitor and alert on users that are retrieving and viewing a large number of documents and pages; this behavior may be indicative of programmatic means being used to retrieve all data within the repository. In environments with high-maturity, it may be possible to leverage User-Behavioral Analytics (UBA) platforms to detect and alert on user based anomalies. The user access logging within Microsoft's SharePoint can be configured to report access to certain pages and documents. Sharepoint audit logging can also be configured to report when a user shares a resource. The user access logging within Atlassian's Confluence can also be configured to report access to certain pages and documents through AccessLogFilter. Additional log storage and analysis infrastructure will likely be required for more robust detection capabilities.

4.2.11.13. *Data from Cloud Storage Object (T1530)*

Technique Information	
Technique ID	T1530
Technique Name	Data from Cloud Storage Object
Technique Description	Adversaries may access data objects from improperly secured cloud storage. Many cloud service providers offer solutions for online data storage such as Amazon S3, Azure Storage, and Google Cloud Storage. These solutions differ from other storage solutions (such as SQL or Elasticsearch) in that there is no overarching application. Data from these solutions can be retrieved directly using the cloud provider's APIs. Solution providers typically offer security guides to help end users configure systems. Misconfiguration by end users is a common problem. There have been numerous incidents where cloud storage has been improperly secured (typically by unintentionally allowing public access by unauthenticated users or overly-broad access by all users), allowing open access to credit cards, personally identifiable information, medical records, and other sensitive information. Adversaries may also obtain leaked credentials in source repositories, logs, or other means as a way to gain access to cloud storage objects that have access permission controls.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Cloud Storage: Cloud Storage Access	14.29%	14.29%

Technique Analysis	
Overall Score	**39.0%**

Status	**Needs imminent remediation**
Sector Specific Priority	**25/100**
Overall Log Source Coverage	**14.29%**
Overall Log Collection Coverage	**14.29%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Audit	Frequently check permissions on cloud storage to ensure proper permissions are set to deny open or unprivileged access to resources.
Encrypt Sensitive Information	Encrypt data stored at rest in cloud storage.
Filter Network Traffic	Cloud service providers support IP-based restrictions when accessing cloud resources. Consider using IP allowlisting along with user account management to ensure that data access is restricted not only to valid users but only from expected IP ranges to mitigate the use of stolen credentials to access data.
Multi-factor Authentication	Consider using multi-factor authentication to restrict access to resources and cloud storage APIs.
Restrict File and Directory Permissions	Use access control lists on storage systems and objects.
User Account Management	Configure user permissions groups and roles for access to cloud storage.
Implement Detection/Monitoring Capabilities	Monitor for unusual queries to the cloud provider's storage service. Activity originating from unexpected sources may indicate improper permissions are set that is allowing access to data.

	Additionally, detecting failed attempts by a user for a certain object, followed by escalation of privileges by the same user, and access to the same object may be an indication of suspicious activity.

4.2.11.14. Archive Collected Data (T1560)

Technique Information	
Technique ID	T1560
Technique Name	Archive Collected Data
Technique Description	An adversary may compress and/or encrypt data that is collected prior to exfiltration. Compressing the data can help to obfuscate the collected data and minimize the amount of data sent over the network. Encryption can be used to hide information that is being exfiltrated from detection or make exfiltration less conspicuous upon inspection by a defender. Both compression and encryption are done prior to exfiltration, and can be performed using a utility, 3rd party library, or custom method.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Script: Script Execution	0.0%	0.0%
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**43.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**24/100**
Overall Log Source Coverage	**39.55%**
Overall Log Collection Coverage	**35.21%**

Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Audit	System scans can be performed to identify unauthorized archival utilities.
Implement Detection/Monitoring Capabilities	Archival software and archived files can be detected in many ways. Common utilities that may be present on the system or brought in by an adversary may be detectable through process monitoring and monitoring for command-line arguments for known archival utilities. This may yield a significant number of benign events, depending on how systems in the environment are typically used. A process that loads the Windows DLL crypt32.dll may be used to perform encryption, decryption, or verification of file signatures. Consider detecting writing of files with extensions and/or headers associated with compressed or encrypted file types. Detection efforts may focus on follow-on exfiltration activity, where compressed or encrypted files can be detected in transit with a network intrusion detection or data loss prevention system analyzing file headers.

4.2.11.15. *Data from Configuration Repository (T1602)*

Technique Information	
Technique ID	T1602
Technique Name	Data from Configuration Repository
Technique Description	Adversaries may collect data related to managed devices from configuration repositories. Configuration repositories are used by management systems in order to configure, manage, and control data on remote systems. Configuration repositories may also facilitate remote access and administration of devices. Adversaries may target these repositories in order to collect large quantities of sensitive system administration data. Data from configuration repositories may be exposed by various protocols and software and can store a wide variety of data, much of which may align with adversary Discovery objectives.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**55.05%**
Overall Log Collection Coverage	**22.73%**

Detection Capability Present	No
Detection Sources	-

Mitigations	
Name	**Description**
Encrypt Sensitive Information	Configure SNMPv3 to use the highest level of security (authPriv) available.
Filter Network Traffic	Apply extended ACLs to block unauthorized protocols outside the trusted network.
Network Intrusion Prevention	Configure intrusion prevention devices to detect SNMP queries and commands from unauthorized sources.
Network Segmentation	Segregate SNMP traffic on a separate management network.
Software Configuration	Allowlist MIB objects and implement SNMP views.
Update Software	Keep system images and software updated and migrate to SNMPv3.
Implement Detection/Monitoring Capabilities	Identify network traffic sent or received by untrusted hosts or networks that solicits and obtains the configuration information of the queried device.

4.2.12. Command and Control

4.2.12.1. Data Obfuscation (T1001)

Technique Information	
Technique ID	T1001
Technique Name	Data Obfuscation
Technique Description	Adversaries may obfuscate command and control traffic to make it more difficult to detect. Command and control (C2) communications are hidden (but not necessarily encrypted) in an attempt to make the content more difficult to discover or decipher and to make the communication less conspicuous and hide commands from being seen. This encompasses many methods, such as adding junk data to protocol traffic, using steganography, or impersonating legitimate protocols.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**16.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**35/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate some obfuscation activity at the network level.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.12.2. *Fallback Channels (T1008)*

Technique Information	
Technique ID	T1008
Technique Name	Fallback Channels
Technique Description	Adversaries may use fallback or alternate communication channels if the primary channel is compromised or inaccessible in order to maintain reliable command and control and to avoid data transfer thresholds.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**79/100**
Overall Log Source Coverage	**55.05%**
Overall Log Collection Coverage	**22.73%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations

Name	Description
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.12.3. *Application Layer Protocol (T1071)*

Technique Information	
Technique ID	T1071
Technique Name	Application Layer Protocol
Technique Description	Adversaries may communicate using application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**61.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**

Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect application layer protocols that do not follow the expected protocol standards regarding syntax, structure, or any other variable adversaries could leverage to conceal data.

4.2.12.4. Proxy (T1090)

Technique Information	
Technique ID	T1090
Technique Name	Proxy
Technique Description	Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command and control server to avoid direct connections to their infrastructure. Many tools exist that enable traffic redirection through proxies or port redirection, including [HTRAN](S0040), ZXProxy, and ZXPortMap. Adversaries use these types of proxies to manage command and control communications, reduce the number of simultaneous outbound network connections, provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion. Adversaries may chain together multiple proxies to further disguise the source of malicious traffic. Adversaries can also take advantage of routing schemes in Content Delivery Networks (CDNs) to proxy command and control traffic.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis

Overall Score	**64.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**58.25%**
Overall Log Collection Coverage	**15.15%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Filter Network Traffic	Traffic to known anonymity networks and C2 infrastructure can be blocked through the use of network allow and block lists. It should be noted that this kind of blocking may be circumvented by other techniques like
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific C2 protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools.
SSL/TLS Inspection	If it is possible to inspect HTTPS traffic, the captures can be analyzed for connections that appear to be domain fronting.

Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server or between clients that should not or often do not communicate with one another). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. Consider monitoring for traffic to known anonymity networks (such as [Tor](S0183)).

4.2.12.5. Communication Through Removable Media (T1092)

Technique Information	
Technique ID	T1092
Technique Name	Communication Through Removable Media
Technique Description	Adversaries can perform command and control between compromised hosts on potentially disconnected networks using removable media to transfer commands from system to system. Both systems would need to be compromised, with the likelihood that an Internet-connected system was compromised first and the second through lateral movement by [Replication Through Removable Media](T1091). Commands and files would be relayed from the disconnected system to the Internet-connected system to which the adversary has direct access.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Drive: Drive Access	45.45%	45.45%
Drive: Drive Creation	45.45%	45.45%

Technique Analysis	
Overall Score	**25.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**45.45%**
Overall Log Collection Coverage	**45.45%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Disable or Remove Feature or Program	Disable Autoruns if it is unnecessary.
Operating System Configuration	Disallow or restrict removable media at an organizational policy level if they are not required for business operations.
Implement Detection/Monitoring Capabilities	Monitor file access on removable media. Detect processes that execute when removable media is mounted.

4.2.12.6. Non-Application Layer Protocol (T1095)

Technique Information	
Technique ID	T1095
Technique Name	Non-Application Layer Protocol
Technique Description	Adversaries may use a non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive. Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL). ICMP communication between hosts is one example. Because ICMP is part of the Internet Protocol Suite, it is required to be implemented by all IP-compatible hosts. However, it is not as commonly monitored as other Internet Protocols such as TCP or UDP and may be used by adversaries to hide communications.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**61.0%**
Status	**Needs future improvements**

Sector Specific Priority	**16/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Filter Network Traffic	Filter network traffic to prevent use of protocols across the network boundary that are unnecessary.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
Network Segmentation	Properly configure firewalls and proxies to limit outgoing traffic to only necessary ports and through proper network gateway systems. Also ensure hosts are only provisioned to communicate over authorized interfaces.
Implement Detection/Monitoring Capabilities	Analyze network traffic for ICMP messages or other protocols that contain abnormal data or are not normally seen within or exiting the network. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. Monitor and investigate API calls to functions

	associated with enabling and/or utilizing alternative communication channels.

4.2.12.7. *Web Service (T1102)*

Technique Information	
Technique ID	T1102
Technique Name	Web Service
Technique Description	Adversaries may use an existing, legitimate external Web service as a means for relaying data to/from a compromised system. Popular websites and social media acting as a mechanism for C2 may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to a compromise. Using common services, such as those offered by Google or Twitter, makes it easier for adversaries to hide in expected noise. Web service providers commonly use SSL/TLS encryption, giving adversaries an added level of protection. Use of Web services may also protect back-end C2 infrastructure from discovery through malware binary analysis while also enabling operational resiliency (since this infrastructure may be dynamically changed).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**64.0%**

Status	**Needs future improvements**
Sector Specific Priority	**15/100**
Overall Log Source Coverage	**58.25%**
Overall Log Collection Coverage	**15.15%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
Restrict Web-Based Content	Web proxies can be used to enforce external network communication policy that prevents use of unauthorized external services.
Implement Detection/Monitoring Capabilities	Host data that can relate unknown or suspicious process activity using a network connection is important to supplement any existing indicators of compromise based on malware command and control signatures and infrastructure or the presence of strong encryption. Packet capture analysis will require SSL/TLS inspection if data is encrypted. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). User behavior monitoring may help to detect abnormal patterns of activity.

4.2.12.8. Multi-Stage Channels (T1104)

Technique Information	
Technique ID	T1104
Technique Name	Multi-Stage Channels
Technique Description	Adversaries may create multiple stages for command and control that are employed under different conditions or for certain functions. Use of multiple stages may obfuscate the command and control channel to make detection more difficult. Remote access tools will call back to the first-stage command and control server for instructions. The first stage may have automated capabilities to collect basic host information, update tools, and upload additional files. A second remote access tool (RAT) could be uploaded at that point to redirect the host to the second-stage command and control server. The second stage will likely be more fully featured and allow the adversary to interact with the system through a reverse shell and additional RAT features. The different stages will likely be hosted separately with no overlapping infrastructure. The loader may also have backup first-stage callbacks or [Fallback Channels](T1008) in case the original first-stage communication path is discovered and blocked.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis

Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**55.05%**
Overall Log Collection Coverage	**22.73%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
Implement Detection/Monitoring Capabilities	Host data that can relate unknown or suspicious process activity using a network connection is important to supplement any existing indicators of compromise based on malware command and control signatures and infrastructure. Relating subsequent actions that may result from Discovery of the system and network information or Lateral Movement to the originating process may also yield useful data.

4.2.12.9. Ingress Tool Transfer (T1105)

Technique Information	
Technique ID	T1105
Technique Name	Ingress Tool Transfer
Technique Description	Adversaries may transfer tools or other files from an external system into a compromised environment. Tools or files may be copied from an external adversary-controlled system to the victim network through the command and control channel or through alternate protocols such as [ftp](S0095). Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. [Lateral Tool Transfer](T1570)). Files can also be transferred using various [Web Service](T1102)s as well as native or otherwise present tools on the victim system. On Windows, adversaries may use various utilities to download tools, such as \`copy\`, \`finger\`, and [PowerShell](T1059.001) commands such as \`IEX(New-Object Net.WebClient).downloadString()\` and \`Invoke-WebRequest\`. On Linux and macOS systems, a variety of utilities also exist, such as \`curl\`, \`scp\`, \`sftp\`, \`tftp\`, \`rsync\`, \`finger\`, and \`wget\`.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Network Traffic: Network Traffic Content	64.66%	0.0%
File: File Creation	72.99%	57.47%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**61.0%**
Status	**Needs future improvements**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**61.94%**
Overall Log Collection Coverage	**25.73%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware or unusual data transfer over known protocols like FTP can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools.
Implement Detection/Monitoring Capabilities	Monitor for file creation and files transferred into the network. Unusual processes with external network connections creating files on-system may be suspicious. Use of utilities, such as [ftp](S0095), that does not normally occur may also be suspicious. Analyze network data for uncommon data flows

	(e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Specifically, for the finger utility on Windows and Linux systems, monitor command line or terminal execution for the finger command. Monitor network activity for TCP port 79, which is used by the finger utility, and Windows `netsh interface portproxy` modifications to well-known ports such as 80 and 443. Furthermore, monitor file system for the download/creation and execution of suspicious files, which may indicate adversary-downloaded payloads. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.12.10. *Data Encoding (T1132)*

Technique Information	
Technique ID	T1132
Technique Name	Data Encoding
Technique Description	Adversaries may encode data to make the content of command and control traffic more difficult to detect. Command and control (C2) information can be encoded using a standard data encoding system. Use of data encoding may adhere to existing protocol specifications and includes use of ASCII, Unicode, Base64, MIME, or other binary-to-text and character encoding systems. Some data encoding systems may also result in data compression, such as gzip.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**16.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**35/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.12.11. Remote Access Software (T1219)

Technique Information	
Technique ID	T1219
Technique Name	Remote Access Software
Technique Description	An adversary may use legitimate desktop support and remote access software, such as Team Viewer, AnyDesk, Go2Assist, LogMein, AmmyyAdmin, etc, to establish an interactive command and control channel to target systems within networks. These services are commonly used as legitimate technical support software, and may be allowed by application control within a target environment. Remote access tools like VNC, Ammyy, and Teamviewer are used frequently when compared with other legitimate software commonly used by adversaries. Remote access tools may be installed and used post-compromise as alternate communications channel for redundant access or as a way to establish an interactive remote desktop session with the target system. They may also be used as a component of malware to establish a reverse connection or back-connect to a service or adversary controlled system. Installation of many remote access tools may also include persistence (ex: the tool's installation routine creates a [Windows Service](T1543.003)). Admin tools such as TeamViewer have been used by several groups targeting institutions in countries of interest to the Russian state and criminal campaigns.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Network Traffic: Network Connection Creation	45.45%	45.45%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**65.0%**
Status	**Needs future improvements**
Sector Specific Priority	**15/100**
Overall Log Source Coverage	**53.5%**
Overall Log Collection Coverage	**20.71%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Execution Prevention	Use application control to mitigate installation and use of unapproved software that can be used for remote access.
Filter Network Traffic	Properly configure firewalls, application firewalls, and proxies to limit outgoing traffic to sites and services used by remote access tools.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures may be able to prevent traffic to remote access services.
Implement Detection/Monitoring Capabilities	Monitor for applications and processes related to remote admin tools. Correlate activity with other suspicious behavior that may reduce false positives if these tools are used by legitimate users and administrators.

	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect application layer protocols that do not follow the expected protocol for the port that is being used. [Domain Fronting](T1090.004) may be used in conjunction to avoid defenses. Adversaries will likely need to deploy and/or install these remote tools to compromised systems. It may be possible to detect or prevent the installation of these tools with host-based solutions.

4.2.12.12. *Dynamic Resolution (T1568)*

Technique Information	
Technique ID	T1568
Technique Name	Dynamic Resolution
Technique Description	Adversaries may dynamically establish connections to command and control infrastructure to evade common detections and remediations. This may be achieved by using malware that shares a common algorithm with the infrastructure the adversary uses to receive the malware's communications. These calculations can be used to dynamically adjust parameters such as the domain name, IP address, or port number the malware uses for command and control. Adversaries may use dynamic resolution for the purpose of [Fallback Channels](T1008). When contact is lost with the primary command and control server malware may employ dynamic resolution as a means to reestablishing command and control.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**60.0%**

Status	**Needs future improvements**
Sector Specific Priority	**17/100**
Overall Log Source Coverage	**58.25%**
Overall Log Collection Coverage	**15.15%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Malware researchers can reverse engineer malware variants that use dynamic resolution and determine future C2 infrastructure that the malware will attempt to contact, but this is a time and resource intensive effort.
Restrict Web-Based Content	In some cases a local DNS sinkhole may be used to help prevent behaviors associated with dynamic resolution.
Implement Detection/Monitoring Capabilities	Detecting dynamically generated C2 can be challenging due to the number of different algorithms, constantly evolving malware families, and the increasing complexity of the algorithms. There are multiple approaches to detecting a pseudo-randomly generated domain name, including using frequency analysis, Markov chains, entropy, proportion of dictionary words, ratio of vowels to other characters, and more . CDN domains may trigger these detections due to the format of their domain names. In addition to detecting algorithm generated domains based on the name, another more general approach for detecting a

	suspicious domain is to check for recently registered names or for rarely visited domains.

4.2.12.13. Non-Standard Port (T1571)

Technique Information	
Technique ID	T1571
Technique Name	Non-Standard Port
Technique Description	Adversaries may communicate using a protocol and port paring that are typically not associated. For example, HTTPS over port 8088 or port 587 as opposed to the traditional port 443. Adversaries may make changes to the standard port used by a protocol to bypass filtering or muddle analysis/parsing of network data.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**33.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**28/100**
Overall Log Source Coverage	**58.25%**
Overall Log Collection Coverage	**15.15%**
Detection Capability Present	**Yes**

Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
Network Segmentation	Properly configure firewalls and proxies to limit outgoing traffic to only necessary ports for that particular network segment.
Implement Detection/Monitoring Capabilities	Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used. Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious.

4.2.12.14. Protocol Tunneling (T1572)

Technique Information	
Technique ID	T1572
Technique Name	Protocol Tunneling
Technique Description	Adversaries may tunnel network communications to and from a victim system within a separate protocol to avoid detection/network filtering and/or enable access to otherwise unreachable systems. Tunneling involves explicitly encapsulating a protocol within another. This behavior may conceal malicious traffic by blending in with existing traffic and/or provide an outer layer of encryption (similar to a VPN). Tunneling could also enable routing of network packets that would otherwise not reach their intended destination, such as SMB, RDP, or other traffic that would be filtered by network appliances or not routed over the Internet. There are various means to encapsulate a protocol within another protocol. For example, adversaries may perform SSH tunneling (also known as SSH port forwarding), which involves forwarding arbitrary data over an encrypted SSH tunnel. [Protocol Tunneling](T1572) may also be abused by adversaries during [Dynamic Resolution](T1568). Known as DNS over HTTPS (DoH), queries to resolve C2 infrastructure may be encapsulated within encrypted HTTPS packets. Adversaries may also leverage [Protocol Tunneling](T1572) in conjunction with [Proxy](T1090) and/or [Protocol Impersonation](T1001.003) to further conceal C2 communications and infrastructure.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%

Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**19.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**34/100**
Overall Log Source Coverage	**58.25%**
Overall Log Collection Coverage	**15.15%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Filter Network Traffic	Consider filtering network traffic to untrusted or known bad domains and resources.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
Implement Detection/Monitoring Capabilities	Monitoring for systems listening and/or establishing external connections using ports/protocols commonly associated with tunneling, such as SSH (port 22). Also monitor for processes commonly associated with tunneling, such as Plink and the OpenSSH client. Analyze network data for uncommon data flows

	(e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect application layer protocols that do not follow the expected protocol standards regarding syntax, structure, or any other variable adversaries could leverage to conceal data.

4.2.12.15. Encrypted Channel (T1573)

Technique Information	
Technique ID	T1573
Technique Name	Encrypted Channel
Technique Description	Adversaries may employ a known encryption algorithm to conceal command and control traffic rather than relying on any inherent protections provided by a communication protocol. Despite the use of a secure algorithm, these implementations may be vulnerable to reverse engineering if secret keys are encoded and/or generated within malware samples/configuration files.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**61.0%**
Status	**Needs future improvements**
Sector Specific Priority	**39/100**
Overall Log Source Coverage	**64.66%**
Overall Log Collection Coverage	**0.0%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender • FortiGate

Mitigations

Name	Description
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level.
SSL/TLS Inspection	SSL/TLS inspection can be used to see the contents of encrypted sessions to look for network-based indicators of malware communication protocols.
Implement Detection/Monitoring Capabilities	SSL/TLS inspection is one way of detecting command and control traffic within some encrypted communication channels. SSL/TLS inspection does come with certain risks that should be considered before implementing to avoid potential security issues such as incomplete certificate validation. In general, analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.13. Exfiltration

4.2.13.1. Exfiltration Over Other Network Medium (T1011)

Technique Information	
Technique ID	T1011
Technique Name	Exfiltration Over Other Network Medium
Technique Description	Adversaries may attempt to exfiltrate data over a different network medium than the command and control channel. If the command and control network is a wired Internet connection, the exfiltration may occur, for example, over a WiFi connection, modem, cellular data connection, Bluetooth, or another radio frequency (RF) channel. Adversaries may choose to do this if they have sufficient access or proximity, and the connection might not be secured or defended as well as the primary Internet-connected channel because it is not routed through the same enterprise network.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
File: File Access	37.14%	29.99%
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**20.0%**

Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**51.58%**
Overall Log Collection Coverage	**24.28%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Operating System Configuration	Prevent the creation of new network adapters where possible.
Implement Detection/Monitoring Capabilities	Monitor for processes utilizing the network that do not normally have network communication or have never been seen before. Processes that normally require user-driven events to access the network (for example, a web browser opening with a mouse click or key press) but access the network without such may be malicious. Monitor for and investigate changes to host adapter settings, such as addition and/or replication of communication interfaces.

4.2.13.2. *Automated Exfiltration (T1020)*

Technique Information	
Technique ID	T1020
Technique Name	Automated Exfiltration
Technique Description	Adversaries may exfiltrate data, such as sensitive documents, through the use of automated processing after being gathered during Collection. When automated exfiltration is used, other exfiltration techniques likely apply as well to transfer the information out of the network, such as [Exfiltration Over C2 Channel](T1041) and [Exfiltration Over Alternative Protocol](T1048).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Script: Script Execution	0.0%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
File: File Access	37.14%	29.99%
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**17.0%**
Status	**Needs immediate remediation**

Sector Specific Priority	**83/100**
Overall Log Source Coverage	**42.98%**
Overall Log Collection Coverage	**20.24%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Monitor process file access patterns and network behavior. Unrecognized processes or scripts that appear to be traversing file systems and sending network traffic may be suspicious.

4.2.13.3. Scheduled Transfer (T1029)

Technique Information	
Technique ID	T1029
Technique Name	Scheduled Transfer
Technique Description	Adversaries may schedule data exfiltration to be performed only at certain times of day or at certain intervals. This could be done to blend traffic patterns with normal activity or availability. When scheduled exfiltration is used, other exfiltration techniques likely apply as well to transfer the information out of the network, such as [Exfiltration Over C2 Channel](T1041) or [Exfiltration Over Alternative Protocol](T1048).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**55.05%**
Overall Log Collection Coverage	**22.73%**
Detection Capability Present	**No**

Detection Sources	-

Mitigations	
Name	**Description**
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary command and control infrastructure and malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools.
Implement Detection/Monitoring Capabilities	Monitor process file access patterns and network behavior. Unrecognized processes or scripts that appear to be traversing file systems and sending network traffic may be suspicious. Network connections to the same destination that occur at the same time of day for multiple days are suspicious.

4.2.13.4. Data Transfer Size Limits (T1030)

Technique Information	
Technique ID	T1030
Technique Name	Data Transfer Size Limits
Technique Description	An adversary may exfiltrate data in fixed size chunks instead of whole files or limit packet sizes below certain thresholds. This approach may be used to avoid triggering network data transfer threshold alerts.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Connection Creation	45.45%	45.45%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**21.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**33/100**
Overall Log Source Coverage	**55.05%**
Overall Log Collection Coverage	**22.73%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations

Name	Description
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary command and control infrastructure and malware can be used to mitigate activity at the network level.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). If a process maintains a long connection during which it consistently sends fixed size data packets or a process opens connections and sends fixed sized data packets at regular intervals, it may be performing an aggregate data transfer. Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.13.5. *Exfiltration Over C2 Channel (T1041)*

Technique Information	
Technique ID	T1041
Technique Name	Exfiltration Over C2 Channel
Technique Description	Adversaries may steal data by exfiltrating it over an existing command and control channel. Stolen data is encoded into the normal communications channel using the same protocol as command and control communications.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
File: File Access	37.14%	29.99%
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**65.0%**
Status	**Needs future improvements**
Sector Specific Priority	**35/100**
Overall Log Source Coverage	**51.58%**
Overall Log Collection Coverage	**24.28%**
Detection Capability Present	**Yes**

Detection Sources	• BitDefender • FortiGate

Mitigations	
Name	**Description**
Data Loss Prevention	Data loss prevention can detect and block sensitive data being sent over unencrypted protocols.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific obfuscation technique used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool command and control signatures over time or construct protocols in such a way to avoid detection by common defensive tools.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.13.6. *Exfiltration Over Alternative Protocol (T1048)*

Technique Information	
Technique ID	T1048
Technique Name	Exfiltration Over Alternative Protocol
Technique Description	Adversaries may steal data by exfiltrating it over a different protocol than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server. Alternate protocols include FTP, SMTP, HTTP/S, DNS, SMB, or any other network protocol not being used as the main command and control channel. Different protocol channels could also include Web services such as cloud storage. Adversaries may also opt to encrypt and/or obfuscate these alternate channels. [Exfiltration Over Alternative Protocol](T1048) can be done using various common operating system utilities such as [Net](S0039)/SMB or FTP. On macOS and Linux `curl` may be used to invoke protocols such as HTTP/S or FTP/S to exfiltrate data from a system.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Network Traffic: Network Connection Creation	45.45%	45.45%
File: File Access	37.14%	29.99%
Network Traffic: Network Traffic Content	64.66%	0.0%

Email: Message Trace	100.0%	100.0%
Email: Threat Protection	100.0%	100.0%

Technique Analysis	
Overall Score	**68.0%**
Status	**Could benefit from improvments**
Sector Specific Priority	**13/100**
Overall Log Source Coverage	**65.41%**
Overall Log Collection Coverage	**45.92%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Data Loss Prevention	Data loss prevention can detect and block sensitive data being uploaded via web browsers.
Filter Network Traffic	Enforce proxies and use dedicated servers for services such as DNS and only allow those systems to communicate over respective ports/protocols, instead of all systems within a network.
Network Intrusion Prevention	Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary command and control infrastructure and malware can be used to mitigate activity at the network level.
Network Segmentation	Follow best practices for network firewall configurations to allow only necessary ports and traffic to enter and exit the network.

Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. Analyze packet contents to detect communications that do not follow the expected protocol behavior for the port that is being used.

4.2.13.7. *Exfiltration Over Physical Medium (T1052)*

Technique Information	
Technique ID	T1052
Technique Name	Exfiltration Over Physical Medium
Technique Description	Adversaries may attempt to exfiltrate data via a physical medium, such as a removable drive. In certain circumstances, such as an air-gapped network compromise, exfiltration could occur via a physical medium or device introduced by a user. Such media could be an external hard drive, USB drive, cellular phone, MP3 player, or other removable storage and processing device. The physical medium or device could be used as the final exfiltration point or to hop between otherwise disconnected systems.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Drive: Drive Creation	45.45%	45.45%
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**53.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**20/100**
Overall Log Source Coverage	**41.96%**
Overall Log Collection Coverage	**39.7%**

Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Data Loss Prevention	Data loss prevention can detect and block sensitive data being copied to physical mediums.
Disable or Remove Feature or Program	Disable Autorun if it is unnecessary.
Limit Hardware Installation	Limit the use of USB devices and removable media within a network.
Implement Detection/Monitoring Capabilities	Monitor file access on removable media. Detect processes that execute when removable media are mounted.

4.2.13.8. *Transfer Data to Cloud Account (T1537)*

Technique Information	
Technique ID	T1537
Technique Name	Transfer Data to Cloud Account
Technique Description	Adversaries may exfiltrate data by transferring the data, including backups of cloud environments, to another cloud account they control on the same service to avoid typical file transfers/downloads and network-based exfiltration detection. A defender who is monitoring for large transfers to outside the cloud environment through normal file transfers or over command and control channels may not be watching for data transfers to another account within the same cloud provider. Such transfers may utilize existing cloud provider APIs and the internal address space of the cloud provider to blend into normal traffic or avoid data transfers over external network interfaces. Incidents have been observed where adversaries have created backups of cloud instances and transferred them to separate accounts.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Cloud Storage: Cloud Storage Creation	14.29%	14.29%
Snapshot: Snapshot Creation	42.3%	40.05%
Snapshot: Snapshot Modification	36.67%	34.54%
Cloud Storage: Cloud Storage Modification	14.29%	14.29%

Technique Analysis

Overall Score	**14.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**36/100**
Overall Log Source Coverage	**26.88%**
Overall Log Collection Coverage	**25.79%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Filter Network Traffic	Implement network-based filtering restrictions to prohibit data transfers to untrusted VPCs.
Password Policies	Consider rotating access keys within a certain number of days to reduce the effectiveness of stolen credentials.
User Account Management	Limit user account and IAM policies to the least privileges required. Consider using temporary credentials for accounts that are only valid for a certain period of time to reduce the effectiveness of compromised accounts.
Implement Detection/Monitoring Capabilities	Monitor account activity for attempts to share data, snapshots, or backups with untrusted or unusual accounts on the same cloud service provider. Monitor for anomalous file transfer activity between accounts and to untrusted VPCs. In AWS, sharing an Elastic Block Store (EBS) snapshot, either with specified users or publicly, generates a ModifySnapshotAttribute event in CloudTrail logs. Similarly, in Azure, creating a Shared Access Signature (SAS) URI for a Virtual Hard Disk

	(VHS) snapshot generates a "Get Snapshot SAS URL" event in Activity Logs.

4.2.13.9. *Exfiltration Over Web Service (T1567)*

Technique Information	
Technique ID	T1567
Technique Name	Exfiltration Over Web Service
Technique Description	Adversaries may use an existing, legitimate external Web service to exfiltrate data rather than their primary command and control channel. Popular Web services acting as an exfiltration mechanism may give a significant amount of cover due to the likelihood that hosts within a network are already communicating with them prior to compromise. Firewall rules may also already exist to permit traffic to these services. Web service providers also commonly use SSL/TLS encryption, giving adversaries an added level of protection.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
File: File Access	37.14%	29.99%
Command: Command Execution	45.98%	45.98%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**55.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**19/100**
Overall Log Source Coverage	**53.11%**

Overall Log Collection Coverage	**18.99%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Data Loss Prevention	Data loss prevention can be detect and block sensitive data being uploaded to web services via web browsers.
Restrict Web-Based Content	Web proxies can be used to enforce an external network communication policy that prevents use of unauthorized external services.
Implement Detection/Monitoring Capabilities	Analyze network data for uncommon data flows (e.g., a client sending significantly more data than it receives from a server). Processes utilizing the network that do not normally have network communication or have never been seen before are suspicious. User behavior monitoring may help to detect abnormal patterns of activity.

4.2.14. Impact

4.2.14.1. Data Destruction (T1485)

Technique Information	
Technique ID	T1485
Technique Name	Data Destruction
Technique Description	Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt availability to systems, services, and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files or data on local and remote drives. Common operating system file deletion commands such as `del` and `rm` often only remove pointers to files without wiping the contents of the files themselves, making the files recoverable by proper forensic methodology. This behavior is distinct from [Disk Content Wipe](T1561.001) and [Disk Structure Wipe](T1561.002) because individual files are destroyed rather than sections of a storage disk or the disk's logical structure. Adversaries may attempt to overwrite files and directories with randomly generated data to make it irrecoverable. In some cases politically oriented image files have been used to overwrite data. To maximize impact on the target organization in operations where network-wide availability interruption is the goal, malware designed for destroying data may have worm-like features to propagate across a network by leveraging additional techniques like [Valid Accounts](T1078), [OS Credential Dumping](T1003), and [SMB/Windows Admin Shares](T1021.002).. In cloud environments, adversaries may leverage access to delete cloud storage, cloud storage accounts, machine images, and other infrastructure crucial to operations to damage an organization or their customers.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Command: Command Execution	45.98%	45.98%
File: File Modification	47.8%	34.23%
Image: Image Deletion	0.0%	0.0%
File: File Deletion	72.99%	72.99%
Instance: Instance Deletion	100.0%	100.0%
Cloud Storage: Cloud Storage Deletion	14.29%	7.14%
Volume: Volume Deletion	100.0%	100.0%
Snapshot: Snapshot Deletion	42.3%	40.05%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**31.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**69/100**
Overall Log Source Coverage	**51.4%**
Overall Log Collection Coverage	**48.64%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Data Backup	Consider implementing IT disaster recovery plans that contain procedures for taking regular data

	backups that can be used to restore organizational data.
Implement Detection/Monitoring Capabilities	Use process monitoring to monitor the execution and command-line parameters of binaries that could be involved in data destruction activity, such as [SDelete](S0195). Monitor for the creation of suspicious files as well as high unusual file modification activity. In particular, look for large quantities of file modifications in user directories and under `C:\Windows\System32\`. In cloud environments, the occurrence of anomalous high-volume deletion events, such as the `DeleteDBCluster` and `DeleteGlobalCluster` events in AWS, or a high quantity of data deletion events, such as `DeleteBucket`, within a short period of time may indicate suspicious activity.

4.2.14.2. Data Encrypted for Impact (T1486)

Technique Information	
Technique ID	T1486
Technique Name	Data Encrypted for Impact
Technique Description	Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted. In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted (and often renamed and/or tagged with specific file markers). Adversaries may need to first employ other behaviors, such as [File and Directory Permissions Modification](T1222) or [System Shutdown/Reboot](T1529), in order to unlock and/or gain access to manipulate these files. In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR. To maximize impact on the target organization, malware designed for encrypting data may have worm-like features to propagate across a network by leveraging other attack techniques like [Valid Accounts](T1078), [OS Credential Dumping](T1003), and [SMB/Windows Admin Shares](T1021.002). Encryption malware may also leverage [Internal Defacement](T1491.001), such as changing victim wallpapers, or otherwise intimidate victims by sending ransom notes or other messages to connected printers (known as "print bombing"). In cloud environments, storage objects within compromised accounts may also be encrypted.

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%
File: File Modification	47.8%	34.23%
Cloud Storage: Cloud Storage Metadata	14.29%	9.52%
Process: Process Creation	39.25%	37.38%
Cloud Storage: Cloud Storage Modification	14.29%	14.29%

Technique Analysis	
Overall Score	**24.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**70/100**
Overall Log Source Coverage	**39.1%**
Overall Log Collection Coverage	**33.15%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Behavior Prevention on Endpoint	On Windows 10, enable cloud-delivered protection and Attack Surface Reduction (ASR) rules to block the execution of files that resemble ransomware.
Data Backup	Consider implementing IT disaster recovery plans that contain procedures for regularly taking and

	testing data backups that can be used to restore organizational data.
Implement Detection/Monitoring Capabilities	Use process monitoring to monitor the execution and command line parameters of binaries involved in data destruction activity, such as vssadmin, wbadmin, and bcdedit. Monitor for the creation of suspicious files as well as unusual file modification activity. In particular, look for large quantities of file modifications in user directories. In some cases, monitoring for unusual kernel driver installation activity can aid in detection. In cloud environments, monitor for events that indicate storage objects have been anomalously replaced by copies.

4.2.14.3. Service Stop (T1489)

Technique Information	
Technique ID	T1489
Technique Name	Service Stop
Technique Description	Adversaries may stop or disable services on a system to render those services unavailable to legitimate users. Stopping critical services or processes can inhibit or stop response to an incident or aid in the adversary's overall objectives to cause damage to the environment. Adversaries may accomplish this by disabling individual services of high importance to an organization, such as `MSExchangeIS`, which will make Exchange content inaccessible . In some cases, adversaries may stop or disable many or all services to render systems unusable. Services or processes may not allow for modification of their data stores while running. Adversaries may stop services or processes in order to conduct [Data Destruction](T1485) or [Data Encrypted for Impact](T1486) on the data stores of services like Exchange and SQL Server.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Process: OS API Execution	49.89%	48.17%
File: File Modification	47.8%	34.23%
Service: Service Metadata	0.23%	0.23%
Process: Process Termination	39.25%	37.38%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**29.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**71/100**
Overall Log Source Coverage	**38.34%**
Overall Log Collection Coverage	**35.62%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Network Segmentation	Operate intrusion detection, analysis, and response systems on a separate network from the production environment to lessen the chances that an adversary can see and interfere with critical response functions.
Restrict File and Directory Permissions	Ensure proper process and file permissions are in place to inhibit adversaries from disabling or interfering with critical services.
Restrict Registry Permissions	Ensure proper registry permissions are in place to inhibit adversaries from disabling or interfering with critical services.
User Account Management	Limit privileges of user accounts and groups so that only authorized administrators can interact with service changes and service configurations.
Implement Detection/Monitoring Capabilities	Monitor processes and command-line arguments to see if critical processes are terminated or stop running. Monitor for edits for modifications to services and

	startup programs that correspond to services of high importance. Look for changes to services that do not correlate with known software, patch cycles, etc. Windows service information is stored in the Registry at `HKLM\SYSTEM\CurrentControlSet\Services`. Systemd service unit files are stored within the /etc/systemd/system, /usr/lib/systemd/system/, and /home/.config/systemd/user/ directories, as well as associated symbolic links. Alterations to the service binary path or the service startup type changed to disabled may be suspicious. Remote access tools with built-in features may interact directly with the Windows API to perform these functions outside of typical system utilities. For example, `ChangeServiceConfigW` may be used by an adversary to prevent services from starting.

4.2.14.4. *Inhibit System Recovery (T1490)*

Technique Information	
Technique ID	T1490
Technique Name	Inhibit System Recovery
Technique Description	Adversaries may delete or remove built-in operating system data and turn off services designed to aid in the recovery of a corrupted system to prevent recovery. This may deny access to available backups and recovery options. Operating systems may contain features that can help fix corrupted systems, such as a backup catalog, volume shadow copies, and automatic repair features. Adversaries may disable or delete system recovery features to augment the effects of [Data Destruction](T1485) and [Data Encrypted for Impact](T1486). A number of native Windows utilities have been used by adversaries to disable or delete system recovery features: * \`vssadmin.exe\` can be used to delete all volume shadow copies on a system - \`vssadmin.exe delete shadows /all /quiet\` * [Windows Management Instrumentation](T1047) can be used to delete volume shadow copies - \`wmic shadowcopy delete\` * \`wbadmin.exe\` can be used to delete the Windows Backup Catalog - \`wbadmin.exe delete catalog -quiet\` * \`bcdedit.exe\` can be used to disable automatic Windows recovery features by modifying boot configuration data - \`bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no\`

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Deletion	72.99%	72.99%

Service: Service Metadata	0.23%	0.23%
Windows Registry: Windows Registry Key Modification	45.98%	45.98%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**36.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**64/100**
Overall Log Source Coverage	**40.88%**
Overall Log Collection Coverage	**40.51%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Data Backup	Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.
Operating System Configuration	Consider technical controls to prevent the disabling of services or deletion of files involved in system recovery.
Implement Detection/Monitoring Capabilities	Use process monitoring to monitor the execution and command line parameters of binaries involved in inhibiting system recovery, such as vssadmin, wbadmin, and bcdedit. The Windows event logs, ex. Event ID 524 indicating a system catalog was deleted, may contain entries associated with suspicious activity. Monitor the status of services involved in system recovery. Monitor the registry for changes associated with system recovery

	features (ex: the creation of `HKEY_CURRENT_USER\Software\Policies\Microsoft\PreviousVersions\DisableLocalPage`).

4.2.14.5. Defacement (T1491)

Technique Information	
Technique ID	T1491
Technique Name	Defacement
Technique Description	Adversaries may modify visual content available internally or externally to an enterprise network, thus affecting the integrity of the original content. Reasons for [Defacement](T1491) include delivering messaging, intimidation, or claiming (possibly false) credit for an intrusion. Disturbing or offensive images may be used as a part of [Defacement](T1491) in order to cause user discomfort, or to pressure compliance with accompanying messages.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
File: File Creation	72.99%	57.47%
Application Log: Application Log Content	66.67%	33.33%
File: File Modification	47.8%	34.23%

Technique Analysis	
Overall Score	**25.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**31/100**
Overall Log Source Coverage	**63.03%**

Overall Log Collection Coverage	**31.26%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Data Backup	Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.
Implement Detection/Monitoring Capabilities	Monitor internal and external websites for unplanned content changes. Monitor application logs for abnormal behavior that may indicate attempted or successful exploitation. Use deep packet inspection to look for artifacts of common exploit traffic, such as SQL injection. Web Application Firewalls may detect improper inputs attempting exploitation.

4.2.14.6. *Firmware Corruption (T1495)*

Technique Information	
Technique ID	T1495
Technique Name	Firmware Corruption
Technique Description	Adversaries may overwrite or corrupt the flash memory contents of system BIOS or other firmware in devices attached to a system in order to render them inoperable or unable to boot, thus denying the availability to use the devices and/or the system. Firmware is software that is loaded and executed from non-volatile memory on hardware devices in order to initialize and manage device functionality. These devices could include the motherboard, hard drive, or video cards. In general, adversaries may manipulate, overwrite, or corrupt firmware in order to deny the use of the system or devices. Depending on the device, this attack may also result in [Data Destruction](T1485).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Firmware: Firmware Modification	91.95%	91.95%

Technique Analysis	
Overall Score	**51.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**20/100**
Overall Log Source Coverage	**91.95%**
Overall Log Collection Coverage	**91.95%**
Detection Capability Present	**No**

Detection Sources	-

Mitigations	
Name	**Description**
Boot Integrity	Check the integrity of the existing BIOS and device firmware to determine if it is vulnerable to modification.
Privileged Account Management	Prevent adversary access to privileged accounts or access necessary to replace system firmware.
Update Software	Patch the BIOS and other firmware as necessary to prevent successful use of known vulnerabilities.
Implement Detection/Monitoring Capabilities	System firmware manipulation may be detected. Log attempts to read/write to BIOS and compare against known patching behavior.

4.2.14.7. Resource Hijacking (T1496)

Technique Information	
Technique ID	T1496
Technique Name	Resource Hijacking
Technique Description	Adversaries may leverage the resources of co-opted systems in order to solve resource intensive problems, which may impact system and/or hosted service availability. One common purpose for Resource Hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. Adversaries may consume enough system resources to negatively impact and/or cause affected machines to become unresponsive. Servers and cloud-based systems are common targets because of the high potential for available resources, but user endpoint systems may also be compromised and used for Resource Hijacking and cryptocurrency mining. Containerized environments may also be targeted due to the ease of deployment via exposed APIs and the potential for scaling mining activities by deploying or compromising multiple containers within an environment or cluster. Additionally, some cryptocurrency mining malware identify then kill off processes for competing malware to ensure it's not competing for resources. Adversaries may also use malware that leverages a system's network bandwidth as part of a botnet in order to facilitate [Network Denial of Service](T1498) campaigns and/or to seed malicious torrents.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
File: File Creation	72.99%	57.47%

Network Traffic: Network Traffic Flow	64.66%	0.0%
Sensor Health: Host Status	36.78%	31.06%
Network Traffic: Network Connection Creation	45.45%	45.45%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**24.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**50.85%**
Overall Log Collection Coverage	**36.22%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Consider monitoring process resource usage to determine anomalous activity associated with malicious hijacking of computer resources such as CPU, memory, and graphics processing resources. Monitor for suspicious use of network resources associated with cryptocurrency mining software. Monitor for common cryptomining software process names and files on local systems that may indicate compromise and resource usage.

4.2.14.8. *Network Denial of Service (T1498)*

Technique Information	
Technique ID	T1498
Technique Name	Network Denial of Service
Technique Description	Adversaries may perform Network Denial of Service (DoS) attacks to degrade or block the availability of targeted resources to users. Network DoS can be performed by exhausting the network bandwidth services rely on. Example resources include specific websites, email services, DNS, and web-based applications. Adversaries have been observed conducting network DoS attacks for political purposes and to support other malicious activities, including distraction, hacktivism, and extortion. A Network DoS will occur when the bandwidth capacity of the network connection to a system is exhausted due to the volume of malicious traffic directed at the resource or the network connections and network devices the resource relies on. For example, an adversary may send 10Gbps of traffic to a server that is hosted by a network with a 1Gbps connection to the internet. This traffic can be generated by a single system or multiple systems spread across the internet, which is commonly referred to as a distributed DoS (DDoS). To perform Network DoS attacks several aspects apply to multiple methods, including IP address spoofing, and botnets. Adversaries may use the original IP address of an attacking system, or spoof the source IP address to make the attack traffic more difficult to trace back to the attacking system or to enable reflection. This can increase the difficulty defenders have in defending against the attack by reducing or eliminating the effectiveness of filtering by the source address on network defense devices. For DoS attacks targeting the hosting system directly, see [Endpoint Denial of Service](T1499).

Related Data Source Components

Name	Log Source Coverage	Log Collection Coverage
Sensor Health: Host Status	36.78%	31.06%
Network Traffic: Network Traffic Flow	64.66%	0.0%

Technique Analysis	
Overall Score	**44.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**23/100**
Overall Log Source Coverage	**50.72%**
Overall Log Collection Coverage	**15.53%**
Detection Capability Present	**Yes**
Detection Sources	• FortiGate

Mitigations	
Name	**Description**
Filter Network Traffic	When flood volumes exceed the capacity of the network connection being targeted, it is typically necessary to intercept the incoming traffic upstream to filter out the attack traffic from the legitimate traffic. Such defenses can be provided by the hosting Internet Service Provider (ISP) or by a 3rd party such as a Content Delivery Network (CDN) or providers specializing in DoS mitigations.
Implement Detection/Monitoring Capabilities	Detection of Network DoS can sometimes be achieved before the traffic volume is sufficient to cause impact to the availability of the service, but such response time typically requires very aggressive monitoring and responsiveness or

	services provided by an upstream network service provider. Typical network throughput monitoring tools such as netflow, SNMP, and custom scripts can be used to detect sudden increases in network or service utilization. Real-time, automated, and qualitative study of the network traffic can identify a sudden surge in one type of protocol can be used to detect an Network DoS event as it starts. Often, the lead time may be small and the indicator of an event availability of the network or service drops. The analysis tools mentioned can then be used to determine the type of DoS causing the outage and help with remediation.

4.2.14.9. *Endpoint Denial of Service (T1499)*

Technique Information	
Technique ID	T1499
Technique Name	Endpoint Denial of Service
Technique Description	Adversaries may perform Endpoint Denial of Service (DoS) attacks to degrade or block the availability of services to users. Endpoint DoS can be performed by exhausting the system resources those services are hosted on or exploiting the system to cause a persistent crash condition. Example services include websites, email services, DNS, and web-based applications. Adversaries have been observed conducting DoS attacks for political purposes and to support other malicious activities, including distraction, hacktivism, and extortion. An Endpoint DoS denies the availability of a service without saturating the network used to provide access to the service. Adversaries can target various layers of the application stack that is hosted on the system used to provide the service. These layers include the Operating Systems (OS), server applications such as web servers, DNS servers, databases, and the (typically web-based) applications that sit on top of them. Attacking each layer requires different techniques that take advantage of bottlenecks that are unique to the respective components. A DoS attack may be generated by a single system or multiple systems spread across the internet, which is commonly referred to as a distributed DoS (DDoS). To perform DoS attacks against endpoint resources, several aspects apply to multiple methods, including IP address spoofing and botnets. Adversaries may use the original IP address of an attacking system, or spoof the source IP address to make the attack traffic more difficult to trace back to the attacking system or to enable reflection. This can increase the difficulty defenders have in defending against the attack by reducing or eliminating the effectiveness of filtering by the source address on network defense devices. Botnets are commonly used to conduct DDoS attacks against networks and services. Large botnets can generate a significant

	amount of traffic from systems spread across the global internet. Adversaries may have the resources to build out and control their own botnet infrastructure or may rent time on an existing botnet to conduct an attack. In some of the worst cases for DDoS, so many systems are used to generate requests that each one only needs to send out a small amount of traffic to produce enough volume to exhaust the target's resources. In such circumstances, distinguishing DDoS traffic from legitimate clients becomes exceedingly difficult. Botnets have been used in some of the most high-profile DDoS attacks, such as the 2012 series of incidents that targeted major US banks. In cases where traffic manipulation is used, there may be points in the global network (such as high traffic gateway routers) where packets can be altered and cause legitimate clients to execute code that directs network packets toward a target in high volume. This type of capability was previously used for the purposes of web censorship where client HTTP traffic was modified to include a reference to JavaScript that generated the DDoS code to overwhelm target web servers. For attacks attempting to saturate the providing network, see [Network Denial of Service](T1498).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Network Traffic: Network Traffic Content	64.66%	0.0%
Sensor Health: Host Status	36.78%	31.06%
Network Traffic: Network Traffic Flow	64.66%	0.0%
Application Log: Application Log Content	66.67%	33.33%

Technique Analysis	
Overall Score	**19.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**34/100**
Overall Log Source Coverage	**58.19%**
Overall Log Collection Coverage	**16.1%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Filter Network Traffic	Leverage services provided by Content Delivery Networks (CDN) or providers specializing in DoS mitigations to filter traffic upstream from services.
Implement Detection/Monitoring Capabilities	Detection of Endpoint DoS can sometimes be achieved before the effect is sufficient to cause significant impact to the availability of the service, but such response time typically requires very aggressive monitoring and responsiveness. Typical network throughput monitoring tools such as netflow, SNMP, and custom scripts can be used to detect sudden increases in circuit utilization. Real-time, automated, and qualitative study of the network traffic can identify a sudden surge in one type of protocol can be used to detect an attack as it starts. In addition to network level detections, endpoint logging and instrumentation can be useful for detection. Attacks targeting web applications may generate logs in the web server, application server, and/or database server that can be used to identify the type of attack, possibly before the impact is felt.

	Externally monitor the availability of services that may be targeted by an Endpoint DoS.

4.2.14.10. System Shutdown/Reboot (T1529)

Technique Information	
Technique ID	T1529
Technique Name	System Shutdown/Reboot
Technique Description	Adversaries may shutdown/reboot systems to interrupt access to, or aid in the destruction of, those systems. Operating systems may contain commands to initiate a shutdown/reboot of a machine or network device. In some cases, these commands may also be used to initiate a shutdown/reboot of a remote computer or network device. Shutting down or rebooting systems may disrupt access to computer resources for legitimate users. Adversaries may attempt to shutdown/reboot a system after impacting it in other ways, such as [Disk Structure Wipe](T1561.002) or [Inhibit System Recovery](T1490), to hasten the intended effects on system availability.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Command: Command Execution	45.98%	45.98%
Sensor Health: Host Status	36.78%	31.06%
Process: Process Creation	39.25%	37.38%

Technique Analysis	
Overall Score	**53.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**47/100**
Overall Log Source Coverage	**40.67%**

Overall Log Collection Coverage	**38.14%**
Detection Capability Present	**Yes**
Detection Sources	• BitDefender

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Use process monitoring to monitor the execution and command line parameters of binaries involved in shutting down or rebooting systems. Windows event logs may also designate activity associated with a shutdown/reboot, ex. Event ID 1074 and 6006. Unexpected or unauthorized commands from network cli on network devices may also be associated with shutdown/reboot, e.g. the `reload` command.

4.2.14.11. *Account Access Removal (T1531)*

Technique Information	
Technique ID	T1531
Technique Name	Account Access Removal
Technique Description	Adversaries may interrupt availability of system and network resources by inhibiting access to accounts utilized by legitimate users. Accounts may be deleted, locked, or manipulated (ex: changed credentials) to remove access to accounts. Adversaries may also subsequently log off and/or perform a [System Shutdown/Reboot](T1529) to set malicious changes into place. In Windows, [Net](S0039) utility, `Set-LocalUser` and `Set-ADAccountPassword` [PowerShell](T1059.001) cmdlets may be used by adversaries to modify user accounts. In Linux, the `passwd` utility may be used to change passwords. Accounts could also be disabled by Group Policy. Adversaries who use ransomware may first perform this and other Impact behaviors, such as [Data Destruction](T1485) and [Defacement](T1491), before completing the [Data Encrypted for Impact](T1486) objective.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Active Directory: Active Directory Object Modification	100.0%	100.0%
User Account: User Account Deletion	100.0%	98.28%
User Account: User Account Modification	66.1%	63.61%

Technique Analysis

Overall Score	**48.0%**
Status	**Needs imminent remediation**
Sector Specific Priority	**52/100**
Overall Log Source Coverage	**88.7%**
Overall Log Collection Coverage	**87.3%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Implement Detection/Monitoring Capabilities	Use process monitoring to monitor the execution and command line parameters of binaries involved in deleting accounts or changing passwords, such as use of [Net](S0039). Windows event logs may also designate activity associated with an adversary's attempt to remove access to an account: * Event ID 4723 - An attempt was made to change an account's password * Event ID 4724 - An attempt was made to reset an account's password * Event ID 4726 - A user account was deleted * Event ID 4740 - A user account was locked out Alerting on [Net](S0039) and these Event IDs may generate a high degree of false positives, so compare against baseline knowledge for how systems are typically used and correlate modification events with other indications of malicious activity where possible.

4.2.14.12. Disk Wipe (T1561)

Technique Information	
Technique ID	T1561
Technique Name	Disk Wipe
Technique Description	Adversaries may wipe or corrupt raw disk data on specific systems or in large numbers in a network to interrupt availability to system and network resources. With direct write access to a disk, adversaries may attempt to overwrite portions of disk data. Adversaries may opt to wipe arbitrary portions of disk data and/or wipe disk structures like the master boot record (MBR). A complete wipe of all disk sectors may be attempted. To maximize impact on the target organization in operations where network-wide availability interruption is the goal, malware used for wiping disks may have worm-like features to propagate across a network by leveraging additional techniques like [Valid Accounts](T1078), [OS Credential Dumping](T1003), and [SMB/Windows Admin Shares](T1021.002).

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
Driver: Driver Load	45.98%	45.98%
Command: Command Execution	45.98%	45.98%
Drive: Drive Access	45.45%	45.45%
Drive: Drive Modification	45.45%	45.45%
Process: Process Creation	39.25%	37.38%

Technique Analysis

Overall Score	**24.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**32/100**
Overall Log Source Coverage	**44.42%**
Overall Log Collection Coverage	**44.05%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Data Backup	Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.
Implement Detection/Monitoring Capabilities	Look for attempts to read/write to sensitive locations like the partition boot sector, master boot record, disk partition table, or BIOS parameter block/superblock. Monitor for direct access read/write attempts using the `\\\\.\\` notation. Monitor for unusual kernel driver installation activity.

4.2.14.13. *Data Manipulation (T1565)*

Technique Information	
Technique ID	T1565
Technique Name	Data Manipulation
Technique Description	Adversaries may insert, delete, or manipulate data in order to influence external outcomes or hide activity, thus threatening the integrity of the data. By manipulating data, adversaries may attempt to affect a business process, organizational understanding, or decision making. The type of modification and the impact it will have depends on the target application and process as well as the goals and objectives of the adversary. For complex systems, an adversary would likely need special expertise and possibly access to specialized software related to the system that would typically be gained through a prolonged information gathering campaign in order to have the desired impact.

Related Data Source Components		
Name	**Log Source Coverage**	**Log Collection Coverage**
File: File Creation	72.99%	57.47%
File: File Metadata	67.37%	55.82%
Process: OS API Execution	49.89%	48.17%
File: File Modification	47.8%	34.23%
Network Traffic: Network Traffic Flow	64.66%	0.0%
File: File Deletion	72.99%	72.99%
Network Traffic: Network Traffic Content	64.66%	0.0%

Technique Analysis	
Overall Score	**27.0%**
Status	**Needs immediate remediation**
Sector Specific Priority	**30/100**
Overall Log Source Coverage	**62.91%**
Overall Log Collection Coverage	**38.38%**
Detection Capability Present	**No**
Detection Sources	-

Mitigations	
Name	**Description**
Encrypt Sensitive Information	Consider encrypting important information to reduce an adversary\u2019s ability to perform tailored data modifications.
Network Segmentation	Identify critical business and system processes that may be targeted by adversaries and work to isolate and secure those systems against unauthorized access and tampering.
Remote Data Storage	Consider implementing IT disaster recovery plans that contain procedures for taking regular data backups that can be used to restore organizational data.
Restrict File and Directory Permissions	Ensure least privilege principles are applied to important information resources to reduce exposure to data manipulation risk.
Implement Detection/Monitoring Capabilities	Where applicable, inspect important file hashes, locations, and modifications for suspicious/unexpected values. With some critical processes involving transmission of data, manual or out-of-band integrity checking may be useful for identifying manipulated data.

4.3. Recommendations

Finally, tailored recommendations and future improvements will be listed here.