Remote Workforce Impact on Threat Defenses

Author: Sean Goodwin, SeanGoodwin@protonmail.ch
Advisor: Lenny Zeltser

Accepted: January 30, 2021

Abstract

As organizations embrace remote work, the defensive security posture needs to be re-examined to effectively address threats while facing new or different constraints and tools. This paper investigates the prevention and detection control effectiveness against the known adversary Tactics, Techniques, and Procedures (TTPs) documented within the MITRE ATT&CK ® taxonomy in a remote working (work from home, WFH) environment.

1. Introduction

According to a June 2020 survey by PWC, over 50% of executives expect to offer some sort of remote work options as a long-term effect of the remote-work surge due to the COVID-19 pandemic (PricewaterhouseCoopers, 2020). As organizations embrace remote work, the defensive security posture needs to be re-examined to effectively address threats while facing new or different constraints and tools. This is highlighted in the 2020 Verizon Data Breach Investigations Report as 45% of breaches were related to hacking, 22% included Social Engineering, and 17% included malware (Verizon, 2020, p. 7).

What impact does a Work from Home (WFH) posture have on an organization's ability to detect and respond to MITRE ATT&CK® techniques (MITRE ATT&CK®, 2020)? The MITRE ATT&CK® taxonomy has classified various attacker tactics and techniques into categories, as well as determining which techniques are used in an enterprise environment. These techniques can be executed in a test environment to emulate the actions of attackers as a means of evaluating an organization's prevention and detection capabilities.

According to MITRE, "MITRE ATT&CK® is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community" (MITRE ATT&CK®, 2020).

Threat emulation exercises allow organizations to execute test scenarios in a controlled manner to test the defensive controls against well-known attacker actions. For this paper, we are using Atomic Red Team to emulate attack actions as documented in the MITRE ATT&CK® taxonomy.

According to the developers, "Atomic Red Team is a library of simple tests that every security team can execute to test their defenses. Tests are focused, have few dependencies, and are defined in a structured format that can be used by automation frameworks" (Atomic Red Team, 2020). Using Atomic Red Team provides two benefits:

Tests are easily repeatable as there are scripts to be executed instead of relying on manual execution. Also, the Atomic Red Team repository has been forked as a means of archiving the tests to preserve the conditions of this testing environment. The archive can be found at https://github.com/0xSeanG/atomic-red-team.

Organizations may need to shift the focus of their defensive controls to adapt to the risks facing a remote workforce. Specifically, organizations need to ensure there are appropriate prevention and detection controls deployed to endpoints that may be working non-standard hours and from several locations. This paper seeks to emulate the attacker techniques typically seen in enterprise-level attacks but applying them against a WFH environment, which typically is a substantially less robust network than an enterprise location. While these attack techniques are classified as those facing the enterprise, PwC's US Remote Work Survey notes that 77% of executives have employees working remotely at least one day per week during the COVID-19 pandemic, placing home networks at risk of attack for a softer target (PricewaterhouseCoopers, 2020).

2. Description of the Testing Environment

There were two different target hosts used in testing these items. The first endpoint was configured to represent an organization with a more mature security program with a pre-existing remote workforce. This endpoint will be referred to as the "Mature Endpoint." This endpoint was configured with the following tools:

- Enterprise VPN
- Endpoint Detection and Response
- Privilege Management utility
- Low-Privileged user account

The second endpoint was configured to represent an immature security program, like those organizations who are not used to a remote workforce. This endpoint will be referred to as the "Basic Endpoint." This endpoint was configured with the built-in Windows Defender Antivirus installed and up-to-date . No additional third-party security tools were added to this endpoint. This endpoint is intended to represent an organization that enables remote work without the addition of their own security tools.

These two endpoints did have a few configurations in common, specifically that both machines were built using Windows 10 Enterprise with current patches installed (10.0.18363 Built 18363). Additionally, the end-user in both scenarios authenticated using password-only local authentication.

The two machines can be easily compared with this table:

Security Controls	Mature Endpoint	Basic Endpoint
User with "Local Admin" Rights	No	Yes
Endpoint Detection and Response (EDR)	Yes	No
Centralized Collection of Endpoint Logs	Yes	No
Enterprise Virtual Private Network	Yes	No

2.1 Assumptions

2.1.1 Local Administrator Rights

The Basic Endpoint was configured so that the end-user would be logged in with a user account that was assigned Local Administrator rights. The idea behind the Basic Endpoint was to emulate an organization that did not have a formal remote workforce process and rushed to get remote workers operational with minimal downtime. Configuring end-users to have Local Administrator rights allows for remote support without having direct connectivity to the endpoint, as the support technician can have the end user perform actions with elevated rights.

The Mature Endpoint was configured to restrict Local Administrator rights in such a way that the user account used for testing was a low privilege user. It is important to note that 26 of the test scenarios require local administrative rights to run successfully.

These scenarios do, however, present another detectable event that could be monitored for further investigation.

2.1.2 Application Whitelisting

The Mature Endpoint was configured to allow the use of PowerShell by a standard end user. Many organizations with a mature security environment limit the use of scripting languages for end-users, but for this research paper, we wanted to make the two test machines comparable in the test approach. If application whitelisting was used to restrict the use of PowerShell, Atomic Red Team would be blocked from running without being able to emulate any of the actual techniques.

2.1.3 Vendor Neutral

The Mature Endpoint was configured with enterprise-grade endpoint controls, but the specific tooling is intentionally excluded from this paper. The focus of the paper is the altered control environment and defensive processes facing a WFH environment and is not meant to be a test of specific tools' capabilities.

3. Attacker activity

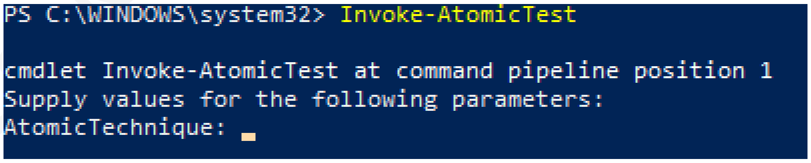
The MITRE ATT&CK® taxonomy includes twelve tactics comprising various techniques. These are the specific actions being taken by threat actors. Atomic Red Team was used to focus on two tactics, Privilege Escalation, and Discovery. These tactics were chosen based on the risks present with successful execution, as well as having the highest coverage from Atomic Red Team. Full details of the Atomic Red Team coverage of the MITRE ATT&CK® taxonomy can be found in Appendix B. The Privilege Escalation tactic has 10 of 12 (83%) techniques with at least one test automated through Atomic Red Team. The Discovery tactic- 20 of 24 (83%) techniques have at least some level of coverage.

3.1 Emulation Overview

Before reviewing the results of the testing, it would be useful to provide a high-level overview of how the tests were executed. A complete guide for configuration and execution exists within the Atomic Read Team documentation, which can be found at https://atomicredteam.io/.

3.1.1 Install Invoke-AtomicRedTeam

There is a second GitHub project, Invoke-AtomicRedTeam, that is used to make the execution of the tests within Atomic Red Team even easier. "Invoke-AtomicRedTeam is a PowerShell module to execute tests as defined in the atomics folder of Red Canary's Atomic Red Team project" ("invoke-atomicredteam," n.d.). This PowerShell module allows the user to supply the Technique ID in question (Figure-1), and the test file from Atomic Red Team is executed.



```
PS C:\WINDOWS\system32> Invoke-AtomicTest

cmdlet Invoke-AtomicTest at command pipeline position 1
Supply values for the following parameters:
AtomicTechnique: _
```

Figure 1 - Execution of the Invoke-AtomicTest PowerShell Module

Installation is as simple as one command (shown in Figure-3), which will download both the Invoke-AtomicTest module and the supporting Atomic Red Team folders:

```
IEX (IWR 'https://raw.githubusercontent.com/redcanaryco/invoke-
atomicredteam/master/install-atomicredteam.ps1' -UseBasicParsing);
Install-AtomicRedTeam -getAtomics
```

Figure 2 – Download and install the relevant files to install Atomic Red Team.

To import the Invoke-AtomicTest module and begin testing, run the command shown in Figure-3:

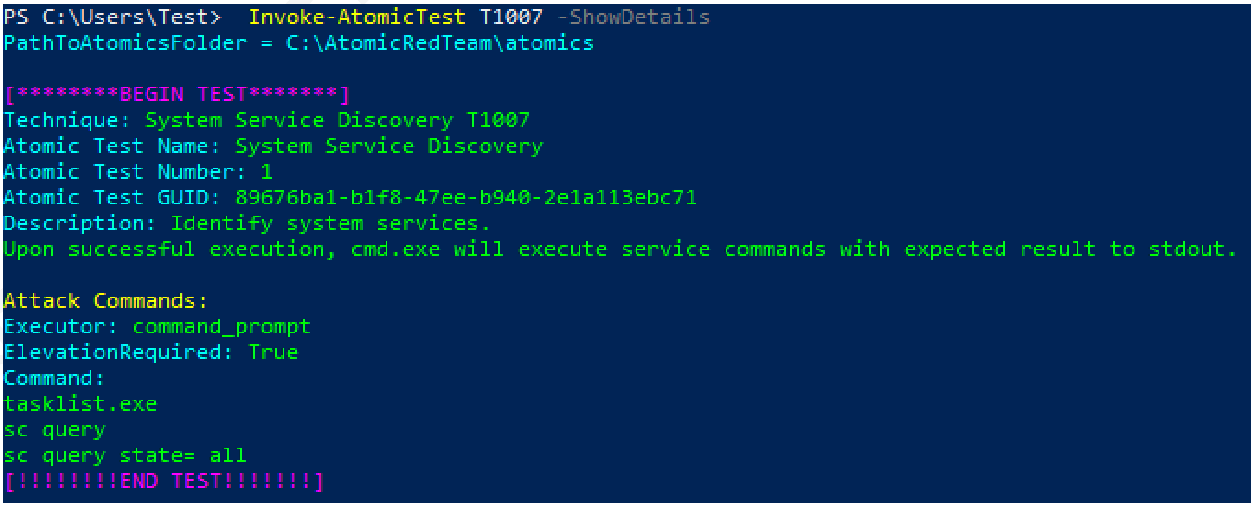
```
Import-Module "C:\AtomicRedTeam\invoke-atomicredteam\Invoke-AtomicRedTeam.psd1" -
Force
```

Figure 3 – Importing the Invoke-AtomicTest PowerShell module.

3.1.2 Running a Test

Once the Invoke-AtomicTests module is installed and imported, you can begin to run tests by supplying the Technique ID as the first parameter. There are several flags that will prove useful in performing your own testing, but at a minimum you should be aware of -ShowDetails and -CheckPrereqs.

The -ShowDetails flag will give you a detailed output of exactly what will be run if a Technique ID is submitted through Invoke-AtomicTest (Figure-4). Many techniques have multiple tests contained within a single technique. Identification of these tests, and selective execution can be useful for fine-tuning logging or alerting capabilities.



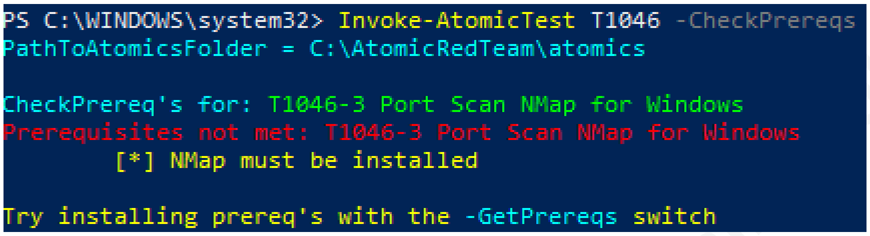
```
PS C:\Users\Test>  Invoke-AtomicTest T1007 -ShowDetails
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

[********BEGIN TEST*******]
Technique: System Service Discovery T1007
Atomic Test Name: System Service Discovery
Atomic Test Number: 1
Atomic Test GUID: 89676ba1-b1f8-47ee-b940-2e1a113ebc71
Description: Identify system services.
Upon successful execution, cmd.exe will execute service commands with expected result to stdout.

Attack Commands:
Executor: command_prompt
ElevationRequired: True
Command:
tasklist.exe
sc query
sc query state= all
[!!!!!!!!END TEST!!!!!!!]
```

Figure 4 - Example "-ShowDetails" output showing the relevant data points and attacker commands.

The -CheckPrereqs flag will validate that all prerequisite conditions have been met (Figure-5). A few common prerequisites to be aware of would be the requirement for the test to be executed with elevated rights and the need for third-party tools, such as Nmap.



```
PS C:\WINDOWS\system32> Invoke-AtomicTest T1046 -CheckPrereqs
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

CheckPrereq's for: T1046-3 Port Scan NMap for Windows
Prerequisites not met: T1046-3 Port Scan NMap for Windows
        [*] NMap must be installed

Try installing prereq's with the -GetPrereqs switch
```

Figure 5 - Example "-CheckPrereqs" output showing errors where prerequisites are not met.

4. Analysis

Our analysis focused on classifying each test under one of three results:

- Emulation was prevented from executing
- Emulation was allowed to execute, but was detected
- Emulation was allowed to execute and was not detected

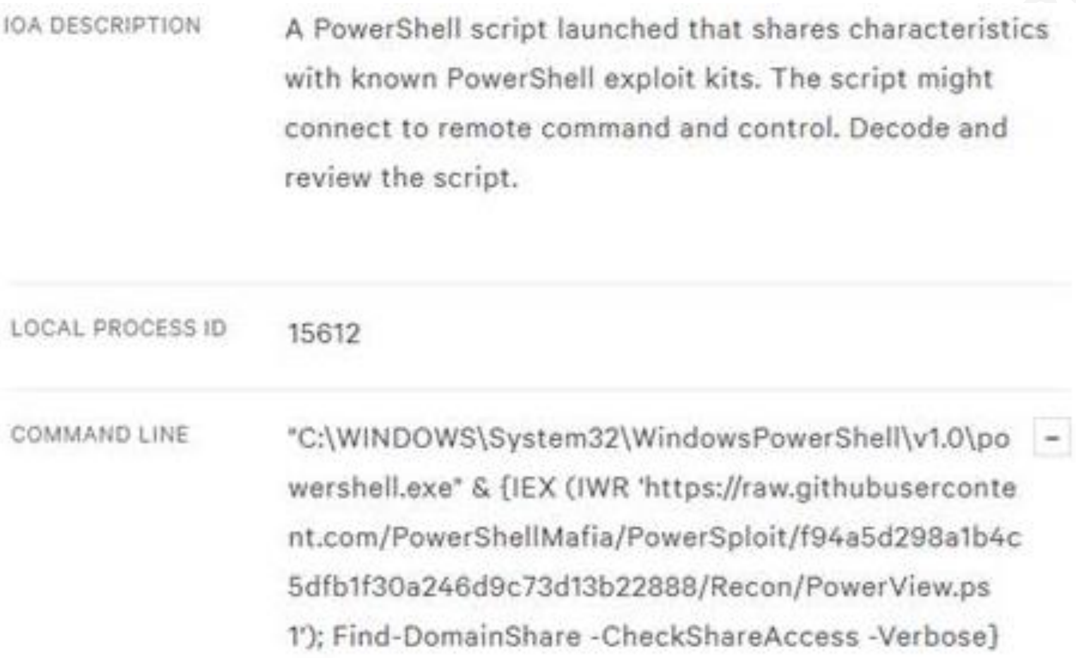
These results were captured in detailed results matrices, which can be found in Appendix C and Appendix D for the Privilege Escalation and Discovery tactics, respectively.

4.1 Common Results

Before diving into the test scenarios where the two environments performed differently, we will cover the areas where the outcomes in both environments were the same. As you will see, the largest commonality between the two environments was, unfortunately, a lack of detection or prevention for many of the Discovery techniques.

4.1.1 Common Detections

In the execution of the Discovery techniques, both endpoints successfully detected the use of PowerView, which is a component of PowerSploit, as part of T1135, T1202.002, and T1033 (shown in Figure-6 and Figure-7). PowerView, as a component of PowerSploit, is a well-known PowerShell script commonly used in the post-exploitation phases of an attack on a Windows environment ("PowerSploit," 2015).



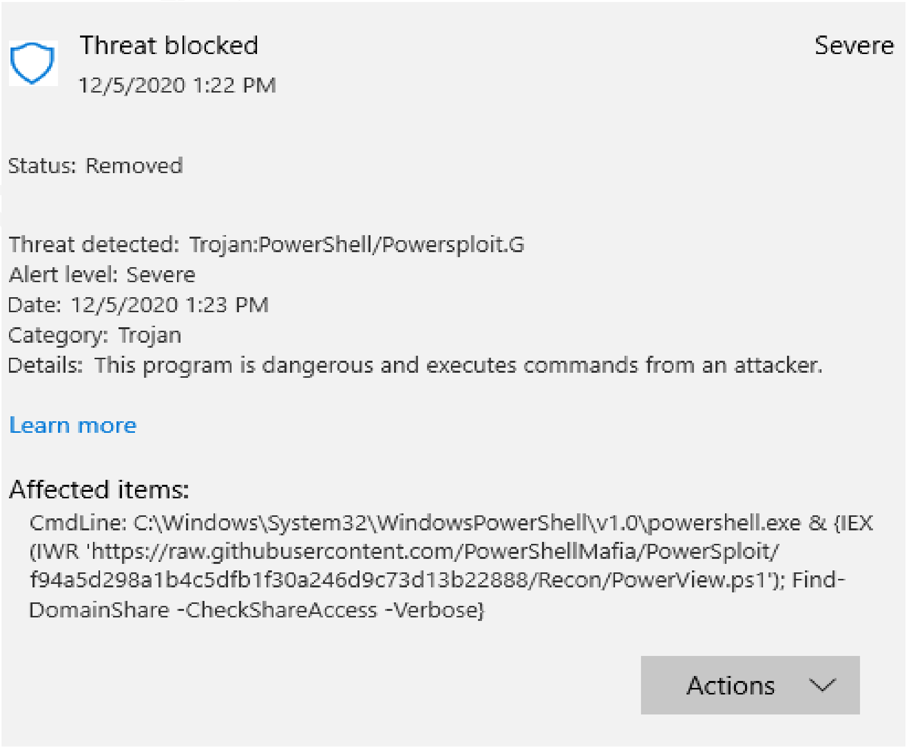
IOA DESCRIPTION: A PowerShell script launched that shares characteristics with known PowerShell exploit kits. The script might connect to remote command and control. Decode and review the script.

LOCAL PROCESS ID: 15612

COMMAND LINE:

```
"C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe" & {IEX (IWR 'https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/f94a5d298a1b4c5dfb1f30a246d9c73d13b22888/Recon/PowerView.ps1'); Find-DomainShare -CheckShareAccess -Verbose}
```

Figure 6 - Example detection notification from the Mature Endpoint



Threat blocked — Severe

12/5/2020 1:22 PM

Status: Removed

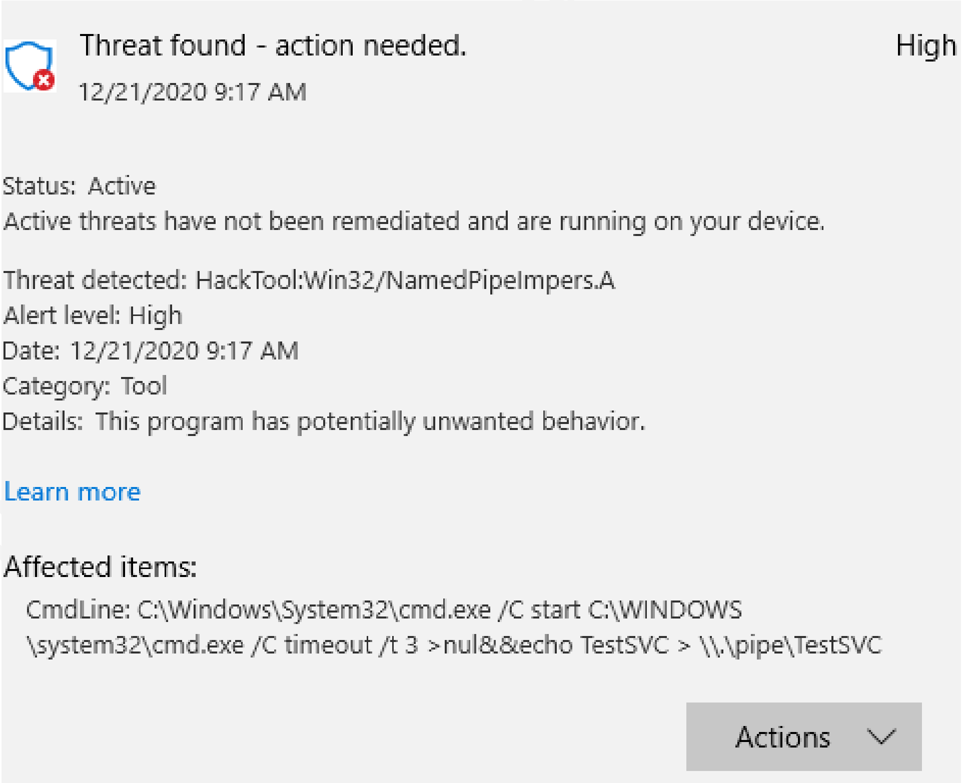
Threat detected: Trojan:PowerShell/Powersploit.G
Alert level: Severe
Date: 12/5/2020 1:23 PM
Category: Trojan
Details: This program is dangerous and executes commands from an attacker.

Affected items:

```
CmdLine: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe & {IEX (IWR 'https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/f94a5d298a1b4c5dfb1f30a246d9c73d13b22888/Recon/PowerView.ps1'); Find-DomainShare -CheckShareAccess -Verbose}
```

Figure 7 - Example detection and prevention notification from the Basic Endpoint.

In the execution of the Privilege Escalation techniques, both endpoints successfully detected the execution of T1548.002 and T1134.001 (Figure-8). These techniques rely on the use of very specific commands, which makes detection easier. It is also interesting to note that both techniques take advantage of using built-in processes that may be used for legitimate purposes, but due to the impact if abused, both environments defaulted to alerting on the activity. The use of legitimate tools and commands by attackers is often referred to as "living off the land." Relying on built-in tools, rather than bringing their own toolset, makes it more difficult to identify attacker activity among the "noise" generated by normal business activity.


Threat found - action needed. High
12/21/2020 9:17 AM

Status: Active
Active threats have not been remediated and are running on your device.

Threat detected: HackTool:Win32/NamedPipeImpers.A
Alert level: High
Date: 12/21/2020 9:17 AM
Category: Tool
Details: This program has potentially unwanted behavior.

Learn more

Affected items:

```
CmdLine: C:\Windows\System32\cmd.exe /C start C:\WINDOWS
\system32\cmd.exe /C timeout /t 3 >nul&&echo TestSVC > \\.\pipe\TestSVC
```

Figure 8 - Example of a detection notification without prevention actions taken on the Basic Endpoint.

4.1.2 Common Preventions

There were only two techniques in our testing that were prevented in both environments, T1548.002 in Privilege Escalation and T1033 in Discovery.

T1548.002 attempts to bypass User Account Control through several well-known methods. Both machines triggered the UAC confirmation prompt, which requires end-user confirmation to allow a process to elevate privileges. This was considered a prevention, as the end-user would need to take an action to allow the technique to continue execution.

T1033 was discussed in the previous section. This test was flagged for the use of PowerView. In addition to raising an alert, both endpoints prevented PowerSploit from running the "Invoke-UserHunter" module.

4.1.3 Common Failures

There were 21 tests in the Discovery tactic that went undetected in both environments. This speaks to the difficulty of detecting attackers who "live off the land" and take advantage of tools and commands native to the environment. T1087.001 - Local Account Discover is an example of a technique that, on paper, is performing an expected operation in a Windows environment. One of the tests within this technique simply executes "query user" to generate a list of users logged on to the endpoint. While this may be a valid command for an administrator to execute while troubleshooting an issue, this should stand out as suspicious when being executed by a standard end user.

The Privilege Escalation tactic had much different results, where there were only a few techniques that successfully executed and avoided triggering an alert on both hosts. These included:

- T1547.005 - Security Support Provider
- T1547.009 - Shortcut Modification
- T1037.001 - Logon Script
- T1546.013 - PowerShell Profile
- T1574.011 - Services Registry Permissions Weakness
- T1574.012 - COR_Profiler

While these tests were able to successfully execute the proof of concept attack, there may be a false sense of control failure at play. These attack emulations are designed to be a quick test to identify the attack path, without necessarily performing any malicious activity. To prove that the attack path was valid, many of these emulations run “calc.exe” to open the Windows calculator app, or “cmd.exe” to open a new command prompt window. Neither of these actions are malicious, and thus may be ignored by endpoint security tools. Those tools may alert on subsequent commands executed through command prompt, or on the execution of a different executable.

4.2 Unique Results

The largest difference between the two environments was performance in the Privilege Escalation tactic, with the Mature Endpoint fairing significantly better than the Basic Endpoint. The Mature Endpoint both prevented and generated alerts for 22 of the 30 techniques (73%), with an additional two techniques with partial prevention. The Basic Endpoint prevented one technique and alerted on a total of four techniques.

4.2.1 Basic Endpoint Unique Results

The performance of the Basic Endpoint against the Privilege Escalation techniques paints a clear picture of the risks organizations may face when allowing a remote workforce to operate with the bare minimum-security controls in place. In addition to the failures to prevent execution, an organization operating in the way is also reliant on the end-user to see the pop-up warning from Defender and reporting these to the security team. The Basis endpoint has two critical flaws, and both were clearly shown in this round of testing.

First, the end-user is provisioned with access rights that violate the idea of the principle of least privilege. While this would make it easier to provide remote technical support, this also allows a potential attacker more opportunity to execute commands and attempt to gain additional access.

The second critical flaw exposed here is the lack of centralized reporting from the endpoint security system. The centralized management for the tools tested in this research paper are only available through an E5 subscription, the highest tier, through Microsoft 365 ("Windows 10 Commercial Edition Comparison," 2020). In all other subscription levels, the tools would generate alerts, but would rely on the end-user reporting the potential issues. A second component of this shortcoming is the fail-open design of this decentralized control. Since there is no central management of the security policies and alerts, the endpoint is designed to alert on suspicious behavior, but only act against actions that are confirmed malicious. In the case of the Privilege Escalation tests, only one technique was blocked, whole three additional tests could run, but an alert was generated asking the end-user to review and decide if action should be taken.

4.2.2 Mature Endpoint Unique Results

Rather than focusing on the large number of techniques that were both blocked and generated alerts, we will discuss eight that either partially or completely evaded the defensive controls on the Mature Endpoint. The attack techniques that evaded both prevention and detection on the mature endpoint were:

- T1547.005 - Security Support Provider
- T1547.009 - Shortcut Modification
- T1037.001 - Logon Script (Windows)
- T1546.013 - PowerShell Profile
- T1574.011 - Services Registry Permissions Weakness
- T1574.012 - COR_PROFILER

The first two techniques to discuss fall under the T1547 - Boot or Logon Autostart Execution family. The first of these techniques was T1547.005 - Security Support Provider which seeks to modify the values under two registry keys. T1547.009 - Shortcut Modification attempts to modify shortcuts in the Startup folder to open both calc.exe and cmd.exe upon user login. This may be another case of the proof of concept being approved, where the value added “not-a-ssp” in place of a malicious dll being loaded, or the linking to calc.exe rather than a malicious executable is given as pass, which is discussed later in the paper.

T1037.001 - Logon Script (Windows) is similar to T1547.009 - Shortcut Modification, in the sense that the technique is seeking to have the end-user unknowingly initiate the attack action by simply logging in. In this case the logon script is modified to include a .bat script that creates a text file as a proof of concept. In the event of a real attack, this logon script would likely be used to execute additional commands under the elevated rights context.

T1546.013 - PowerShell Profile was the next technique that was executed without raising any alarms. This technique modified the user's PowerShell Profile to include an entry that opens calc.exe when PowerShell is launched.

T1574.011 - Services Registry Permissions Weakness attempts to identify registry keys with weak permissions that would allow the attacker to redirect a service to launch an executable under the attackers control, rather than the intended executable. This executable would be run under whatever context the hijacked service would run under.

T1574.012 - COR_PROFILER takes a similar approach to modifying registry key entries in order to initiate a downstream change in the executable being launched. This technique focuses on changing an environment variable to include launching a .dll put in place by the attacker.

The Mature Endpoint had two techniques that were partially blocked, T1547.001 - Registry Run Keys / Startup Folder and T1053.005 - Scheduled Task. T1547.001 - Registry Run Keys / Startup Folder seeks to add several scripts to be launched as part of the Startup process in both .vbs and .jse. The Mature Endpoint did prevent the modification of the RunOnce Key but allowed the placement of the .vbs and .jse scripts in the Startup folder. T1053.005 - Scheduled Task was partially blocked when attempting to initiate a remote scheduled task against "localhost" due to a credential failure. A failed logon attempt for the user "Domain\User" also provided an easily identifiable indicator of suspicious activity.

4.3 Key Takeaways

Both endpoints had successes as well as failures. It should come as no surprise that the Mature Endpoint fared much better, though the Basic Endpoint did generate pop-up alerts for the end user to report suspicious activity, but failed to centralize the event logs to a place where security personnel could detect and respond. The Mature Endpoint took this a step further by collecting log centrally, which would allow a security team to detect the activity independent of the end user reporting a pop-up alert but fell short in stopping techniques that abuse legitimate administrative tools.

The key takeaways from this research are summarized in the table below.

Endpoint	Notable Success	Notable Failures
Basic	● Windows Defender generated pop-up alerts where actions were suspicious.	● Lack of centralized log collection creates a single point of failure for detection.
Mature	● Centralized log collection allowed for detection where prevention was not successful.	● "Living off the Land" attacks leveraging legitimate administrative tools.

5. Implications for Future Research

5.1 Further Testing of Possible False Negatives

There were some techniques that were "allowed" to run or did not trigger alerts that did not meet expectations, especially when considering where other techniques of less-invasive means were blocked and detected. Our hypothesis is that the proof of concept commands were benign enough to be ignored by the security tools. These actions included things like launching calc.exe or appending harmless text to a registry key value. Further testing would allow for manual attempts to replace the proof of concept code with something more impactful.

5.2 Additional Tactics

This paper focused on two out of 14 tactics documented within MITRE ATT&CK ®. Organizations that are concerned about the full lifecycle of an attack against their remote workforce should consider evaluating their preventative and detective controls against additional tactics and techniques. Appendix A shows the full coverage of MITRE ATT&CK ® by the Atomic Red Team tool, which includes automated tests for 12 of the 14 tactics.

5.3 Manual Testing

Given more time, it would be interesting to complete manual tests for the remaining techniques that could not be automated through Atomic Red Team. Each of these technique entries in MITRE ATT&CK ® There were two techniques under the Privilege Escalation tactic, and four techniques under the Discovery tactic that were excluded from the scope of this research paper.

- Privilege Escalation
    - T1068 - Exploitation for Privilege Escalation
    - T1484 - Group Policy Modification
- Discovery
    - T1580 - Cloud Infrastructure Discovery
    - T1538 - Cloud Service Dashboard
    - T1526 - Cloud Service Discovery
    - T1120 - Peripheral Device Discovery

5.4 Advanced Toolsets

This testing focused on comparing endpoints at opposite ends of the spectrum of maturity in a control environment. This was done intentionally, as many organizations were forced into deploying remote working environments without planning or testing due to the COVID-19 pandemic. There are many different points along that spectrum where an organization can find themselves, and an organization will likely see themselves moving over time. As noted in the PWC survey, 55% or organizations plan to continue offering some form of remote work after the pandemic is over (PricewaterhouseCoopers, 2020).

6. Conclusion

The Basic endpoint fared considerably worse than the Mature endpoint in this testing. The Basic endpoint's largest failure was the lack of centralized log collection. This creates a single point of failure insofar as the end user is responsible for noticing the endpoint security alerts and reporting the alerts to the appropriate security personnel. A sound security program is built upon the combination of people, processes, and technology. Any organization that is stuck deploying machines like the Basic endpoint tested here must seek ways to bolster the people and process inputs to the equation.

Both endpoints struggled in preventing the malicious use of legitimate administrative tools. Organizations that have a mature control environment may benefit from implementing controls such as privileged account management to provide an additional layer of prevention controls, specifically those that rely on abusing legitimate administrative tools.

Remote work is not going away any time soon. Organizations need to prepare to address the threats facing employees and their endpoints as they leave the walled structure of a classic corporate network. Reviewing the documented attack techniques as present in the MITRE ATT&CK ® taxonomy provides organizations with tangible results with which they can compare various proposed security controls. When it comes to Privilege Escalation and Discovery, controls such as centralized alerting and restriction of Local Administrator rights stand out as two of the most critical controls to implement.

References

(2020). Atomic Red Team. Retrieved December 12, 2020, from https://atomicredteam.io/

(2020). MITRE ATT&CK®. Retrieved December 12, 2020, from https://attack.mitre.org/

invoke-atomicredteam. (n.d.). GitHub/Redcanaryco. Retrieved December 20, 2020, from https://github.com/redcanaryco/invoke-atomicredteam/

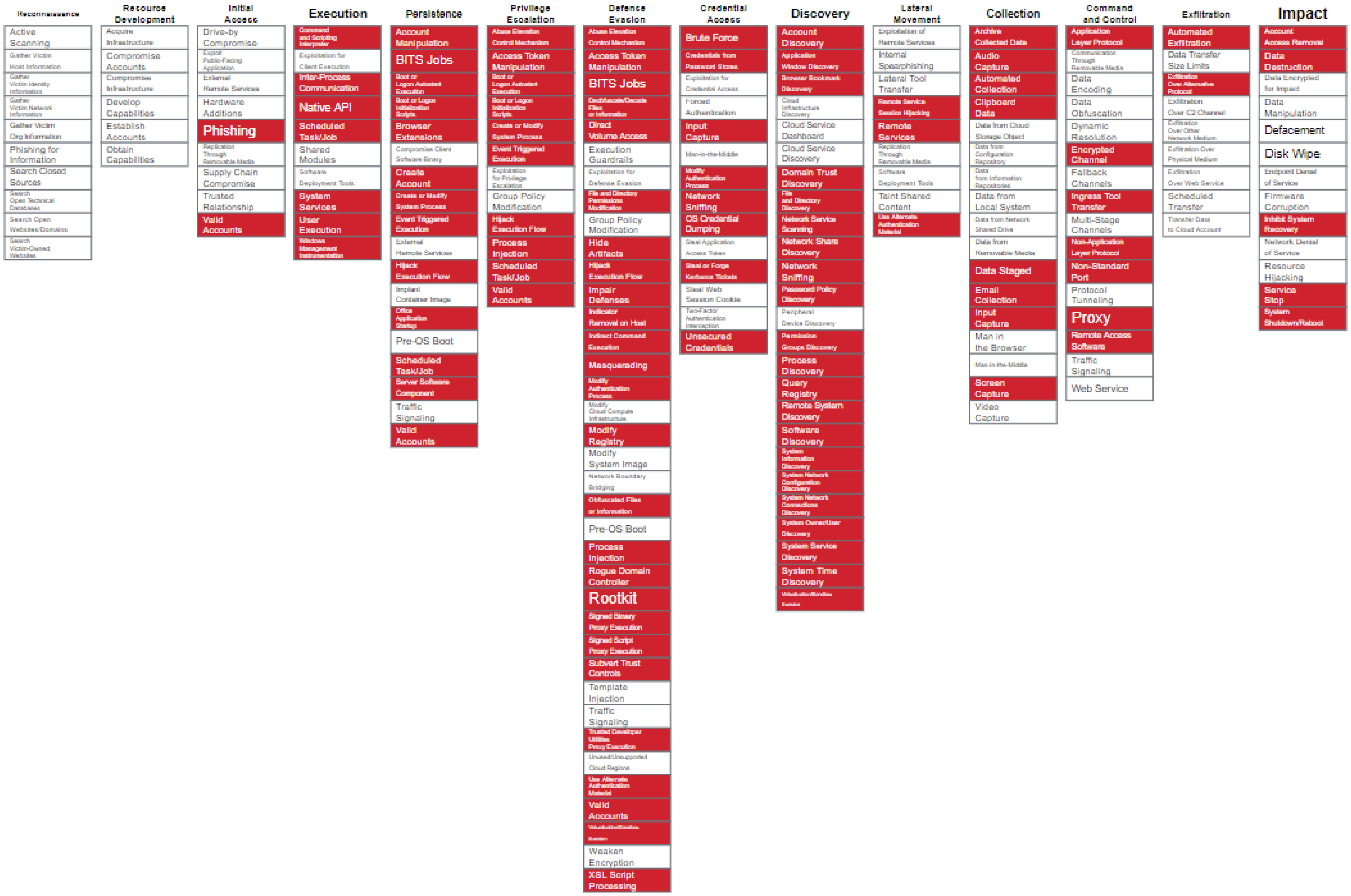
PowerSploit. (2015, December 18). GitHub/PowerShellMafia. Retrieved December 20, 2020, from https://github.com/PowerShellMafia/PowerSploit

PricewaterhouseCoopers. (2020, June 25). US Remote Work Survey: PwC. PwC. https://www.pwc.com/us/en/library/covid-19/us-remote-work-survey.html

Verizon. (2020). 2020 Data Breach Investigations Report. https://enterprise.verizon.com/resources/reports/dbir/

Windows 10 Commercial Edition Comparison. (2020, April). https://docs.microsoft.com/. https://wfbdevicemanagementprod.blob.core.windows.net/windowsforbusiness/Windows10_CommercialEdition_Comparison.pdf

Appendix - A: Atomic Red Team MITRE ATT&CK Coverage



Reconnaissance
- Active Scanning
- Gather Victim Host Information
- Gather Victim Identity Information
- Gather Victim Network Information
- Gather Victim Org Information
- Phishing for Information
- Search Closed Sources
- Search Open Technical Databases
- Search Open Websites/Domains
- Search Victim-Owned Websites

Resource Development
- Acquire Infrastructure
- Compromise Accounts
- Compromise Infrastructure
- Develop Capabilities
- Establish Accounts
- Obtain Capabilities

Initial Access
- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- 🟥 Phishing
- Replication Through Removable Media
- Supply Chain Compromise
- Trusted Relationship
- 🟥 Valid Accounts

Execution
- 🟥 Command and Scripting Interpreter
- Exploitation for Client Execution
- 🟥 Inter-Process Communication
- 🟥 Native API
- 🟥 Scheduled Task/Job
- Shared Modules
- Software Deployment Tools
- 🟥 System Services
- 🟥 User Execution
- 🟥 Windows Management Instrumentation

Persistence
- 🟥 Account Manipulation
- 🟥 BITS Jobs
- 🟥 Boot or Logon Autostart Execution
- 🟥 Boot or Logon Initialization Scripts
- 🟥 Browser Extensions
- Compromise Client Software Binary
- 🟥 Create Account
- 🟥 Create or Modify System Process
- 🟥 Event Triggered Execution
- External Remote Services
- 🟥 Hijack Execution Flow
- Implant Container Image
- 🟥 Office Application Startup
- Pre-OS Boot
- 🟥 Scheduled Task/Job
- 🟥 Server Software Component
- Traffic Signaling
- 🟥 Valid Accounts

Privilege Escalation
- 🟥 Abuse Elevation Control Mechanism
- 🟥 Access Token Manipulation
- 🟥 Boot or Logon Autostart Execution
- 🟥 Boot or Logon Initialization Scripts
- 🟥 Create or Modify System Process
- 🟥 Event Triggered Execution
- Exploitation for Privilege Escalation
- Group Policy Modification
- 🟥 Hijack Execution Flow
- 🟥 Process Injection
- 🟥 Scheduled Task/Job
- 🟥 Valid Accounts

Defense Evasion
- 🟥 Abuse Elevation Control Mechanism
- 🟥 Access Token Manipulation
- 🟥 BITS Jobs
- 🟥 Deobfuscate/Decode Files or Information
- 🟥 Direct Volume Access
- Execution Guardrails
- Exploitation for Defense Evasion
- 🟥 File and Directory Permissions Modification
- Group Policy Modification
- 🟥 Hide Artifacts
- 🟥 Hijack Execution Flow
- 🟥 Impair Defenses
- 🟥 Indicator Removal on Host
- 🟥 Indirect Command Execution
- 🟥 Masquerading
- 🟥 Modify Authentication Process
- Modify Cloud Compute Infrastructure
- 🟥 Modify Registry
- Modify System Image
- Network Boundary Bridging
- 🟥 Obfuscated Files or Information
- Pre-OS Boot
- 🟥 Process Injection
- 🟥 Rogue Domain Controller
- 🟥 Rootkit
- 🟥 Signed Binary Proxy Execution
- 🟥 Signed Script Proxy Execution
- 🟥 Subvert Trust Controls
- Template Injection
- Traffic Signaling
- 🟥 Trusted Developer Utilities Proxy Execution
- Unused/Unsupported Cloud Regions
- 🟥 Use Alternate Authentication Material
- 🟥 Valid Accounts
- 🟥 Virtualization/Sandbox Evasion
- Weaken Encryption
- 🟥 XSL Script Processing

Credential Access
- 🟥 Brute Force
- 🟥 Credentials from Password Stores
- Exploitation for Credential Access
- Forced Authentication
- 🟥 Input Capture
- Man-in-the-Middle
- 🟥 Modify Authentication Process
- 🟥 Network Sniffing
- 🟥 OS Credential Dumping
- Steal Application Access Token
- 🟥 Steal or Forge Kerberos Tickets
- Steal Web Session Cookie
- Two-Factor Authentication Interception
- 🟥 Unsecured Credentials

Discovery
- 🟥 Account Discovery
- 🟥 Application Window Discovery
- 🟥 Browser Bookmark Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- 🟥 Domain Trust Discovery
- 🟥 File and Directory Discovery
- 🟥 Network Service Scanning
- 🟥 Network Share Discovery
- 🟥 Network Sniffing
- 🟥 Password Policy Discovery
- Peripheral Device Discovery
- 🟥 Permission Groups Discovery
- 🟥 Process Discovery
- 🟥 Query Registry
- 🟥 Remote System Discovery
- 🟥 Software Discovery
- 🟥 System Information Discovery
- 🟥 System Network Configuration Discovery
- 🟥 System Network Connections Discovery
- 🟥 System Owner/User Discovery
- 🟥 System Service Discovery
- 🟥 System Time Discovery
- 🟥 Virtualization/Sandbox Evasion

Lateral Movement
- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- 🟥 Remote Service Session Hijacking
- 🟥 Remote Services
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- 🟥 Use Alternate Authentication Material

Collection
- 🟥 Archive Collected Data
- 🟥 Audio Capture
- 🟥 Automated Collection
- 🟥 Clipboard Data
- Data from Cloud Storage Object
- Data from Configuration Repository
- Data from Information Repositories
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- 🟥 Data Staged
- 🟥 Email Collection
- 🟥 Input Capture
- Man in the Browser
- Man-in-the-Middle
- 🟥 Screen Capture
- Video Capture

Command and Control
- 🟥 Application Layer Protocol
- Communication Through Removable Media
- Data Encoding
- Data Obfuscation
- Dynamic Resolution
- 🟥 Encrypted Channel
- Fallback Channels
- 🟥 Ingress Tool Transfer
- Multi-Stage Channels
- 🟥 Non-Application Layer Protocol
- 🟥 Non-Standard Port
- Protocol Tunneling
- 🟥 Proxy
- 🟥 Remote Access Software
- Traffic Signaling
- Web Service

Exfiltration
- 🟥 Automated Exfiltration
- Data Transfer Size Limits
- 🟥 Exfiltration Over Alternative Protocol
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium
- Exfiltration Over Physical Medium
- Exfiltration Over Web Service
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
- 🟥 Account Access Removal
- 🟥 Data Destruction
- Data Encrypted for Impact
- Data Manipulation
- Defacement
- Disk Wipe
- Endpoint Denial of Service
- Firmware Corruption
- 🟥 Inhibit System Recovery
- Network Denial of Service
- Resource Hijacking
- 🟥 Service Stop
- 🟥 System Shutdown/Reboot

The list below includes the steps to recreate the image shown above:

1. Browse to the MITRE ATT&CK® Navigator page: https://mitre-attack.github.io/attack-navigator/
2. Select “Open Existing Layer”
3. Insert the following URL in the box labeled “Load from URL”: https://raw.githubusercontent.com/0xSeanG/atomic-red-team/master/atomics/Indexes/Attack-Navigator-Layers/art-navigator-layer-windows.json
4. Click the arrow (“>”) to submit and load the color-coded layer.

Note: This pulls data from an archived version of the GitHub repository to match the configuration used in this research paper. To see the latest version, please visit the master Atomic Red Team repository at https://github.com/redcanaryco/atomic-red-team.

Appendix B – Scope of Automation

This table summarizes the number of techniques included in each tactic within MITRE ATT&CK® along with an indication of how many techniques can be automated via Atomic Red Team. This is discussed in detail in Section 3. Attacker activity (p. 5).

Tactic	Total Number of Techniques	Number available for automation
Initial Access	9	2
Execution	10	7
Persistence	18	12
Privilege Escalation	**12**	**10**
Defense Evasion	34	23
Credential Access	14	8
Discovery	**24**	**19**
Lateral Movement	9	3
Collection	16	8
Command and Control	16	8
Exfiltration	9	3
Impact	13	6

Appendix C - Privilege Escalation Detailed Results

Technique Number	Technique Name	Mature Prevent?	Mature Detect?	Basic Prevent?	Basic Detect?
T1548	Abuse Elevation Control Mechanism	N/A	N/A	N/A	N/A
0.002	Bypass User Account Control	Yes	Yes	Yes	Yes
T1134	Access Token Manipulation	N/A	N/A	N/A	N/A
0.001	Token Impersonation/Theft	Yes	Yes	Yes	Yes
0.004	Parent PID Spoofing	Yes	Yes	No	No
T1547	Boot or Logon Autostart Execution	N/A	N/A	N/A	N/A
0.001	Registry Run Keys / Startup Folder	Partial	Yes	No	No
0.004	Winlogon Helper DLL	Yes	Yes	No	No
0.005	Security Support Provider	No	No	No	No
0.009	Shortcut Modification	No	No	No	No
T1037	Boot or Logon Initialization Scripts	N/A	N/A	N/A	N/A
0.001	Logon Script (Windows)	No	No	No	No
T1543	Create or Modify System Process	N/A	N/A	N/A	N/A
0.003	Windows Service	Yes	Yes	No	No
T1546	Event Triggered Execution	N/A	N/A	N/A	N/A
0.001	Change Default File Association	Yes	Yes	No	No
0.002	Screensaver	Yes	Yes	No	No

Technique Number	Technique Name	Mature Prevent?	Mature Detect?	Basic Prevent?	Basic Detect?
0.003	Windows Management Instrumentation Event Subscription	Yes	Yes	No	No
0.007	Netsh Helper DLL	Yes	Yes	No	No
0.008	Accessibility Features	Yes	Yes	No	Yes
0.010	AppInit DLLs	Yes	Yes	No	No
0.011	Application Shimming	Yes	Yes	No	No
0.012	Image File Execution Options Injection	Yes	Yes	No	No
0.013	PowerShell Profile	No	No	No	No
T1574	Hijack Execution Flow	N/A	N/A	N/A	N/A
0.001	DLL Search Order Hijacking	Yes	Yes	No	No
0.002	DLL Side-Loading	Yes	Yes	No	No
0.009	Path Interception by Unquoted Path	Yes	Yes	No	No
0.011	Services Registry Permissions Weakness	No	No	No	No
0.012	COR_PROFILER	No	No	No	No
T1055	Process Injection	N/A	N/A	N/A	N/A
0.004	Asynchronous Procedure Call	Yes	Yes	No	Yes
0.012	Process Hollowing	Yes	Yes	No	No
T1053	Scheduled Task/Job	N/A	N/A	N/A	N/A
0.002	At (Windows)	Yes	Yes	No	No
0.005	Scheduled Task	Partial	Yes	No	No

Technique Number	**Technique Name**	**Mature Prevent?**	**Mature Detect?**	**Basic Prevent?**	**Basic Detect?**
T1078	Valid Accounts	N/A	N/A	N/A	N/A
0.001	Default Accounts	Yes	Yes	No	No

Appendix D - Discovery Detailed Results

Technique Number	Technique Name	Mature Prevent?	Mature Detect?	Basic Prevent?	Basic Detect?
T1087	Account Discovery	N/A	N/A	N/A	N/A
0.001	Local Account	No	No	No	No
0.002	Domain Account	No	No	No	No
T1010	Application Window Discovery	No	No	No	No
T1217	Browser Bookmark Discovery	No	No	No	No
T1482	Domain Trust Discovery	No	No	No	No
T1083	File and Directory Discovery	No	No	No	No
T1046	Network Service Scanning	Yes	No	No	No
T1135	Network Share Discovery	Partial	Yes	No	Yes
T1040	Network Sniffing	Yes	No	No	No
T1201	Password Policy Discovery	No	No	No	No
0.001	Local Groups	No	No	No	No
0.002	Domain Groups	Partial	Yes	No	Yes
T1057	Process Discovery	No	No	No	No
T1012	Query Registry	No	No	No	No
T1018	Remote System Discovery	No	No	No	No
T1518	Software Discovery	No	No	No	No

Technique Number	Technique Name	Mature Prevent?	Mature Detect?	Basic Prevent?	Basic Detect?
0.001	Security Software Discovery	No	No	No	No
T1082	System Information Discovery	No	No	No	No
T1016	System Network Configuration Discovery	No	No	No	No
T1049	System Network Connections Discovery	No	No	No	No
T1033	System Owner/User Discovery	Yes	No	Partial	Partial
T1007	System Service Discovery	No	No	No	No
T1124	System Time Discovery	No	No	No	No
0.001	System Checks	No	No	No	No