Attacking a Domain with Speed: PowerShell Remoting Versus GPO

Author: Justin Weis, jusweis@gmail.com
Advisor: *L. Crognale*

Accepted: *30 December, 2022*

Abstract

An organization sends a mass notification, "We are under attack. Disconnect all devices from the network." By removing any device before the malicious payload executes, the attacker fails to maximize their attack. In a Windows environment, multiple ways exist to execute commands remotely throughout the domain. This paper reviews various attacker techniques and identifies which technique achieves the attacker's objective the fastest. It also demonstrates how defenders can continue the defense once a domain controller is compromised.

1. Introduction

On 27 April 2022, Austin Peay State University tweeted, “Shut down all computers now” after experiencing a ransomware attack. For network defenders, the defense mechanism is a common approach to isolate an attack and prevent the further spread of malware. However, attackers missed an opportunity to infect as many systems as possible.

Figure 1. Announcement to Disconnect

When robust cyber reports are written (Biasini, 2022) (The Health and Safety Executive, 2021), there is considerable discussion on initial access, the elevation of privilege, and domain compromise. Missing from these reports is how the attacker spreads ransomware. The following research examines native Microsoft tools PowerShell and Group Policy Object (GPO) and the speed of each technique.

PowerShell remoting is a method for one computer to run code on remote machines (Stewart 2010). Attackers may exploit weaknesses and gain an administrative account with Domain Administer (DA) privileges. With a DA account, if PowerShell Remoting (PS-Remoting) configurations are enabled, the attacker can pass a list of computers to a PowerShell command and run commands on remote machines.

Justin Weis, jusweis@gmail.com

Another method employed by attackers is leveraging GPOs Logon Logoff scripts to instruct remote machines to run commands. An attacker must gain access to a Domain Controller (DC) and add the GPO script. Windows workstation queries the DC every 90 minutes (Mitwirkenden 2021). Once the machine logs off, after receiving the group policy, the remote host will run commands.

Security researchers have documented adversaries using both PowerShell and GPOs to deliver ransomware (Abrams 2021) (Goliath 2016). Heimdel Security attributed Conti ransomware using both GPO and PowerShell remoting (Tudor, 2022). Both methods have their strengths and weaknesses. The research will examine if one is more effective than the other.

2. Research Method

With 25 hosts joined to one domain with one domain controller, each host will run a command from PowerShell and GPO Logoff. First, PowerShell remoting will instruct each host to start a timer and generate a log when the command completes. Then GPO will send the same command and generate a new log.

2.1. Lab Environment

The environment for testing consists of three Intel NUC servers, each running VMware ESXI 7.0.3. Two of the NUC servers have 10 Windows 10 Enterprise workstations “https://info.microsoft.com/ww-landing-windows-10-enterprise.html” and the third NUC has one Windows 2019 Server and five Windows 10 workstations.

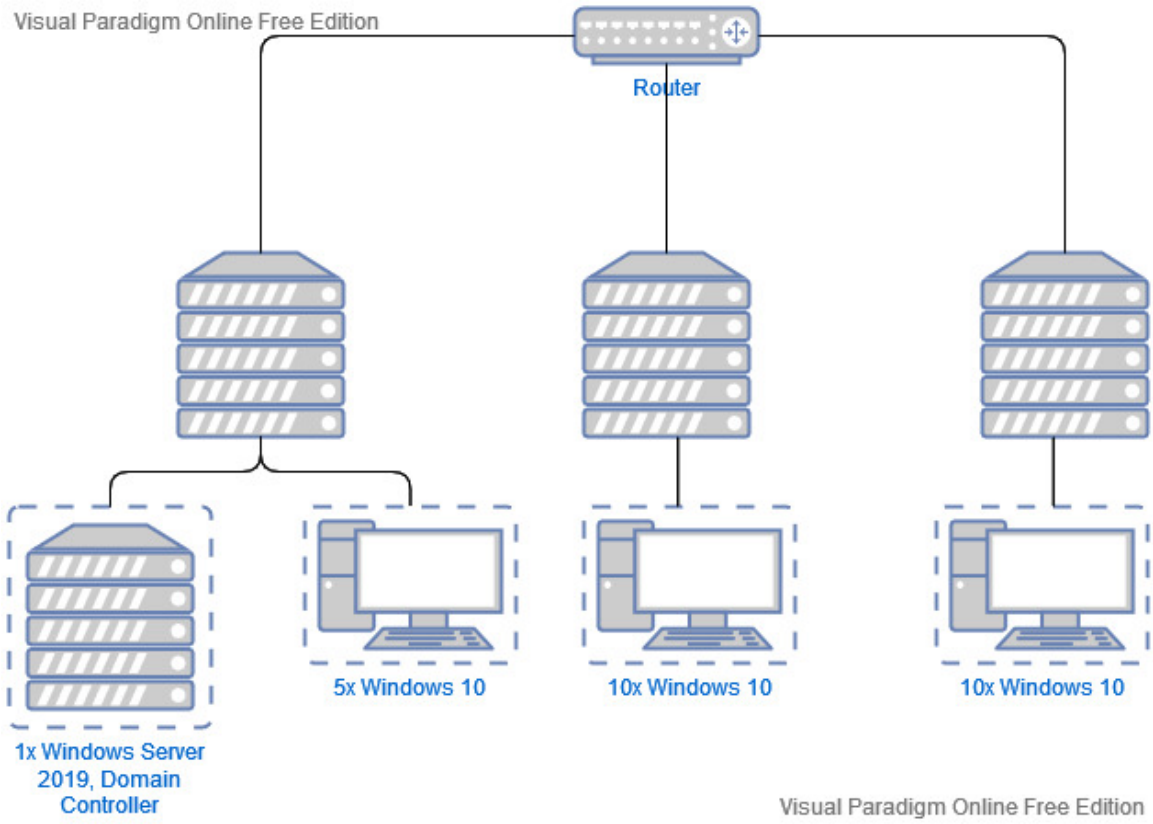


Figure 2. Lab Diagram

One router connects all of the virtual infrastructures. Enterprise environments have considerably more infrastructure hosting their workstations. However, in measuring the time taken to run remote commands, packet loss and network transport time is in seconds and would not create a significant difference. The 25 workstations and the measure of time taken to run remote commands is a sufficient representation of results within 10 seconds of precision for larger environments.

2.2. Script

Group Policy Logoff Script and PowerShell remoting both utilize PowerShell. Each of the 25 workstations run the following command.

```
$method="power"
mkdir C:\temp
$WriteVar = "C:\temp\$method$env:COMPUTERNAME$env:USERNAME.txt"
$ntptime=w32tm.exe /stripchart /computer:time-a-g.nist.gov /samples:2 /rdtsc
$ntptimeout = Out-String -InputObject $ntptime
hostname | Out-File -filepath $WriteVar
$ntptimeout | Out-File -filepath $WriteVar -Append
$startime=Get-Date -Format "HH:mm:ss:ffff"
if ((Start-Sleep -Seconds 217) -and ($stoptime=Get-Date -Format "HH:mm:ss:ffff")) {}else {$stoptime=Get-Date -Format "HH:mm:ss:ffff"}
"Started at $startime`nStopped at $stoptime`nUser ran is $env:USERNAME" | Out-File -FilePath $WriteVar -Append
```

Figure 3. PowerShell Script

To measure time across multiple systems, w32tm.exe is used to query a local Network Transport Protocol (NTP) Server. The function, Get-Date, queries the local workstation time. Windows workstations synchronize local time with the domain controller; however, NTP is a more accurate measure of time. To ensure precise time measurements, local workstation time and NTP differences are normalized.

Windows hosts query the domain controller every 90 minutes for updates to Microsoft Group Policy Objects. The PowerShell script adds a delay of 217 seconds to simulate if each workstation ran the command iteratively. Twenty-five hosts x 217 seconds is 90.41 minutes. Microsoft states the Start-Sleep timer will delay functions (Start-Sleep (Microsoft.PowerShell.Utility) - PowerShell, n.d.-b), however, while testing the start-sleep function did not wait before executing additional commands. To force the function "start-sleep," a conditional statement is utilized; first, the evaluation parameter completes ensuring "start-sleep" ends before the following command.

2.3. PowerShell Remoting

PowerShell remoting is unavailable when installing a new system from the Optical disc image (ISO) file format. In order to utilize PowerShell remoting, the workstations first must install the WinRM service and permit the Windows Firewall. In the test environment, each host enabled PS-Remoting through the function Enable-PSRemoting (Enable-PSRemoting (Microsoft.PowerShell.Core) - PowerShell, n.d.-b). When Enable-PSRemoting is performed the host listens on ports 5985 and 5986. In the test environment, the windows firewall is disabled instead of creating new firewall rules.

The following script is run from the Domain Controller to run a remote command on all endpoints.

```
(Get-ADComputer -LDAPFilter "(name=w*)").Name | sort-object > endpoints.txt
|
$parameters = @{
Comp= (Get-Content C:\Users\Priv\Documents\Test1PSremoting\endpoints.txt)
FilePath = 'C:\Users\Priv\Documents\Test1PSremoting\pscmdtest1.ps1'
}
Invoke-Command @parameters
```

Figure 4. PowerShell Remoting

The first line creates a file listing the workstations to run the remote commands. The parameters variable is a splat that stores variables to pass to a PowerShell function. Then Invoke-Command utilizes the splat parameters. Utilizing a splat parameter argument is the most efficient method as opposed to using Invoke-Command –Computer <file list>.

2.4. Group Policy Object

In research conducted by Tudor (2022), the malicious threat actor utilized Group Policy Logoff Scripts to deliver malware. Windows Domain Controllers provide the option to set LogOff Scripts for scripts that are run when the user logs off a workstation. Logoff is different than timeout with screen locking. In order to apply the Ground Policy to all workstations, in the Default Domain Policy a script is added to Configuration>Policies>Windows Settings>Scripts>Logoff. In order for workstations to run the script, the location of the script must be readable. Utilizing the domain controller's sysvol network file share, the script is placed in \\<domain.local>\SYSVOL\<domain.local>\\Scripts\ (Logon/Logoff). Note the script can be placed in any location that is readable by the endpoint. In order for the Logoff Script to run the user must be forced to logoff. This introduced a challenge that will be discussed in the findings. The logoff scripts failed to run as the user could not be forced to logoff.

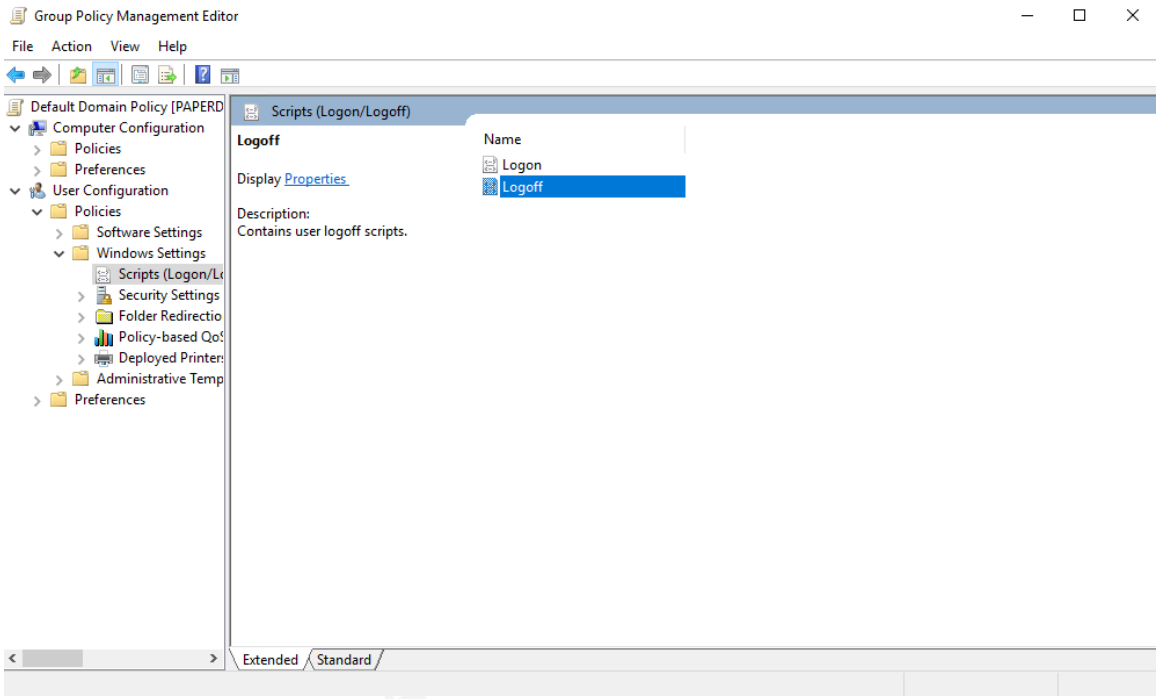


Figure 5. Logoff Script

3. Findings and Discussion (Exposition of the Data)

This section compares PowerShell Remoting and Group Policy Objects to run a PowerShell script across all workstations in a domain.

3.1. PowerShell Remoting

PowerShell remoting is the fastest method to deploy a script across all workstations. Within 10 seconds after initiating the command, all 25 workstations started the PowerShell script. Using the splatting method, the domain controller immediately sent the instructions to all workstations. When first evaluating, passing a list directly to the parameter "Computer Name," such as Invoke-Command – ComputerName <file list>, resulted in the command running iteratively instead of simultaneously. The method used with PowerShell remoting Invoke-Command determines speed.

Started 19:16:57				
19:17:01	19:17:01	19:17:01	19:17:05	19:17:06
19:17:01	19:17:01	19:17:02	19:17:05	19:17:06
19:17:01	19:17:01	19:17:03	19:17:05	19:17:06
19:17:01	19:17:01	19:17:03	19:17:05	19:17:07

Justin Weis, jusweis@gmail.com

19:17:01	19:17:01	19:17:03	19:17:06	19:17:07

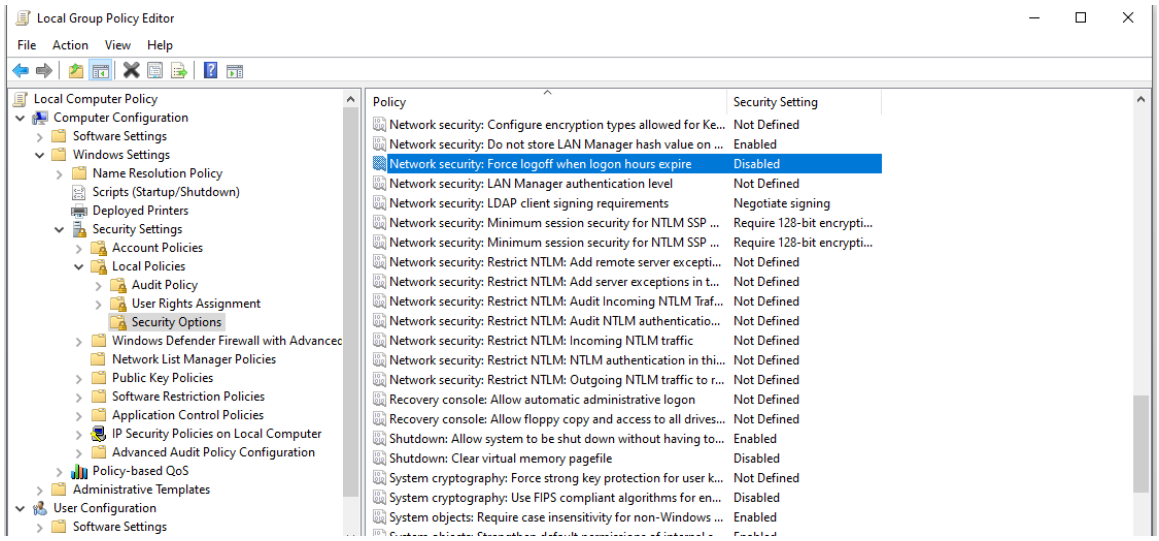
Table 1. PowerShell Remoting Completion Time

While PowerShell remoting is the fastest to run remote commands across every host, the method relied on all endpoints running the service and network and host rules permitting the traffic. Group policy modifications can modify host settings when settings are not enabled. However, modifying the Group takes 90 minutes to apply to all hosts before the settings enable PS-Remoting.

3.2. Group Policy Log-Off Scripts

Log-off scripts run when a user logs off; forcing the end user to log off proved challenging. While malicious actors use the technique, it is a poor solution. Log-off scripts, combined with an immediate task or separate Windows Management Instrument (WMI) command, are necessary to force a workstation to log off.

At first, the Group Policy Object "Network security: Force log-off when logon hours expire" appeared as a solution in conjunction with a log-off script to force a user to log-off a machine. However, the policy only applies to servers and does not force end users to log off on workstations and trigger the log-off script.



Justin Weis, jusweis@gmail.com

Figure 6. Network Security: Force Logoff

After attempting the Group Policy Object "Network security: Force log-off when logon hours expire," the next attempt utilized a scheduled task under Group Policy Management Editor. Adding a policy under User Configurations>Preferences>Control Panel Settings>Templates>Control Panel Settings>Scheduled Tasks adds a scheduled task to the Group Policy. The command shutdown –L is given on a schedule for when idle, with a setting set to start when the computer is idle. When first applied, the condition field in the Scheduled Task Group Policy Object for idle is not issued to the endpoint. Figure 7 displays the Scheduled Task on a remote workstation with the conditions field blank. After rebooting the machine, the conditional object appears, and the scheduled task works as expected. As the Scheduled Task Group Policy required a reboot to implement, the method is inadequate to force a log-off for a user to trigger the log-off script.

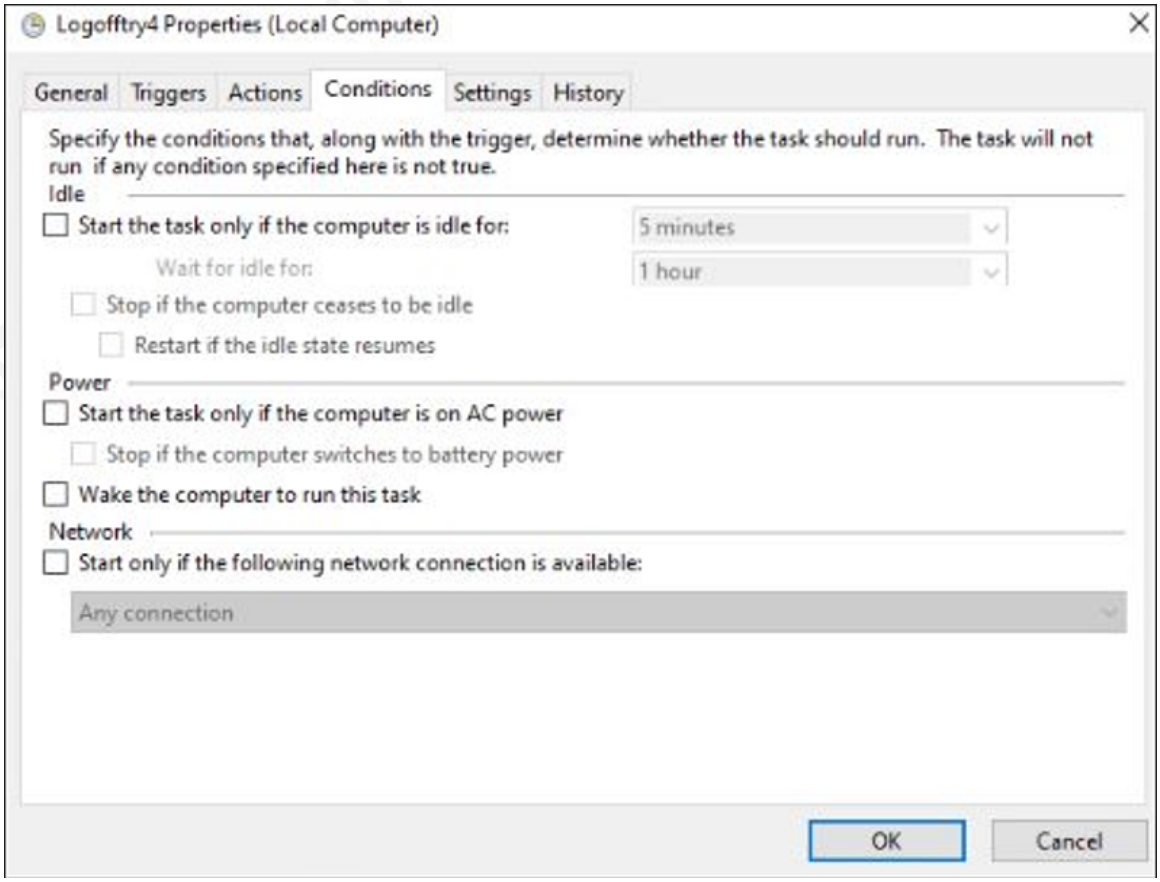


Figure 7. Scheduled Task Conditions

While the Network Security Group Policy Object and the scheduled task could not force a log-off, remote commands from a DC using Windows Management Instrument (WMI) can still force a user to log off. Several remote WMI command methods include PowerSploit's Invoke-WmiCommand, Impacket's wmiexec.py, and WmiExec.ps1. Any of those tools allows sending a remote command to trigger the log-off command. The value of WMI is that the port used is 135, and windows hosts have that port open by default.

3.3. Immediate Task

Using an Immediate Task through Group Policy can execute the script on remote machines. In research conducted by Richark (2022), he describes implementing an immediate task. First, add the PowerShell script to a readable location such as \\<domain.local>\SYSVOL\<domain.local>\\Scripts\. Next, configure the Immediate task in Group Policy Computer Configuration > Control Panel Settings > Scheduled Task > Immediate Task (for Windows7). Configure the action to “create” and Security options to run as SYSTEM. In the Action Tab, set the action to “Start a program” and specify the PowerShell command to run, pointing to the script that is readable by all workstations. Setting –ExecutionPolicy Bypass to the PowerShell script is necessary. The default setting of the Setting and Common tabs are sufficient. Once configured, the script will run on all workstations.

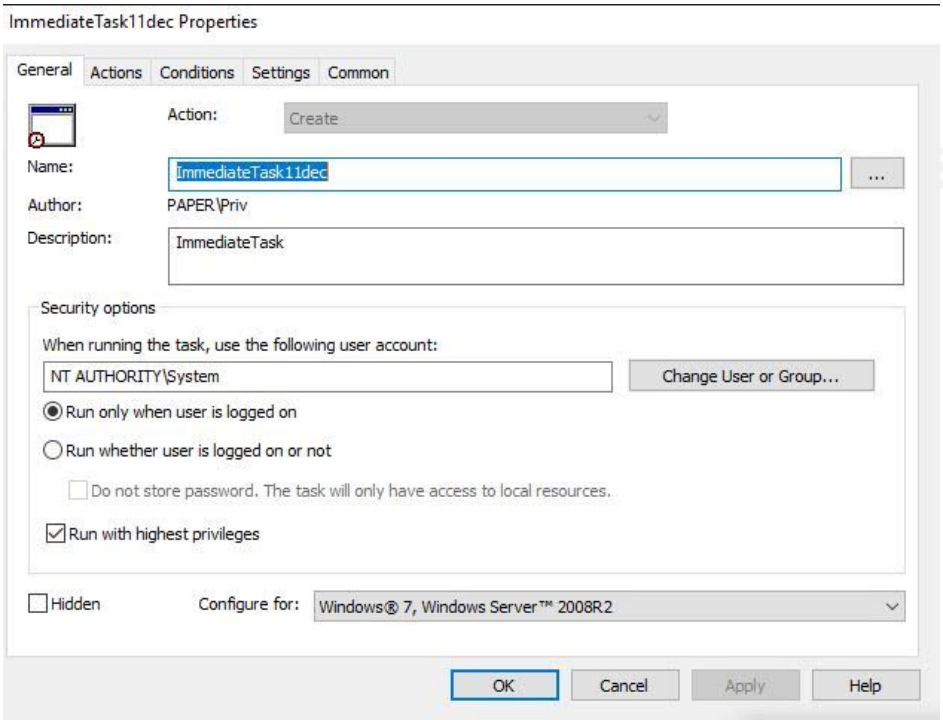


Figure 8. Immediate Task

In Figure 8, the built-in account System is used to run the Immediate Task. Utilizing the System command, the Immediate Task will run regardless if a user is logged on or off of the workstation.

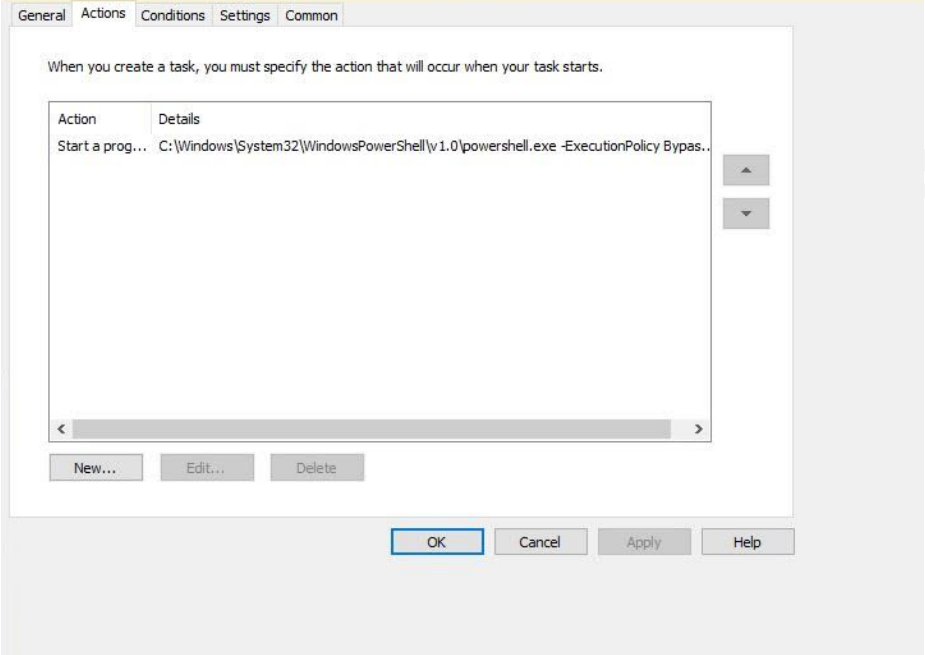


Figure 9. Immediate Task Action

In the results, the first remote machine ran the script 22 seconds after setting the GPO. Half of all workstations ran the script after one hour, and the last workstation ran the script 94 minutes after creating the Immediate Task.

Started 18:33:06				
18:33:28	19:05:36	19:21:33	19:40:55	19:56:39
18:36:18	19:08:50	19:24:20	19:44:25	19:59:22
18:41:20	19:10:13	19:27:00	19:47:33	20:03:38
18:45:09	19:11:33	19:37:57	19:50:00	20:05:21
18:54:11	19:15:13	19:39:05	19:54:35	20:08:34

Table 2. Immediate Task Completion Time

4. Recommendations and Implication

Knowing the artifacts and evidence of PowerShell remoting and GPO Immediate Tasks enables defenders to craft defenses to block, blunt, and deter attacker behavior.

Justin Weis, jusweis@gmail.com

4.1. Recommendations for Defenses for PowerShell

PowerShell remoting is the fastest method to execute commands. Some organizations may utilize the technology for remote system administration. If organizations do not use PowerShell remoting, the most straightforward defense mitigation is to uninstall the service. By default, windows servers have Windows Remote Management enabled.

By default, every domain member can log on to workstation machines remotely. To restrict permissions modify the Group Policy Object Security Settings > Local Policy > User Assignment Rights > Allow this computer from the network. Some devices, such as a file server, may require all authenticated users' remote access. Workstations, however, most likely only need some administrative accounts to have remote access. By removing the default entries and adding only necessary accounts, malicious actors will have limited accounts available to run malicious code remotely.

In addition or as an alternate, another User Assignment Right object can also restrict access, "Deny access to this computer from the network." The Group Policy restriction includes other services and protocols besides PowerShell/WinRM but also includes Windows Management Interface Console. In this lab, Windows Firewall is disabled for ease of use and simplicity. Unfortunately, some system administrators may take the same action or have a full permit rule for windows firewall for simplicity. Reviewing the endpoint windows firewall and restricting unneeded IP addresses access to the WinRM service limits the availability.

A malicious actor can bypass the two defensive measures by reconfiguring Group Policy. However, workstations must update with the Domain Controller before those changes are applied, thereby increasing the time to detect a malicious actor before they can successfully run remote PowerShell. Enterprise organizations may have multiple Organization Units (OU), and the most specific OU GPO takes precedence (Allen 2022). The malicious actor would have to change each OU or delete every OU GPO and place a domain-wide group policy. As defenders, this is an opportunity to detect the activity.

4.1.1. PowerShell Remoting Network Defenses

Below, in Figure 10, is a Wireshark display of PowerShell Remoting. The address “192.168.4.7” is a domain controller, and “192.168.3.11” is a host. In documentation

from Microsoft Corporation (2021), the PowerShell Remoting Protocol is encapsulated in Web Services Management Protocol Extensions and SOAP protocols. The payload transmitted across the wire is encrypted. Therefore an intrusion prevention system is not a solution to detect traffic as the traffic is encrypted.

However, for organizations that do not utilize PowerShell, blocking port 5985 between hosts at the network level will prevent PS-Remoting. For organizations that utilize PS-Remoting, a restrictive firewall that limits traffic between machines that require PS-Remoting will prevent attackers from using the technique on non-approved devices. Monitoring blocked internal traffic for port 5985 allows defenders to detect malicious activity.

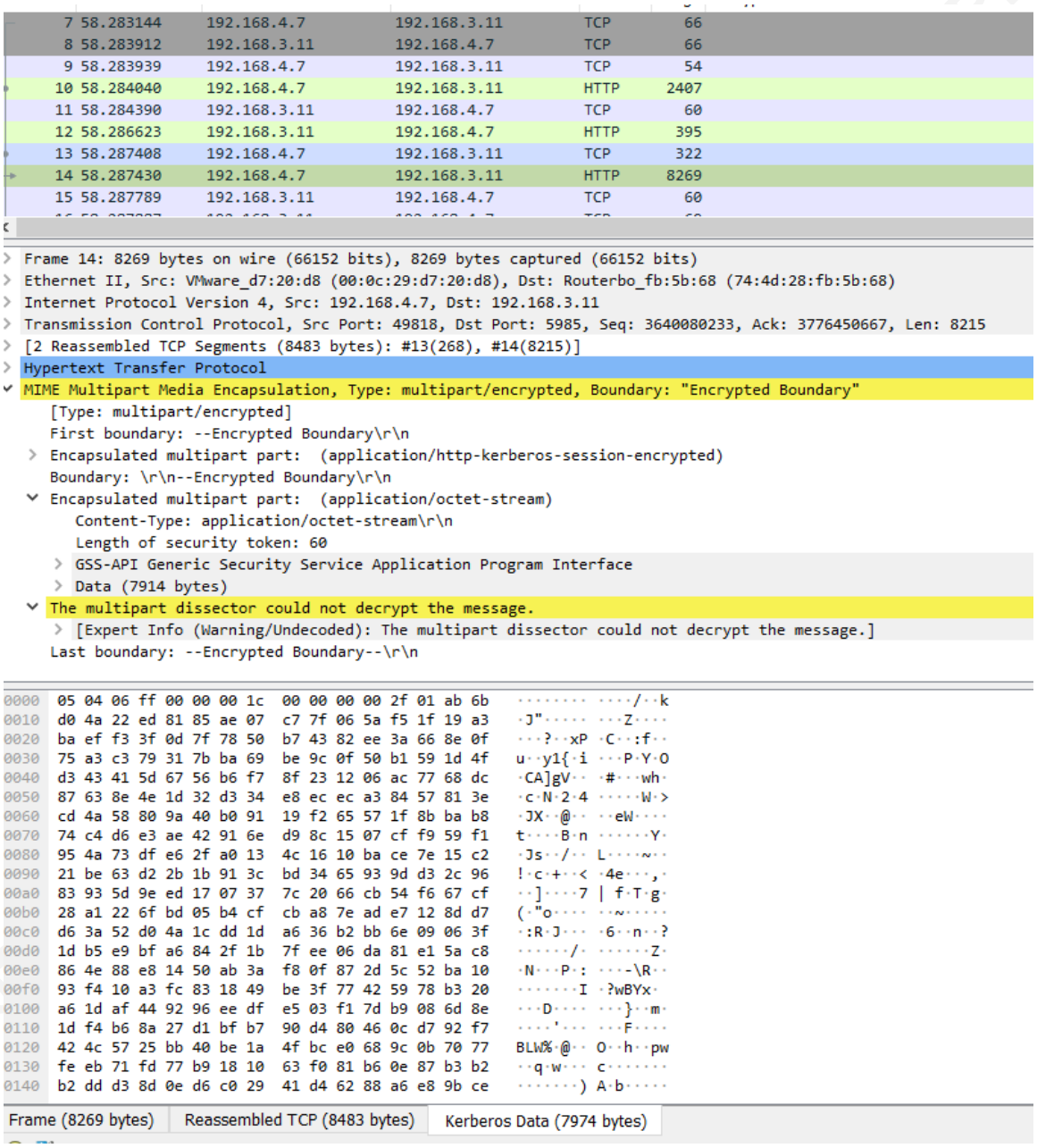


Figure 10. PowerShell Remoting Packet

4.2. Recommendations for Defenses for GPO

After performing the tests, we can concluded that group policy is a less efficient solution to PowerShell remoting. However, adversaries may utilize group policy when PowerShell is unavailable. There are few defenses for defenders to defend against malicious GPOs.

Justin Weis, jusweis@gmail.com

Detecting when Group Policy changes will ensure defenders are aware of potentially malicious activity. Enabling auditing is required to detect changes to Group Policy. To enable auditing, Domain Controllers require the following GPO setting Computer Configuration > Policies > Windows Settings > Security Settings > Advanced Audit Policy Configuration > Audit Policies/DS Access → Click "Audit Directory Service Changes"→ Click "Define" → Choose "Success." Once enabled, the server that changes Group Policy will generate Event ID 5136 (How to Audit Group Policy Changes Using the Security Event Log, n.d.).

To deploy a GPO all endpoints must read a remote file located in the domain on any system. Enabling File Integrity Monitoring on world-readable directors will empower defenders to detect unwanted activity. While some areas may be noisy, utilizing a Security Event and Incident Manager to detect script files (ps1, bat, wsf, hta) for files added to directories can notify defenders of potentially malicious activity.

4.2.1. Group Policy Network Defenses

Unlike PowerShell Remoting, Group Policy Objects are not encrypted. The unencrypted traffic allows defenders to stop or detect the activity.

Below in Figure 11 is a Wireshark display of a packet from the host "192.168.3.11" requesting to read the Immediate Task Group Policy Object from the domain controller "192.168.3.1". The host requests to read the GPO Globally Unique Identifier from the settings defined in the GPO under "ScheduledTasks.xml."

```
> Frame 634: 171 bytes on wire (1368 bits), 171 bytes captured (1368 bits)
> Ethernet II, Src: Routerbo_fb:5b:68 (74:4d:28:fb:5b:68), Dst: VMware_d7:20:d8 (00:0c:29:d7:20:d8)
> Internet Protocol Version 4, Src: 192.168.3.11, Dst: 192.168.4.7
> Transmission Control Protocol, Src Port: 53626, Dst Port: 445, Seq: 3390706491, Ack: 336006818, Len: 117
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v Read Request (0x08)
    > StructureSize: 0x0031
      Padding: 0x50
    > Flags: 0x00
      Read Length: 1966
      File Offset: 0
    > GUID handle File: paper.local\Policies\{3CA5CE12-7B0C-4B0E-9DAB-8BE253661C02}\User\Preferences\ScheduledTasks\ScheduledTasks.xml
      Min Count: 0
      Channel: None (0x00000000)
      Remaining Bytes: 0
      Blob Offset: 0x00000000
      Blob Length: 0
      Channel Info Blob: NO DATA
```

Figure 11. GPO Immediate Task Packet

After the read request from the host, the domain Controller responds with an SMB Read Response to the request and provides the host with a PowerShell script. Figure 12 displays the script name in the Data Field. Network security tools can detect the traffic as the traffic passes through the network.

Defenders can utilize an Intrusion Prevention System to block the traffic or an Intrusion Detection System to detect the traffic. The following detection rule will detect or block Group Policy Object traffic with the content matching ".ps1":

alert tcp $HOME_NET 445 -> $HOME_NET any (msg:"Immediate Task GPO PowerShell"; content "|2E 70 73 31|";)

The rule is suitable for organizations that do not use PowerShell scheduled tasks from a GPO. Attackers can modify their behavior and implement other Microsoft code-executing scripts; however, the alert will notify defenders of a failed PowerShell GPO attempt.

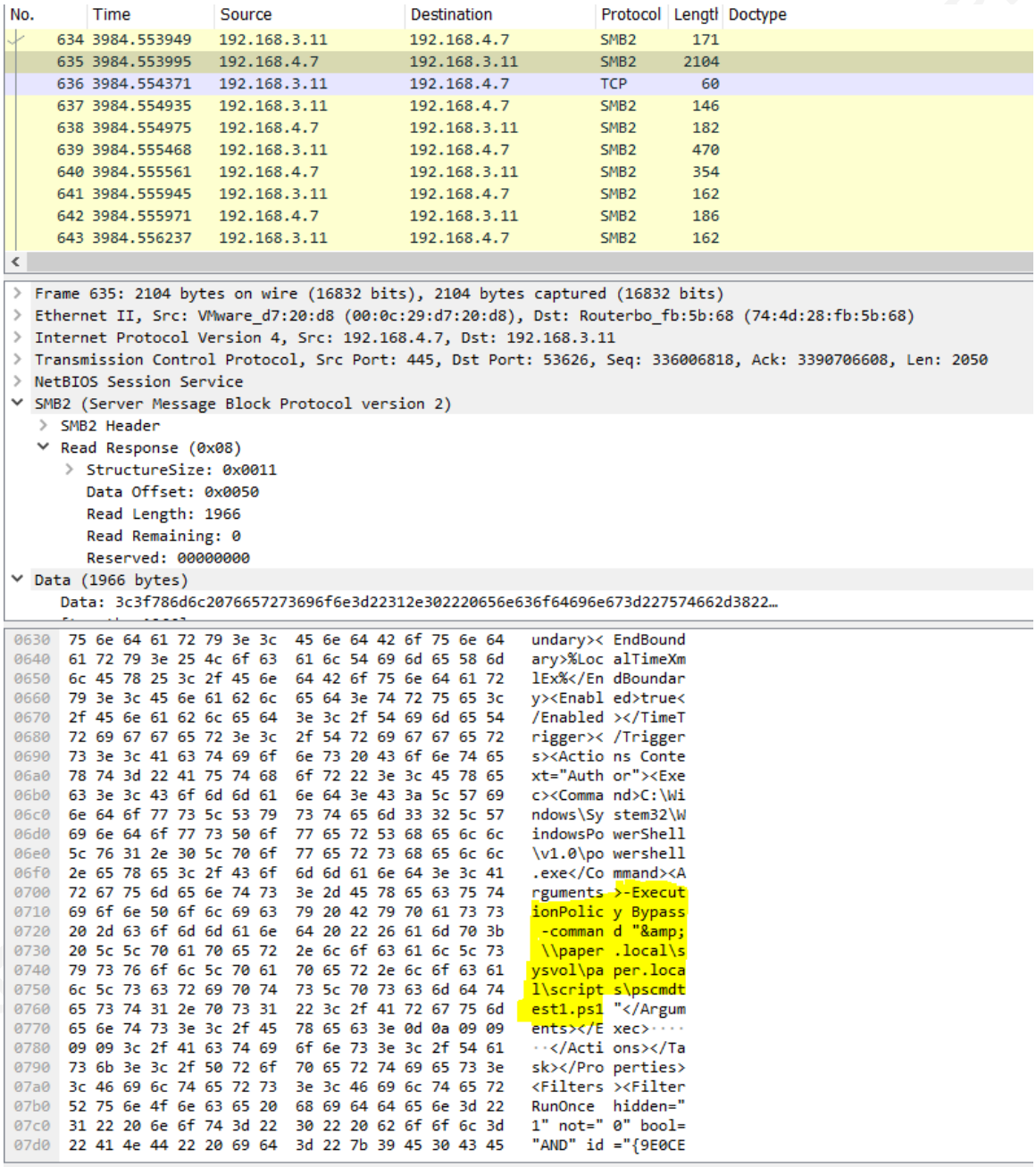


Figure 12. GPO Immediate Task Packet

5. Conclusion

Once a malicious actor gains access to a domain controller, the opportunity to defend the domain exists. Defenders have more options available beyond instructing all users to disconnect workstations. PowerShell remoting is the most effective method to run malicious commands; however, preexisting settings must be enabled. The second

speediest method is to employ immediate tasks. Utilizing log-off scripts and scheduled tasks should be considered the last measure, as some input is required.

While PowerShell remoting is the fastest, defenders must understand each attack vector. Incident responders should include how adversaries deliver ransomware or execute domain attacks. Doing so provides an opportunity for defenders to tailor their defenses. The defender's understanding of the attack vector assists with developing additional signatures. Documenting the behaviors of malicious actors assists in tailoring defenses. If a threat actor uses a specific directory to write world-readable scripts, then more organizations should monitor that activity. Sharing the post-domain compromise actions will increase awareness for defenders.

After an adversary gains access to a domain controller. The potential for a defender to stop the malicious actor defending still exists. Hardening systems to increase defenses improves the amount of time the attacker must reside on a system to achieve their objective. Security Operations Centers should use the findings discussed to tailor response times and playbooks for scenarios where a domain controller is compromised.

References

Biasini, N. (2022, November 2). Cisco Talos shares insights related to recent cyber attack on Cisco. Cisco Talos Blog. Retrieved December 21, 2022, from https://blog.talosintelligence.com/recent-cyber-attack/ Windows Powershell 2.0

The Health and Safety Executive. (2021, December 2). Conti cyber attack on the HSE. The Health and Safety Executive. Retrieved from https://www.hse.ie/eng/services/publications/conti-cyber-attack-on-the-hse-full-report.pdf

Stewart, B (2010, September).Windows PowerShell 2.0 Remoting. Windows IT Pro.

Mitwirkenden, A. (2021, November 22). Configure Automatic Updates by Using Group Policy. Microsoft Learn https://learn.microsoft.com/de-de/security-updates/windowsupdateservices/18127451

Abrams, L. (2021, July 28). LockBit ransomware now encrypts Windows domains using group policies. BleepingComputer. https://www.bleepingcomputer.com/news/security/lockbit-ransomware-now-encrypts-windows-domains-using-group-policies/

Goliath, D. (2016). M-Trends 2016. Mandiant M-Trends, Special Report February 2016. https://content.fireeye.com/m-trends/rpt-m-trends-2016

Tudor, D. (2022, December 7). All about Conti Ransomware. Heimdal Security Blog. https://heimdalsecurity.com/blog/what-is-conti-ransomware/

Start-Sleep (Microsoft.PowerShell.Utility) - PowerShell. (n.d.). Microsoft Learn. https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/start-sleep?view=powershell-7.3

Enable-PSRemoting (Microsoft.PowerShell.Core) - PowerShell. (n.d.). Microsoft Learn. https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7.3

Rickard, J., (2017, June 13). Run PowerShell scripts as Immediate Scheduled Tasks with Group Policy. 4sysops. https://4sysops.com/archives/run-powershell-scripts-as-immediate-scheduled-tasks-with-group-policy/

Allen, R. (2022, September 30). Group Policy: The Ultimate Guide. Active Directory Pro. https://activedirectorypro.com/group-policy-guide/

How to Audit Group Policy Changes using the Security Event Log. (n.d.). Netwrix. https://www.netwrix.com/group_policy_modification_using_logs.html

Microsoft Corporation. (2021). [MS-PSRP]. PowerShell Remoting Protocol, 15. https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-PSRP/[MS-PSRP].pdf