WST-2 Instructor Setup

# Schedule

## IOR Setup

1. Configure pfSense to capture traffic on the appropriate network.
2. Run WST-2 RSoM (*see below*)
3. Verify the below data points:
   * For Sysmon Log validation, search for event ID 8 with the time of Powershell DL&X
   * For Windows Log validation, use the following query to ID initial access with Powershell DL&X: event.code:4688 AND (winlog.event\_data.CommandLine: \*powershell.exe\* OR winlog.event\_data.CommandLine: \*cmd.exe\*)
     + Create query line from PS-Session
   * For PCAP validation use the following query to identify initial stager payload being sent to target: TODO
4. Verify Windows logging policy (specifically rollover)
5. Annotate time stamps of beginning/end of VUL.
6. Update Lab Guide based on any deviations to WST-2 RSoM and including the time stamps from the previous step.
7. Wait at least a full duty day (*OBE*)
   * Depends on Windows Logging Policy
8. Run P&T-2 RSoM (*see below*)
9. Verify the below data points:
   * For Sysmon Log validation, search for event ID 8 with the time of Powershell DL&X
   * For Windows Log validation, use the following query to ID initial access with Powershell DL&X: event.code:4688 AND (winlog.event\_data.CommandLine: \*powershell.exe\* OR winlog.event\_data.CommandLine: \*cmd.exe\*)
     + Create query line from PS-Session
   * For PCAP validation use the following query to identify initial stager payload being sent to target: TODO
10. Annotate time stamps of beginning/end of VUL.
11. Update FRAGO/Student handouts with associated time stamps.

## Students Schedule

* 45min Academics (instructor led)
* 10min Break
* 1hr Lab Guide (student led, instructor assisted)
* 2hr Assessment (student led, instructor troubleshoot only)

# Academics

* Windows Logs
  + Differences between Sysmon/Windows Logs
  + Differences between OS versions
  + How logging policy affects available logs and most common logging policies
* Commands
  + Querying Windows logs via PS-Session using Powershell
  + Querying Windows logs via PS-Session using MSDOS
  + Querying PCAP via Wireshark
* Basic Collection & Analysis
  + “Create a hypothesis” -> “Identify data points that support your hypotheses” -> “What to do with the data” -> “How to move forward”
  + “Use previous experience, but most of all use logic”
* Case Study
  + Given a scenario of “Indirect Command Execution” identify collection/queries
  + How this information drives conclusions/investigations
* Lab Guide/Assessment
  + Provide directions to students on accomplishing the lab guide and assessment

# Lab Guide

* Identify suspicious scheduled tasks on Client 3
* Identify suspicious HTTP traffic from “suspicious IP” to Client 3

# Assessment

* What port did the adversary use for C2?
  + 8081
* What was the C2 IP address?
  + X.X.X.254
* What port did the adversary use for persistence?
  + 8080
* What was the application used for initial access?
  + Powershell.exe
* What was the initial access application renamed to?
  + WINWORD.exe
* What was the initial target?
  + Client1
* What was the second target?
  + DC01
* What exploit was used to access the second target?
  + Eternal Romance
* What was the command used on the second target for persistence?
  + net user /add “BrockHard”
* What was the artifact left behind?
  + Brock Hard
* Besides Persistence, what other MITRE tactic was accomplished on the second target?
  + Credential Access
* What was the third target?
  + Client3
* What was the malicious payload uploaded for persistence?
  + ChromeUpdater.exe
* What was the command line argument used to execute persistence?
  + schtask /TN …
* What file(s) were exfiltrated
  + viccash-bin.c

# WST-2 RSoM

## Pastables

gitlab.wicnet.mil/32WPS/WHITECELL/-/blob/master/Automated\_Check\_Out/WST2\_RSOM\_Manual

## Overview

1. Turn spool on
2. Create 2 meterpreter handlers
   1. Reverse\_http on port 8080 (for Persistence)
   2. Reverse\_http on port 8081 (for C2)
3. Generate payload using the reverse\_http 8080 handle as “ChromeUpdate.exe”
4. Host a python webserver in the folder “ChromeUpdate.exe” was created in
5. Execution on “Client 1”
   1. Copy C:\Windows\System32\WindowsPowerShell\v1\powershell.exe to C:\Program Files\Microsoft Office\Office16\ as WINWORD.exe (replace or rename WINWORD.exe)
   2. Use PowerShell download and execute, masqueraded as WINWORD.exe, to download and execute “ChromeUpdate.exe”
6. On “Client1”
   1. Run initial checks (getpid, getuid, sysinfo, ps)
   2. Enumerate user/group, schtasks and services
7. Pivot to “DC01” by adding a route to 192.168.10.0/24 through session 1
8. Exploit “DC01” via EternalRomance using the reverse\_http handler
9. On “DC01”
   1. Run initial checks (getpid, getuid, sysinfo, ps)
   2. Run hashdump on
   3. Add user “BrockHard” to the domain
   4. Add user “BrockHard” to the “Domain Admins” group
   5. Pivot to “Client3” via PSExec using the credentials (and hash) received in the previous hashdump using the 8081 http handler as VIC\a.start:*hash*
10. On “Client3”
    1. Run initial checks (getpid, getuid, sysinfo, ps)
    2. Enumerate user/groups, schtasks and services
    3. Upload “ChromeUpdater.exe” to C:\Windows\Temp
    4. Create a scheduled task to run “on boot” as SYSTEM and execute “C:\Windows\Temp\ChromeUpdater.exe”
    5. Inject into explorer process to use credentials of logged in user to access share
    6. List the directories of [\\fs01](file:///\\fs01)
    7. Navigate to [\\192.168.10.12\Development\VicNet](file:///\\192.168.10.12\Development\VicNet) Budgeter Pro Prime\viccash\viccash-bin.c and download this file
11. Kill all sessions
12. Turn spool off

## Rationale

This SoM is distinctly different from P&T-2 in that it does not target the Windows XP device at all, uses a http handler instead of TCP, uses a scheduled task instead of a registry run key, and accesses shares via query of the fileserver by IP instead of mounted shares. This should provide enough distinct indicators in Sysmon/WinEvent/Network Traffic logs to allow students to see a similar SoM without getting all the answers the day prior.

# P&T-2 RSoM

## Pastables

gitlab.wicnet.mil/32WPS/WHITECELL/-/blob/master/Automated\_Check\_Out/PnT2\_RSOM\_Manual

## Overview

1. Turn spool on
2. Create 3 meterpreter handlers
   1. Reverse\_http on port 8080
   2. Reverse\_tcp on port 8081
   3. Reverse\_tcp on port 22188
3. Generate payload using the last reverse\_tcp handle as “EdgeUpdate.exe”
4. Execute “EdgeUpdate.exe” on target1 (move using “Copy Files to VM.ps1”)
   1. *Note: The initial target is unclear from pastables*
5. On target1
   1. Enumerate user/group, schtasks and services
6. Pivot to “Client5” by adding a route to 192.168.10.0/24 through session 1
7. Exploit “Client5” via NetAPI using the reverse\_tcp handler
8. On “Client5”
   1. Run hashdump
   2. Pivot to “Client3” via PSExec using the credentials (and hash) received in the previous hashdump using the reverse\_tcp handler
      1. Domain – vic
      2. Smbuser – a.stark
      3. Smbpass – *hash*
9. On “Client3”
   1. Run initial checks (getpid, getuid, sysinfo, ps)
   2. Enumerate user/groups, schtasks and services
   3. Add user JackHammer to the domain
   4. Add user JackHammer to the “Domain Admins” group
   5. Upload “EdgeUpdate.exe”
   6. Create a registry run key to execute “EdgeUpdate.exe” in the C:\Windows\Temp folder
   7. Use meterpreter multi\_meterpreter\_inject to inject into the explorer process and call back to the reverse\_tcp handler (same handle for everything)
   8. Enumerate mounted volumes
   9. Find the target file in Y:\VicNet Budgeter Pro Prime\viccash\viccash-bin.c
   10. Move target file to viccash-bin-good.c as an interim measure
   11. Upload “evil” target file as the original filename
   12. (?) Remove viccash-bin-good.c
       1. *Note: This step is currently missing from pastables*
10. Kill all sessions
11. Turn spool off

# Appendix A

## WST-2

### Collection Plan/ Hypothesis Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hypothesis | Tactic | Result | Truth Data | Error Type |
| APT uses Drive-by Compromise through compromising trusted domains website | Initial Access | N/A | APT uses a combination of Cmd Line and Powershell to obfuscate actions and download malicious files | N/A |
| APT uses Indirect Command Execution to obfuscate actions on target | Defense Evasion | N/A | APT uses WINWORD.exe to masquerade use of Powershell | N/A |
| APT uses Data Obfuscation via MetaSploit framework to masquerade C2 traffic as encrypted HTTP | C2 | N/A | APT uses Metasploit HTTP payload to communicate | N/A |
| APT uses Account Discovery to identify accounts who can access their targeted resources | Discovery | N/A | APT uses command line utilities to enumerate valid accounts on the domain | N/A |
| APT uses Command and Scripting Interpreter to execute command line tools (LOLBins) on network | Execution | N/A | APT uses powershell.exe to execute payloads | N/A |
| APT uses Process Injection to escalate privileges | Privilege Escalation | N/A | APT uses meterpreter injects to masquerade as other users and escalate local privileges | N/A |
| APT uses Remote Services via PSExec to move payloads onto another system | Lateral Movement | N/A | APT uses PSExec to move MSF payload to target and remotely execute for lateral movement | N/A |
| APT creates domain accounts to allow access to the target network | Persistence | N/A | APT creates a domain account with appropriate permissions to access network | N/A |
| APT uses Scheduled Tasks as a backup persistence method | Persistence | N/A | APT creates a scheduled task to execute malicious code on system boot | N/A |
| APT uses Exfiltration over C2 Channel to steal target data | Exfiltration | N/A | APT uses meterpreter payload download commands to steal data through C2 channel |  |

### Analysis Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tactic | Detection Tool | Detection | Data Source | Analysis Method |
| Execution/ C2 | SecurityOnion  (KQL) | event.code:4688 AND (winlog.event\_data.CommandLine: \*powershell.exe\* OR winlog.event\_data.CommandLine: \*cmd.exe\*) | Windows Log | * Non Admin users utilizing cmd.exe/powershell.exe * PowerShell using obfuscation methods |
| Privilege Escalation | SecurityOnion  (KQL) | event.code: 8 | Sysmon | It is atypical for most windows processes to CreateRemoteThreads, any results are anomolous |
| Lateral Movement/ C2 | SecurityOnion  (KQL) | event.code: 3 AND NOT (destination.hostname: dc01.vic.net OR destination.hostname: fs01.vic.net) | Sysmon | It is atypical for hosts to communicate with other hosts. Search for source/destination IPs that are outside vic.net. |
| Lateral Movement | SecurityOnion (KQL) | event.code:4624 AND winlog.event\_data.LogonType: 3 AND user.name:\* (Add NOT filters for ComputerAccounts e.g. CLIENT1$) | Windows Logs | Identify non-computer accounts remotely accessing based on:   * Target Machine (accessing * User Role (only administrators should be using remote access tools) |
| Lateral Movement | SecurityOnion (KQL) | Event.code:4697 | Windows Logs | Query for all services created and monitor the Service Name (winlog.event\_data.ServiceName) for random strings. |
| Persistence | SecurityOnion (KQL) | Event.code:4720 | Windows Logs | Any domain accounts created should be considered anomalous after deconfliction with mission partner |
| Persistence | SecurityOnion (KQL) | Process.child.command\_line:\*schtasks\* | Windows Logs | Any scheduled tasks being created with malicious/unknown target executables should be investigated |
| Exfiltration | SecurityOnion | **Network Traffic** | PCAP | Utilize time based analysis based on previous indicators to locate adversary activity, then isolate specific hosts affected, then identify largest network sessions, then correlate time with files accessed |

### Artifact List

|  |  |  |
| --- | --- | --- |
| Host | Data Type | Artifact Location |
| Client1 | File | C:\Program Files\Microsoft Office\Office\16\WINWORD.exe (replaced by powershell.exe) |
| Client1 | Events | *Covered in Analysis Plan* |
| DC01 | Events | *Covered in Analysis Plan* |
| DC01 | User | B.Hard added to “Domain Admins” |
| Client3 | Events | *Covered in Analysis Plan* |
| Client3 | SchTask | Task Name: “Chrome Update Service” |
| Client3 | File | C:\Windows\Temp\ChromeUpdater.exe (msf payload) |

## P&T-2

### Collection Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hypothesis | Tactic | Result | Truth Data | Error Type |
|  |  |  |  |  |
|  |  |  |  |  |

### Analysis Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Hypothesis | Tool | Detection | Analysis Method |
|  |  |  |  |
|  |  |  |  |

## Range

4 x copies of the below VM’s

### VM’s

#### Management

* SecurityOnion
* Windows Defense Server1 (Win11)
* Windows Defense Server2 (Win11)

#### VIC.Net

* Client1 (Win10)
* Client2 (Win10)
* Client3 (Win10)
* Client5 (XP Sp3)
* DC01 (Win 2k16)
* FS01 (Win 2k16)
* pfSense

### White Cell

* Kali

### Snapshots

Use “Sensor Deployment” snapshot for reverting the host machines before running the RSoM, if necessary. The students should not require any reverts during the WST or P&T as they are only working off historical data.

## Sensor/Security Onion Configuration

*OBE, no longer introducing SIEM during P&T*

## PCAP Configuration

\*Include PFSense PCAP capture, and share drive location that is accessible from OpStation\*