Today is the log analysis of network traffic. The story goes as follows:

An employee has taken revenged for being demoted, he installed the CrypTOYminer, a malware he downloaded from the dark web, on all workstations and servers. The malware includes a data stealing functionality, which the malware author benefits from. The malware has been executed, and now, a lot of unusual traffic is being generated. Large bandwidth of data is seen leaving the network.

**Learning Objectives:**

1. Understanding log files & their importance.
2. Understanding what a proxy is and breaking down the contents of a proxy log.
3. Building a linux command-line skills to parse log entries manually.
4. Analysing a proxy log based on typical use cases.

**What is a Log File?**

A log file is a digital trail of what’s happening behind the scenes in a computer or software application. It records important events, actions, errors, or information as they happen. It helps diagnose problems, monitor performance, and record what a program or application is doing.

**Example**: Apache web server log.

158.32.51.188 - - [25/Oct/2023:09:11:14 +0000] "GET /robots.txt HTTP/1.1" 200 11173 "-" "curl/7.68.0"

|  |  |  |
| --- | --- | --- |
| **Field** | **Value** | **Description** |
| **Source IP** | 158.32.51.188 | Source that initiated the HTTP request. |
| **Timestamp** | [25/Oct/2023:09:11:14 +0000] | Date and time when event occurred. |
| **HTTP Request** | GET /robots.txt HTTP/1.1 | Actual HTTP request made, including the request method, URI path, and HTTP version. |
| **Status Code** | 200 | Response of the web application. |
| **User Agent** | curl/7.68.0 | User agent used by the source of the request. Typically tied up to the application used to invoke the HTTP request. |

**Why is a proxy server important?**

A proxy server offers enhanced visibility into network traffic and user activities since it logs all web requests and responses. This enables system administrators and security analysts to monitor which websites users will access, when, and how much bandwidth is used. It also allows administrators to enforce policies and block specific websites or content categories.

Common malicious activity that can be logged on a proxy server are:

|  |  |
| --- | --- |
| **Attack Technique** | **Potential Indicator** |
| Download Attempt of malicious binary | Connection to a known malicious URL binary. |
| Data exfiltration | High count of outbound bandwidth due to file upload (*outbound connection to OneDrive*). |
| Continuous C2 connection. | High count of outbound connections to a single domain in regular intervals (*connections every five minutes to a single domain*). |

To make life simpler, and easier to identify the malicious activity, first you must understand the layout of the logs.

**Layout**:

Timestamp – source\_ip – domain:port – http\_method – http\_uri – status\_code – response\_size – user\_agent.

**Question 1) How many unique IP addresses are connected to the proxy server?**

-d ‘ ’ means delimeter is ‘ ‘ (blank).

-f2 means the field position (this case the second position in the log rows).

Sort – sorts the output.

Uniq – finds the unique output.

Wc -l – word count tool, counts the number of lines.

**Answer**: 9.

**Question 2) How many unique domains were access by all workstations?**

The first part of the command displays the access log.

Second part will retrieve the domain and the port.

The third part will remove the port by using the colon as the delimiter, so we obtain only the domain.

The next part is sort is not necessary, only the uniq and last bit which will count the number of lines outputted.

**Answer**: 111

**Question 3) What status code is generated by the HTTP requests to the least access domain?**

 I first needed to separate the domain and display the number of connections that each domain got. I did this by first obtain all the domains (-f3) and then set up a new delimiter to find the port accessed. Then sort them, and then count all the unique domains, but then I sort again based on the count.

Once I have the domain, I can then find all mentions of that domain and find the parameter 6 (Status code).

**Answer**: 503.

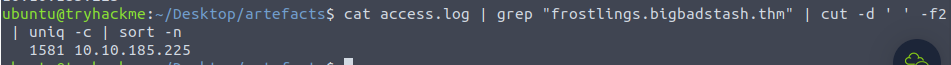
A close up of a computer screen

Description automatically generated**Question 4) Based on the high count of connection attempts, what is the name of the suspicious domain?**

The command looks for the domain value (field 3) and then finds the source IP and sorts them, then only retrieves the unique values. I then sort again based on numbers, then I find allow of the unique based on count (yes I didn’t mean to add two sort -n).

I did think that the malicious domain would have a lot of connections since the prompt mentioned that all workstations were infected with the malware.

**Answer**: frostlings.bigbadstash.thm

**Question 5) What is the source IP of the workstation that accessed the malicious domain?**

**Answer**:10.10.185.225

A close up of a computer screen

Description automatically generated**Question 6) How many requests were made on the malicious domain in total?**

The command looks for the domain value (field 3) and then finds the source IP and sorts them, then only retrieves the unique values. I then sort again based on numbers, then I find allow of the unique based on count (yes, I didn’t mean to add two sort -n).

**Answer**: 1581

**Question 7) Having retrieved the exfiltrated data, what is the hidden flag?**

A blue screen with white text

Description automatically generated

The command above takes the URI and takes only the base64 part.

A close up of a computer screen

Description automatically generated

The common flag used by Try Hack Me is THM, so I decided to use grep to find the flag.

**Answer**: THM{a\_gift\_for\_you\_awesome\_analyst!}