Incident handling with Splunk

This lab was provided by the TryHackMe platform, and aims to demonstrate in a practical way the use of Splunk for incident handling.

Scenario

A Big corporate organization **Wayne Enterprises** has recently faced a cyber-attack where the attackers broke into their network, found their way to their web server, and have successfully defaced their website. Their website is now showing the trademark of the attackers with the message **YOUR SITE HAS BEEN DEFACED** as shown below.

They have requested me to join them as a Security Analyst and help them investigate this cyber attack and find the root cause and all the attackers' activities within their network.

I need to explore the records and find how the attack got into their network and what actions they performed.

Reconnaissance Phase

Reconnaissance is an attempt to discover and collect information about a target. It could be knowledge about the system in use, the web application, employees or location, etc.

I will start our analysis by examining any reconnaissance attempt against the webserver imreallynotbatman.com

```
1   index=botsv1 imreallynotbatman.com
```
✓ **78,683 events** (before 10/18/25 2:33:54.000 PM)   No Event Sampling ▾
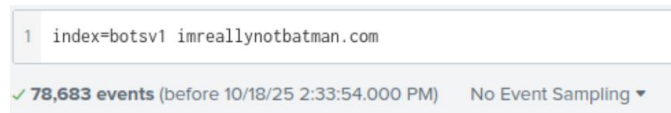
And what source types we have.

Values	Count	%	
suricata	30,625	38.922%	
stream:http	22,200	28.214%	
fortigate_utm	13,918	17.689%	
iis	11,940	15.175%	

Looking at the log source stream:http, which contains the http traffic logs, and examine the src_ip field from the left panel.

```
1   index=botsv1 imreallynotbatman.com sourcetype=stream:http
```

So far, i have found two IPs in the src_ip field 40.80.148.42 and 23.22.63.114. The first IP seems to contain a high percentage of the logs as compared to the other IP, which could be the answer.

Values	Count	%	
40.80.148.42	17,483	93.402%	
23.22.63.114	1,235	6.598%	

I have narrowed down the results to only show the logs from the source IP 40.80.148.42, looked at the fields of interest and found the traces of the domain being probed.

Validate the IP that is scanning

This query will show the logs from the suricata log source that are detected/generated from the source IP 40.80.248.42

```
1  index=botsv1 imreallynotbatman.com sourcetype=suricata src_ip="40.80.148.42"
```

Q1. One suricata alert highlighted the CVE value associated with the attack attempt. What is the CVE value?

I added "CVE-" to the search query to display only alerts that have CVE associations in their metadata. Out of the 46 total alerts, only three contained CVE values. This made it easy to identify that the alert highlighting a CVE was associated with CVE-2014-6271

ET WEB_SERVER Possible CVE-2014-6271 Attempt	18	46.154%	
ET WEB_SERVER Possible CVE-2014-6271 Attempt in Headers	18	46.154%	

Q2. What is the CMS our web server is using?

I filtered the search results by the most common CMS indicators and analyzed the remaining events. After reviewing the URI patterns and signatures, I identified that the website in this case is using Joomla.

```
APPPOOL\joomla</Data><Data
```

Q3. What is the web scanner, the attacker used to perform the scanning attempts?

I added the keyword "scan" to my query to filter the alerts related to scanning activity, this way i easily found out that the web scanner used was Acunetix.

```
"ET SCAN Acunetix Accept HTTP Header
```

Exploitation Phase

The attacker needs to exploit the vulnerability to gain access to the system/server.

In this task, i will look at the potential exploitation attempt from the attacker against our web server and see if the attacker got successful in exploiting or not.

To begin our investigation, let's note the information we have so far:

- We found two IP addresses from the previous phase with sending requests to our server.
- The attacker was using the web scanner Acunetix for the scanning attempt.

Count

I used the following search query to check how many requests were sent to our web server.
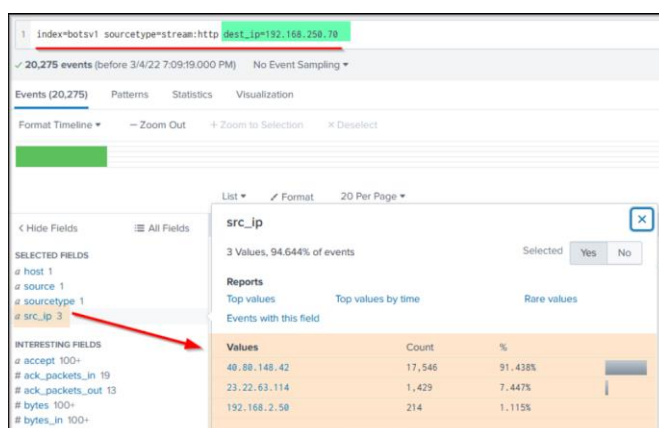
To see what kind of traffic is coming through the POST requests, we will narrow down on the field http_method=POST as shown
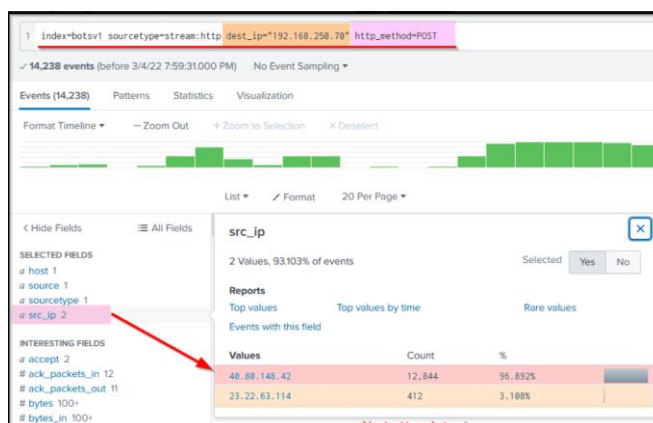
Joomla showed up in fields like uri, uri_path and http_referrer, so the site is using Joomla CMS on the backend.

The admin login page usually appears at /joomla/administrator/index.php.

That URL is the site's login panel, so I'll focus on traffic to that page next to check for brute-force attempts.
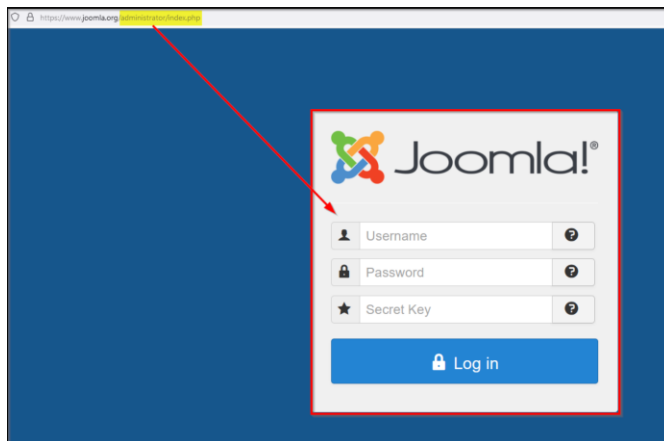
By this query:

```
1  index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70" uri="/joomla/administrator/index.php"
```

We can see that the field form_data has a lot of events, This field contains the requests sent through the form on the admin panel page, which has a login page. The attacker may have tried multiple credentials in an attempt to gain access to the admin panel.

Going a bit deeper, we can see from this table that "username" and "passwd" appear successively with very short time intervals between attempts — this clearly confirms a brute-force attack by the ip 23.22.63.114.

Extracting Username and Passwd Fields using Regex

Looking into the logs, we see that these fields are not parsed properly. Let us use Regex in the search to extract only these two fields and their values from the logs and display them.

We can display only the logs that contain the username and passwd values in the form_data field by adding form_data=*username*passwd* in the above search.

Let's use Regex. rex field=form_data "passwd=(?<creds>\w+)" To extract the passwd values only.

We have extracted the passwords being used against the username admin on the admin panel of the webserver.

If we examine the fields in the logs, we will find two values against the field http_user_agent as shown below:
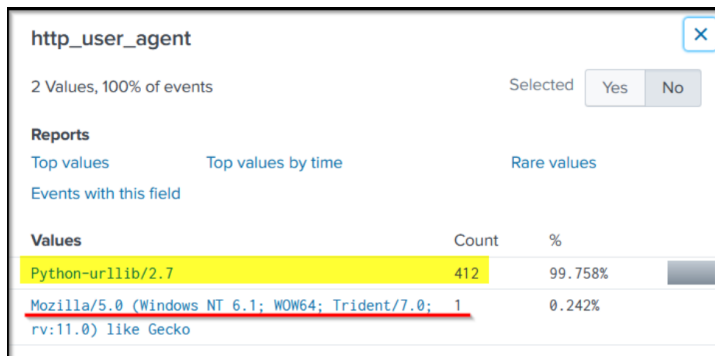
The first value clearly shows attacker used a python script to automate the brute force attack against our server. But one request came from a Mozilla browser.

_time ⇕	src_ip ⇕	url ⇕	http_user_agent ⇕	creds ⇕
2016-08-10 21:48:05.858	40.80.148.42	/joomla/administrator/index.php	Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko	batman
2016-08-10 21:46:33.689	23.22.63.114	/joomla/administrator/index.php	Python-urllib/2.7	batman

The Login was made by the "40.80.148.42".

Installation Phase

Once the attacker has successfully exploited the security of a system, he will try to install a backdoor or an application for persistence or to gain more control of the system.

In the previous Exploitation phase, we found evidence of the webserver iamreallynotbatman.com getting compromised via brute-force attack by the attacker using the python script to automate getting the correct password. The attacker used the IP" for the attack and the IP to log in to the server. This phase will investigate any payload / malicious program uploaded to the server from any attacker's IPs and installed into the compromised server.

First i will narrow down any http traffic coming into the server with the term ".exe ".

```
index=botsv1 sourcetype=stream:http dest_ip="192.168.250.70" *.exe
```

Observing the interesting fields and values, we can see the field part_filename{} contains the two file names. an executable file 3791.exe and a PHP file agent.php.

part_filename{}   ☒

2 Values, 5.882% of events   Selected   Yes   No

Reports

Top values   Top values by time   Rare values

Events with this field

Values	Count	%	
3791.exe	1	100%	
agent.php	1	100%	

The 3791.exe file came from the same IP that successfully logged in: 40.80.148.42.

Q1.Was this file executed on the server after being uploaded?

By analyzing the Sysmon logs related to 3791.exe using the query index=botsv1 "3791.exe" sourcetype="XmlWinEventLog" EventCode=1, it was possible to confirm through the command line field that the file 3791.exe was indeed executed on the compromised server.

Q2. Sysmon also collects the Hash value of the processes being created. What is the MD5 HASH of the program 3791.exe?

```
index=botsv1 "3791.exe" sourcetype=xmlwineventlog EventCode=1 CommandLine="3791.exe" MD5
```

MD5= AAE3F5A29935E6ABCC2C2754D12A9AF0

Q3. Search hash on the virustotal. What other name is associated with this file 3791.exe?

Action on Objective

As the website was defaced due to a successful attack by the adversary, it would be helpful to understand better what ended up on the website that caused defacement.

As the logs do not show any external IP communicating with the server. I change the flow direction to see if any communication originates from the server. What is interesting about the output?

Usually, the web servers do not originate the traffic. The browser or the client would be the source, and the server would be the destination. Here we see three external IPs towards which our web server initiates the outbound traffic. There is a large chunk of traffic going to these external IP addresses.
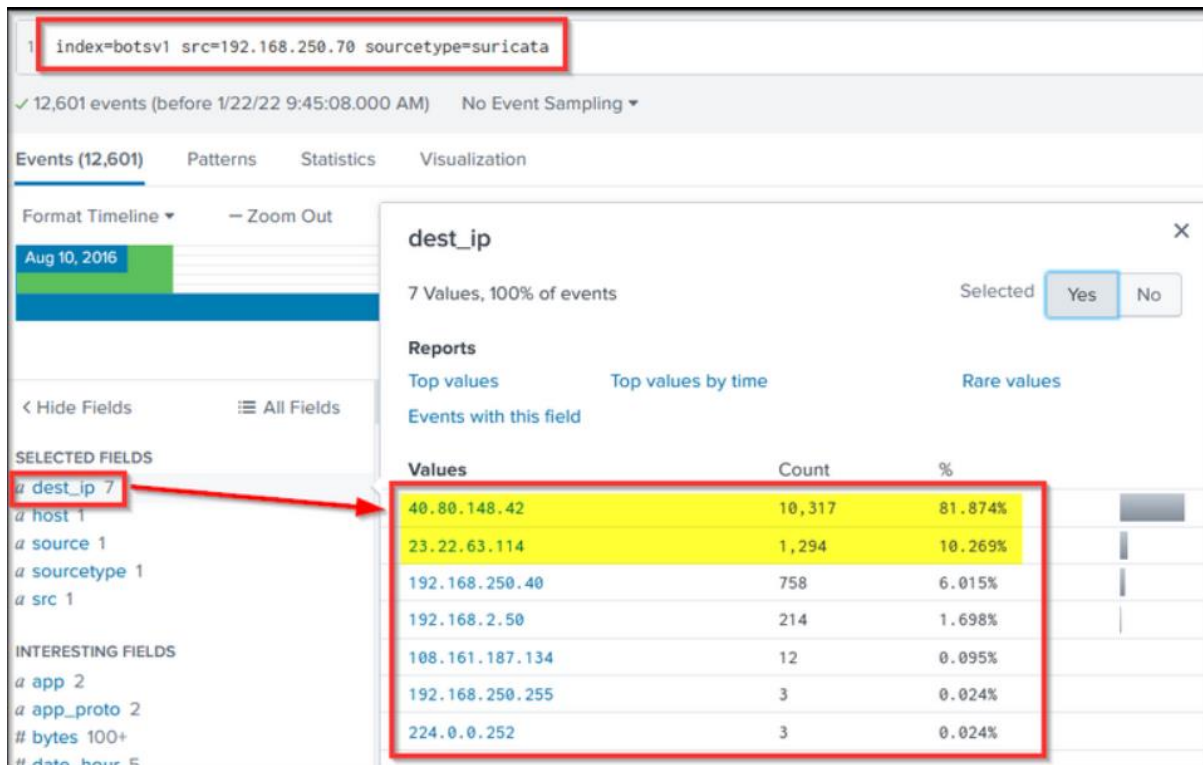
By analyzing the connections with each IP individually, we can see that a ".jpeg" file was installed from the attacker's host prankglassinebracket.jumpingcrab.com that defaced the site.

Command and Control:

The attacker uploaded the file to the server before defacing it. While doing so, the attacker used a Dynamic DNS to resolve a malicious IP. Our objective would be to find the IP that the attacker decided the DNS.

By investigating the fortigate_utm and stream:http sourcetypes, we identified the suspicious domain as a Command and Control (C2) server that the attacker contacted after gaining control of

the compromised server.


```
dest_ip: 23.22.63.114
dest_mac: 08:5B:0E:93:92:AF
dest_port: 1337
duplicate_packets_in: 2
duplicate_packets_out: 0
endtime: 2016-08-10T22:13:46.915172Z
http_method: GET
missing_packets_in: 0
missing_packets_out: 0
network_interface: eth1
packets_in: 6
packets_out: 5
reply_time: 0
request: GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0
request_ack_time: 3246
request_time: 61714
response_ack_time: 0
response_time: 0
server_rtt: 32357
server_rtt_packets: 2
server_rtt_sum: 64714
site: prankglassinebracket.jumpingcrab.com:1337
src_headers: GET /poisonivy-is-coming-for-you-batman.jpeg HTTP/1.0
Host: prankglassinebracket.jumpingcrab.com:1337

src_ip: 192.168.250.70
src_mac: 00:0C:29:C4:02:7E
src_port: 63139
time_taken: 61715
timestamp: 2016-08-10T22:13:46.853458Z
transport: tcp
uri: /poisonivy-is-coming-for-you-batman.jpeg
uri_path: /poisonivy-is-coming-for-you-batman.jpeg
```

Q1. This attack used dynamic DNS to resolve to the malicious IP. What fully qualified domain name (FQDN) is associated with this attack?

```
index=botsv1 sourcetype=stream:dns "23.22.63.114"
08-10T22:06:21.440131Z","timestamp":"2016-08-10
com","prankglassinebracket.jumpingcrab.com"],"re
599,32768],"bytes":162,"src_ip":"192.168.250.20
```
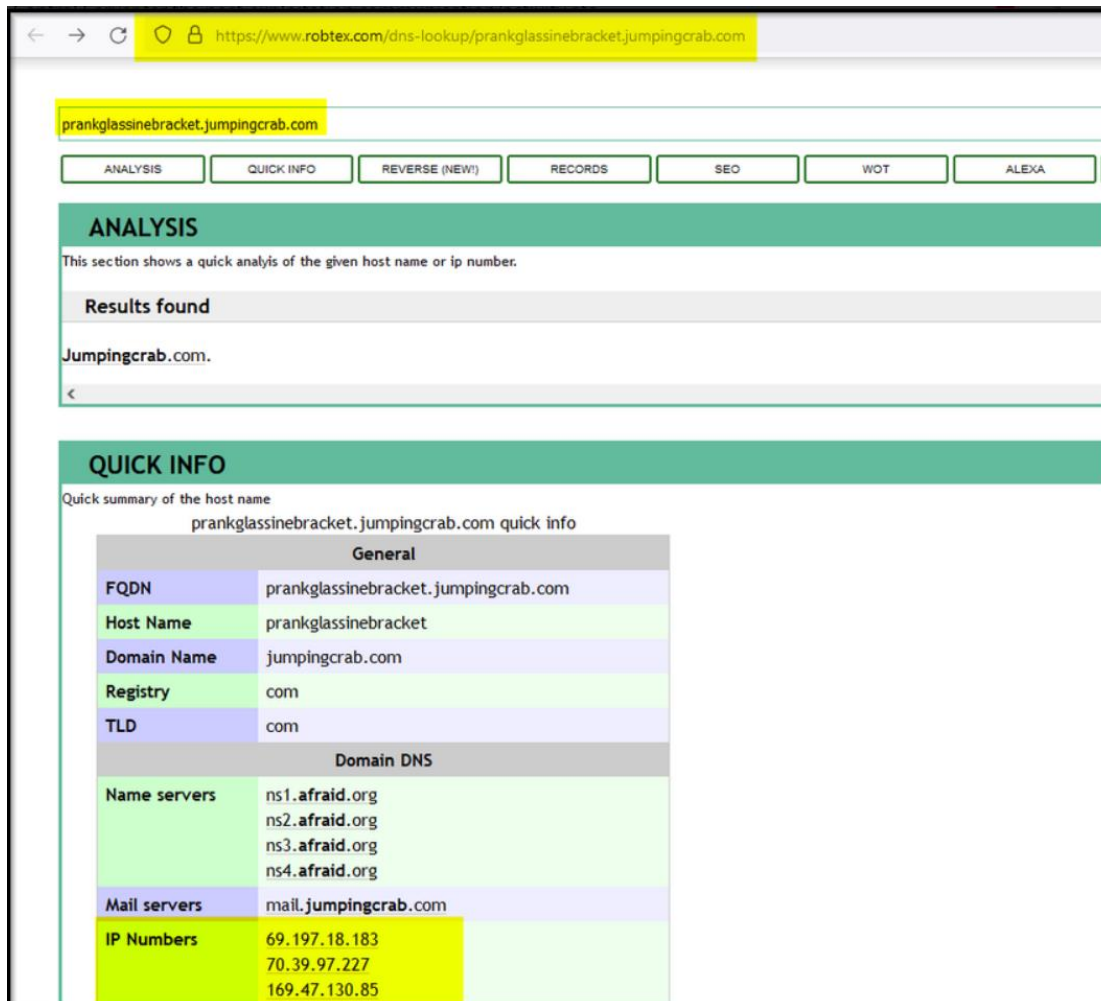
Weaponization

In the weaponization phase, the adversaries would:

- Create Malware / Malicious document to gain initial access / evade detection etc.
- Establish domains similar to the target domain to trick users.
- Create a Command and Control Server for the post-exploitation communication/activity etc.

We have found some domains / IP addresses associated with the attacker during the investigations. This task will mainly look into OSINT sites to see what more information we can get about the adversary.

Robtex:

Robtex is a Threat Intel site that provides information about IP addresses, domain names, etc.

Next, search for the IP address 23.22.63.114 on this Threat Intel site.

Virustotal

Virustotal is an OSINT site used to analyze suspicious files, domains, IP, etc. Let's now search for the IP address on the virustotal site. If we go to the RELATIONS tab, we can see all the domains associated with this IP which look similar to the Wayn Enterprise company.

Delivery

Attackers create malware and infect devices to gain initial access or evade defenses and find ways to deliver it through different means. We have identified various IP addresses, domains and Email addresses associated with this adversary. Now my task is to use the information we have about the adversary and use various Threat Hunting platforms and OSINT sites to find any malware linked with the adversary.

Threat Intel report suggested that this adversary group Poison Ivy appears to have a secondary attack vector in case the initial compromise fails. Our objective is to understand more about the attacker and their methodology and correlate the information found in the logs with various threat Intel sources.

ThreatMiner

By looking for the IP 23.22.63.114 on the Threat Intel site ThreatMiner we found three files associated with this IP, from which one file with the hash value c99131e0169171935c5ac32615ed6261 seems to be malicious.

VirusTotal
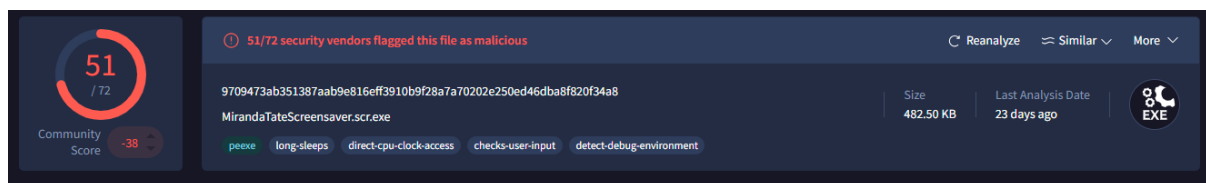
Hybrid-Analysis

Hybrid Analysis is a beneficial site that shows the behavior Analysis of any malware. Here you can look at all the activities performed by this Malware after being executed.

Conclusion:

In this exercise, as a SOC Analyst, i have investigated a cyber-attack where the attacker had defaced a website 'imreallynotbatman.com' of the Wayne Enterprise. We mapped the attacker's activities into the 7 phases of the Cyber Kill Chain.

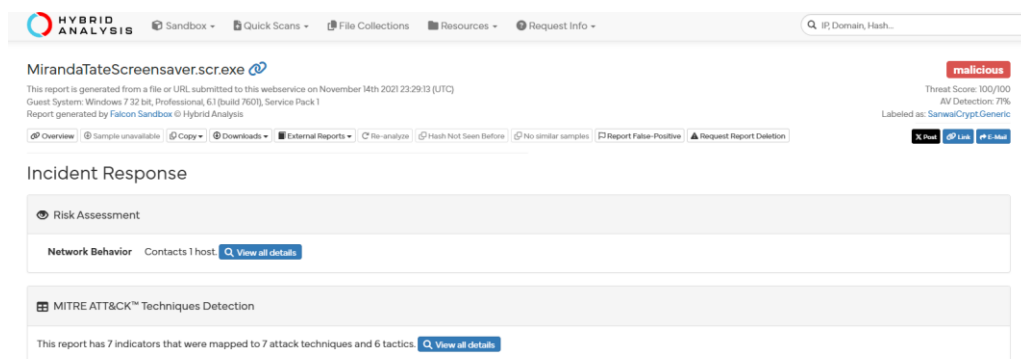## Reconnaissance Phase:

We first looked at any reconnaissance activity from the attacker to identify the IP address and other details about the adversary.

Findings:

IP Address 40.80.148.42 was found to be scanning our webserver.

The attacker was using Acunetix as a web scanner.

Exploitation Phase:

We then looked into the traces of exploitation attempts and found brute-force attacks against our server, which were successful.

Findings:

Brute force attack originated from IP 23.22.63.114.

The IP address used to gain access: 40.80.148.42

142 unique brute force attempts were made against the server, out of which one attempt was successful

Installation Phase:

Next, we looked at the installation phase to see any executable from the attacker's IP Address uploaded to our server.

Findings:

A malicious executable file 3791.exe was observed to be uploaded by the attacker.

We looked at the sysmon logs and found the MD5 hash of the file.

Action on Objective:

After compromising the web server, the attacker defaced the website.

Findings:

We examined the logs and found the file name used to deface the webserver.

Weaponization Phase:

We used various threat Intel platforms to find the attacker's infrastructure based on the following information we saw in the above activities.

Findings:

Multiple masquerading domains were found associated with the attacker's IPs.

An email of the user Lillian.rose@po1s0n1vy.com was also found associated with the attacker's IP address.

Deliver Phase:

In this phase, we again leveraged online Threat Intel sites to find malware associated with the adversary's IP address, which appeared to be a secondary attack vector if the initial compromise failed.

Findings:

A malware name MirandaTateScreensaver.scr.exe was found associated with the adversary.

MD5 of the malware was c99131e0169171935c5ac32615ed6261