Boss Of The SOC v1

CyberDefenders - Boss Of The SOC v1 Write-up (Splunk)

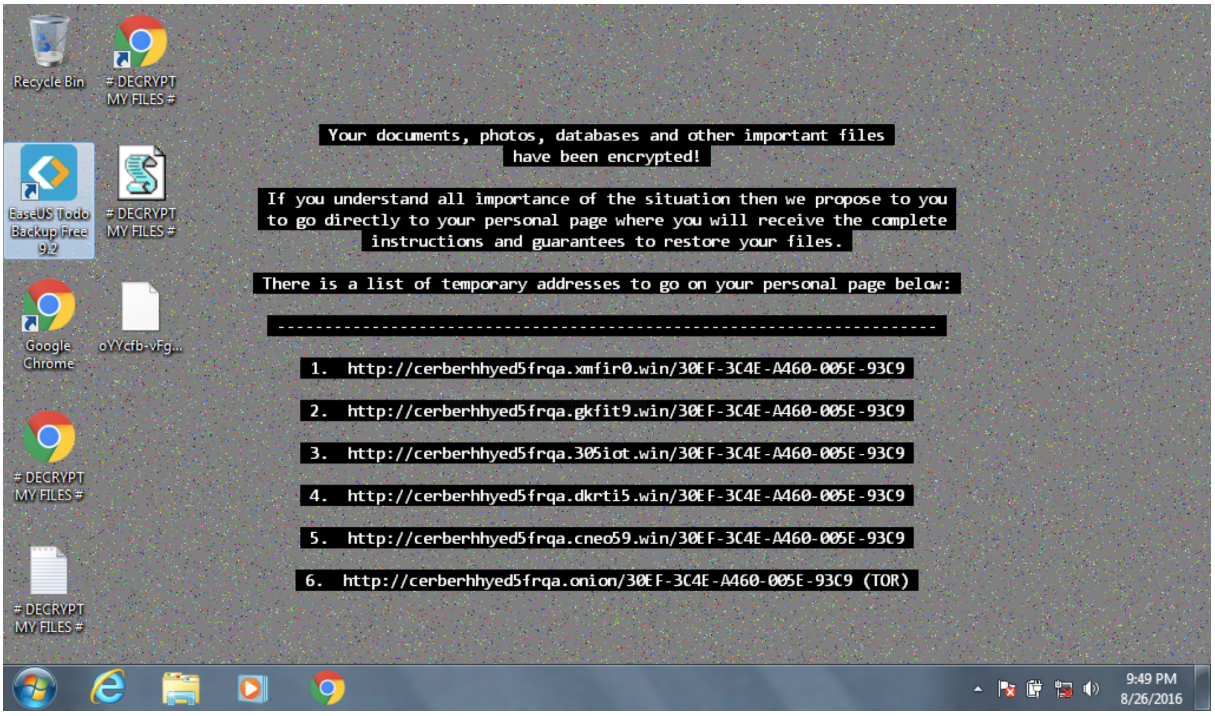
Scenario 1 (APT)

The focus of this hands on lab will be an APT scenario and a ransomware scenario. You assume the persona of Alice Bluebird, the analyst who has recently been hired to protect and defend Wayne Enterprises against various forms of cyberattack.

In this scenario, reports of the below graphic come in from your user community when they visit the Wayne Enterprises website, and some of the reports reference “P01s0n1vy.” In case you are unaware, P01s0n1vy is an APT group that has targeted Wayne Enterprises. Your goal, as Alice, is to investigate the defacement, with an eye towards reconstructing the attack via the Lockheed Martin Kill Chain.

Scenario 2 (Ransomware)

In the second scenario, one of your users is greeted by this image on a Windows desktop that is claiming that files on the system have been encrypted and payment must be made to get the files back. It appears that a machine has been infected with Cerber ransomware at Wayne Enterprises and your goal is to investigate the ransomware with an eye towards reconstructing the attack.



Challenge link - https://cyberdefenders.org/blueteam-ctf-challenges/15

Artifacts Provided

- Splunk VM

Data Summary

Index

- Name - botsv1
- Event Count - 955,807
- Event timeline - 8/10/16, 8/24/16
- Event Sources
  - Windows Event Log (Sysmon, Security, System, Application)
  - Suricata IDS/IPS
  - Fortigate UTM log (syslog)
  - Splunk WinRegistry - view Windows Registry change data
  - Splunk Stream - capture, filter, index, and analyze streams of network event data
  - IIS Server

- Nessus scan

Challenge Questions and Answers

Question 1 - This is a simple question to get you familiar with submitting answers. What is the name of the company that makes the software that you are using for this competition? Just a six-letter word with no punctuation.

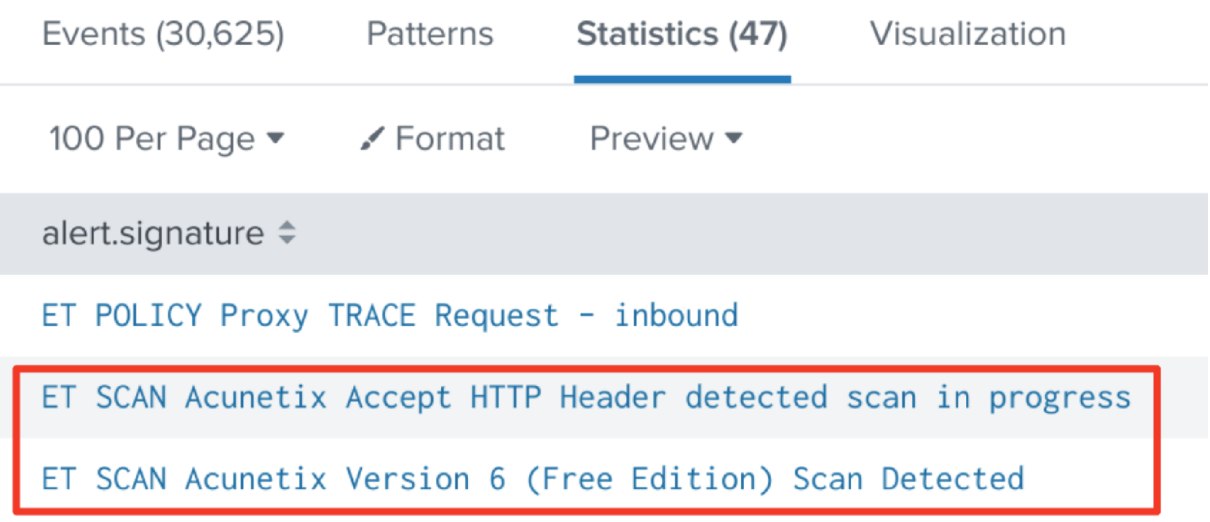
- **Answer: splunk**

Question 2 - What is the likely IP address of someone from the Po1s0n1vy group scanning imreallynotbatman.com for web application vulnerabilities?

- I searched the Suricata log to look for any alerts triggered by the vulnerability scans.
- I filtered the alerts related to the site `imreallynotbatman.com`.
- Search query:

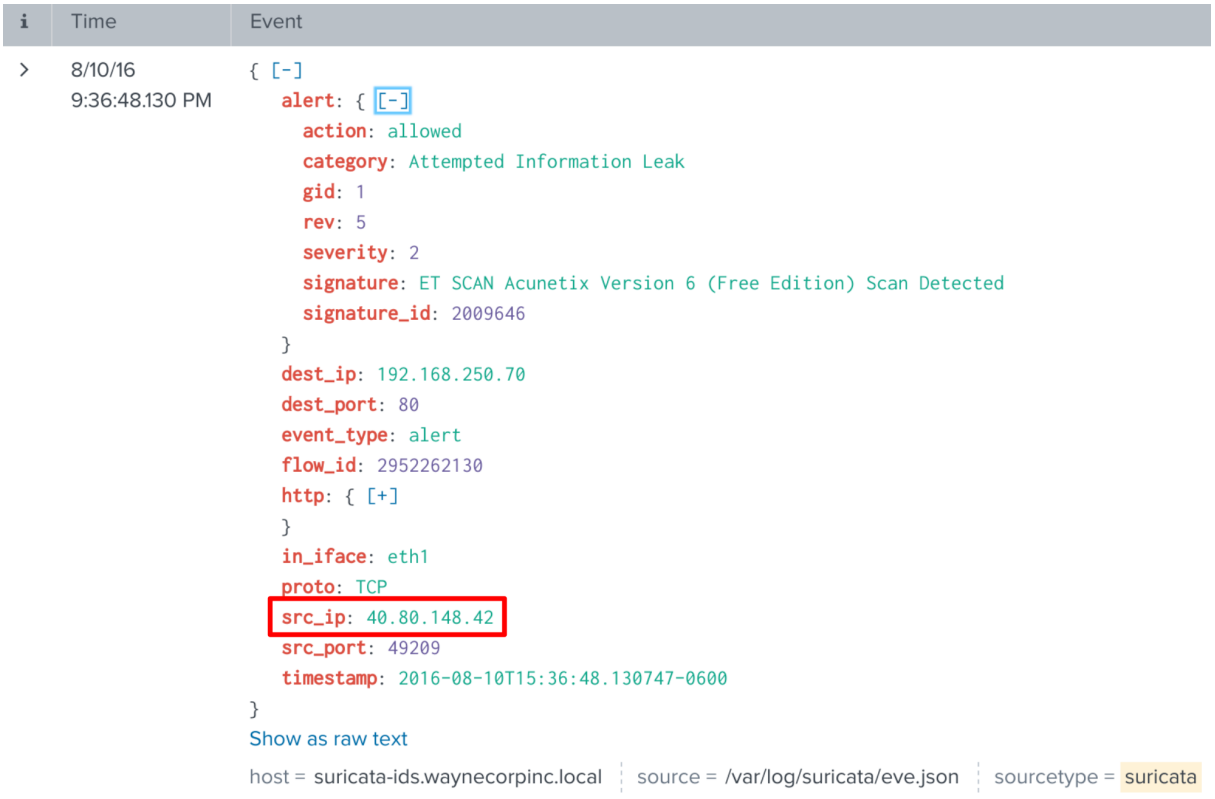
```
index="botsv1" sourcetype="suricata" imreallynotbatman.com
| stats count by alert.signature
```

- The query returned 47 unique alert signatures, which was enough for me to search manually.
- I noticed a signature `ET SCAN Acunetix Version 6 (Free Edition) Scan Detected` and `ET SCAN Acunetix Version 6 (Free Edition) Scan Detected`. Both had the word 'scan' in it.



- I searched the web for Acunetix scan and discovered that Acunetix was a web app vulnerability testing software.

- I viewed the event of one of the scans.



- The source IP address of the scan was `40.80.148.42`.
- **Answer: 40[.]80.148.42**

Question 3 - What company created the web vulnerability scanner used by Po1s0n1vy? Type the company name. (For example, “Microsoft” or “Oracle”)

- As discovered in Question 2, Acunetix was a web app vulnerability testing software.
- **Answer: Acunetix**

Question 4 - What content management system is imreallynotbatman.com likely using? (Please do not include punctuation such as . , ! ? in your answer. We are looking for alpha characters only.)

- For this question, I searched the IIS log and observed the URIs requested.
- Search query:

```
index="botsv1" sourcetype="iis" imreallynotbatman.com
```

- Most events logged contained `/joomla/...` in the URI. Upon searching the web, I found out that Joomla is the content management system for the website.

- **Answer: joomla**

Question 5 - What is the name of the file that defaced the imreallynotbatman.com website? Please submit only the name of the file with the extension (For example, “notepad.exe” or “favicon.ico”).

- My hypothesis was that the threat actor uploaded a webshell by exploiting the vulnerability found during the scan. Using the webshell, they downloaded the defacement file to the web server.
- I searched for `GET` requests coming from the web server. The query returned too many results, and the destination IP address did not contain the known malicious host that initiated the vulnerability scan. I figured that there were more IP addresses controlled by the attacker, so I decided to narrow down the search by filtering out benign IP addresses.
- Since the webshell would interact with the administrative account, I searched for external IP addresses that requested the URI containing `administrator/index.php`. There were two IP addresses - one that initiated the scan and `23.22.63.114`.
- I searched for `GET` requests to the IP address and discovered that the threat actor downloaded a file `poisonivy-is-coming-for-you-batman.jpeg` to the compromised web server. The host had a domain name `prankglassinebracket.jumpingcrab.com`.
- Search query:

```
index="botsv1" sourcetype="stream:http" http_method=GET dest_ip=23.22.63.114
| table timestamp  http_method uri dest_ip site
```

- I searched the Fortigate UTM logs to verify the host. I queried the traffics that were labelled “Malicious Websites”, and I was able to confirm that `prankglassinebracket.jumpingcrab.com` was malicious.

- **Answer: poisonivy-is-coming-for-you-batman.jpeg**

Question 6 - This attack used dynamic DNS to resolve to the malicious IP. What is the fully qualified domain name (FQDN) associated with this attack?

- As discovered in Question 5, the compromised web server requested the defacement file from `prankglassinebracket.jumpingcrab.com`, which resolved to the malicious IP.
- **Answer: prankglassinebracket[.]jumpingcrab.com**

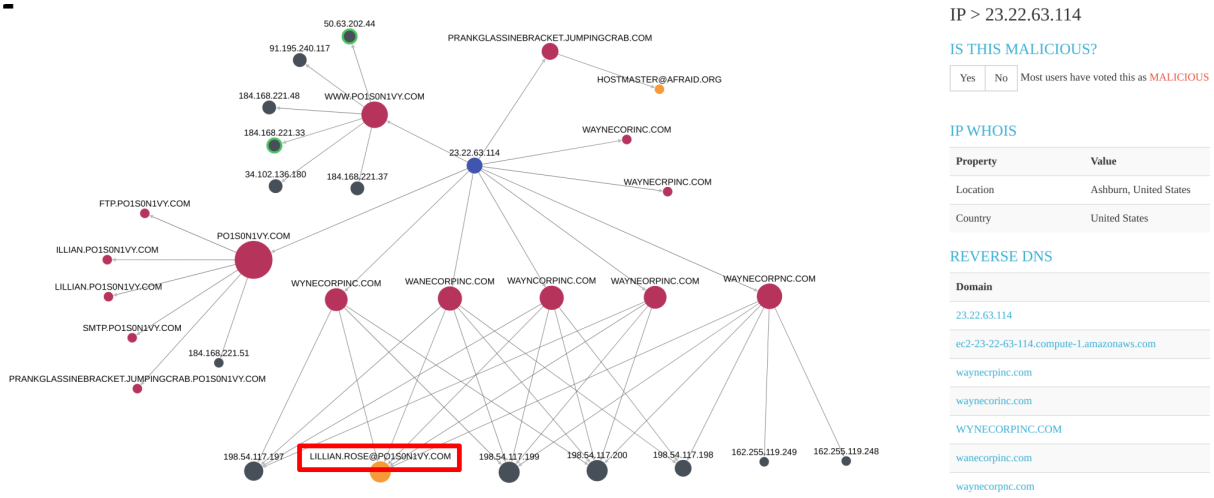
Question 7 - What IP address has Po1s0n1vy tied to domains that are pre-staged to attack Wayne Enterprises?

- As discovered in Question 5, the malicious domain `prankglassinebracket.jumpingcrab.com` resolves to the IP address `23.22.63.114`.
- **Answer: 23[.]22.63.114**

Question 8 - Based on the data gathered from this attack and common open-source intelligence sources for domain names, what is the email address most likely associated with the Po1s0n1vy APT group?

- I used ThreatCrowd to search for email addresses associated with the malicious IP address. I discovered the email address `LILLIAN.ROSE@PO1S0N1VY.COM` associated with the APT group.

-

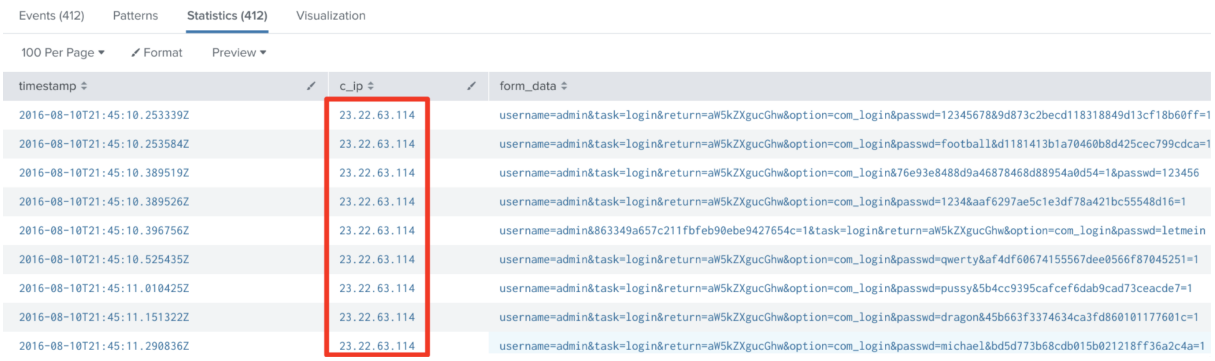


- **Answer: LILLIAN.ROSE[@]PO1S0N1VY.COM**

Question 9 - What IP address is likely attempting a brute force password attack against imreallynotbatman.com?

- Since logging in usually involves `POST` request, I searched the HTTP request to URIs that contain `administrator/index.php`. There query returned many results. I was able to narrow down the search by filtering the suspicious user-agent `Python-urllib/2.7`.
- The query returned 412 requests that contained `username` and `passwd` fields in the form.
- The source IP of the brute force was `23.22.63.114`.
- Search query:

```
index="botsv1" sourcetype="stream:http" http_method=POST uri="*administrator/index.php*"
http_user_agent="Python-urllib/2.7"
| table timestamp c_ip form_data
| sort - timestamp desc
```



- **Answer: 23[.]22.63.114**

Question 10 - What is the name of the executable uploaded by Po1s0n1vy? Please include the file extension. (For example, “notepad.exe” or “favicon.ico”)

- To answer this question, I searched the Sysmon log for `cmd.exe` process creation and its parent process. The thought process is that a legitimiate web server typically does not spawn command line shells.
- The query returned two parent processes - `php-cgi.exe` and `3791.exe`
- `php-cgi.exe`, which looks like a webshell, executed several bash reconnaissance commands such as `ls` and `ifconfig`. It then executed `cmd.exe` to run `3719.exe`, which then also executed `cmd.exe`. This behavior seems malicious.
- Search query:

```
index="botsv1" source="wineventlog:microsoft-windows-sysmon/operational" host="we1149srv"
EventCode=1 Image="*cmd.exe*"
| table UtcTime parent_process cmdline
| sort - UtcTime desc
```

Events (19) Patterns **Statistics (19)** Visualization

100 Per Page ▾ Format Preview ▾

UtcTime	parent_process	cmdline
2016-08-10 21:55:22.945	"C:\Program Files (x86)\PHP\v5.5\php-cgi.exe"	cmd.exe /c "echo 24365"
2016-08-10 21:55:24.133	"C:\Program Files (x86)\PHP\v5.5\php-cgi.exe"	cmd.exe /c "dir 2>&1"
2016-08-10 21:55:26.101	"C:\Program Files (x86)\PHP\v5.5\php-cgi.exe"	cmd.exe /c "ls 2>&1"
2016-08-10 21:55:33.960	"C:\Program Files (x86)\PHP\v5.5\php-cgi.exe"	cmd.exe /c "ifconfig 2>&1"
2016-08-10 21:56:18.142	"C:\Program Files (x86)\PHP\v5.5\php-cgi.exe"	cmd.exe /c "3791.exe 2>&1"
2016-08-10 21:58:23.896	3791.exe	C:\Windows\system32\cmd.exe
2016-08-10 22:05:42.860	"C:\Program Files (x86)\PHP\v5.5\php-cgi.exe"	cmd.exe /c "echo 63059"
2016-08-10 22:08:13.902	3791.exe	C:\Windows\system32\cmd.exe

- I dug further for suspicious process behavior and found that there was a Sysmon Event ID 3 where the process was establishing connection to `23.22.63.114`.
- Search query:

```
index="botsv1" source="wineventlog:microsoft-windows-sysmon/operational" host="we1149srv"
Image="*3791.exe*"  EventCode=3
| table  UtcTime Image EventDescription DestinationIp
| sort  - UtcTime desc
```

Events (1) Patterns **Statistics (1)** Visualization

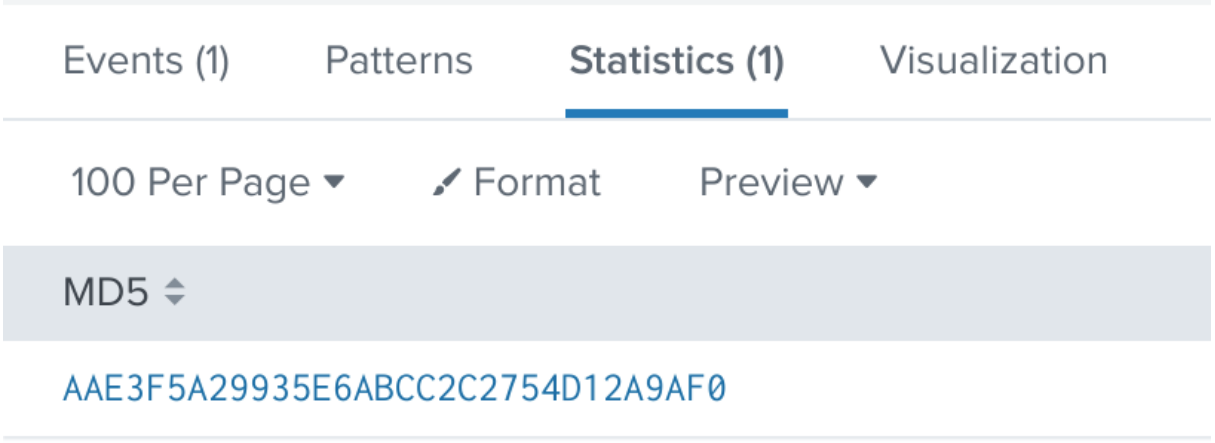
100 Per Page ▾ Format Preview ▾

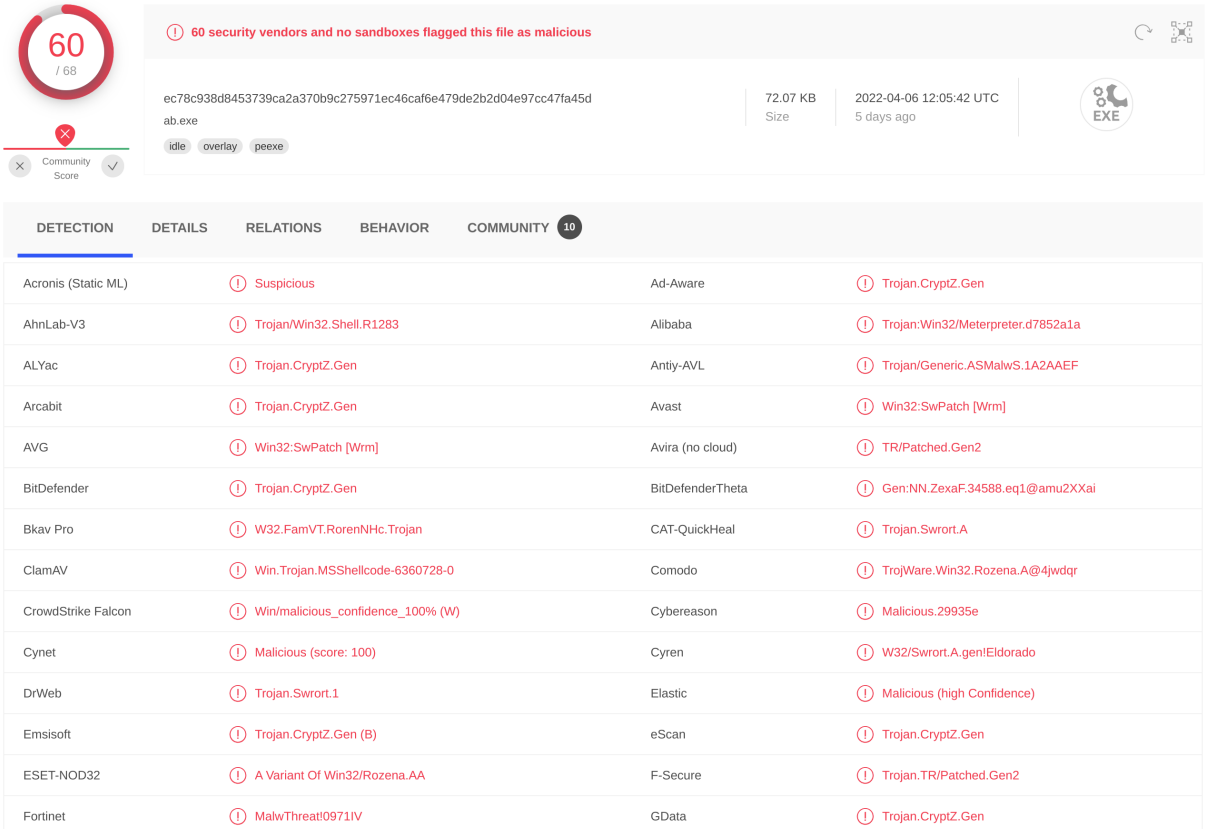
UtcTime	Image	EventDescription	DestinationIp
2016-08-10 12:53:32.270	C:\inetpub\wwwroot\joomla\3791.exe	Network Connect	23.22.63.114

- I extracted the hash value and searched on VirusTotal. The process was indeed malicious.

- Search query:

```
index="botsv1" source="wineventlog:microsoft-windows-sysmon/operational" host="we1149srv"
Image="*3791.exe*"  EventCode=1
| table MD5
```





- **Answer: 3791.exe**

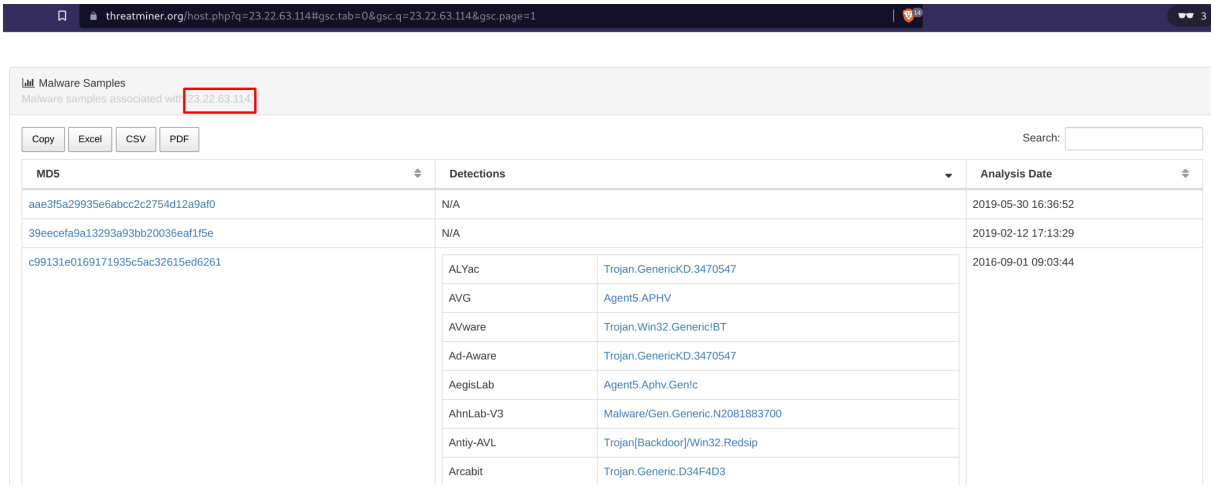
Question 11 - What is the MD5 hash of the executable uploaded?

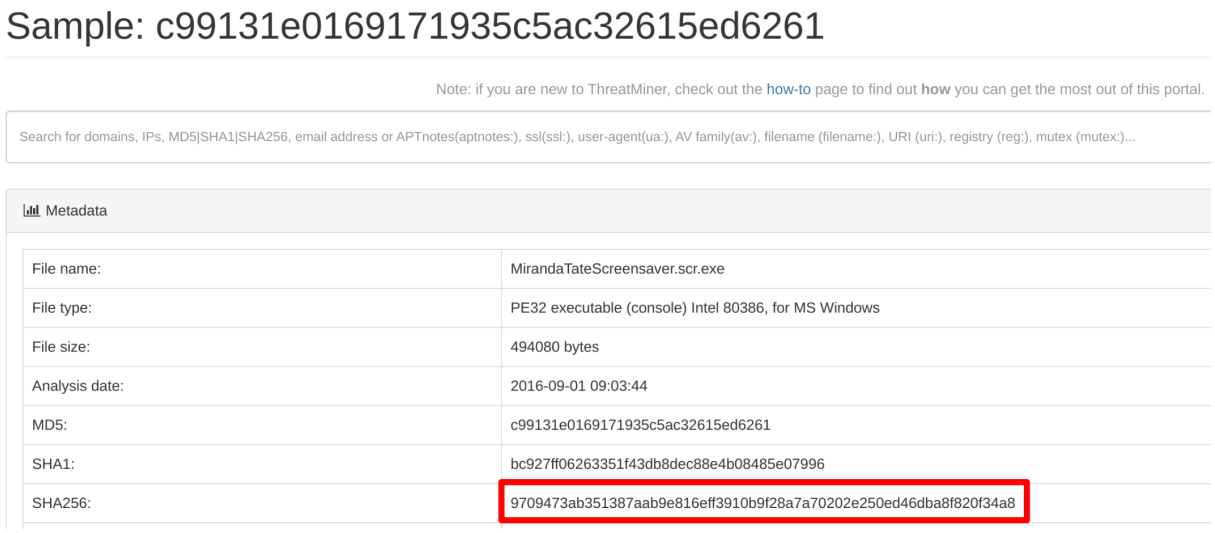
- As discovered in Question 10, the MD5 hash of `3791.exe` is `AAE3F5A29935E6ABCC2C2754D12A9AF0`

- **Answer: AAE3F5A29935E6ABCC2C2754D12A9AF0**

Question 12 - GCPD reported that common TTP (Tactics, Techniques, Procedures) for the Po1s0n1vy APT group, if initial compromise fails, is to send a spear-phishing email with custom malware attached to their intended target. This malware is usually connected to Po1s0n1vy's initial attack infrastructure. Using research techniques, provide the SHA256 hash of this malware.

- I used ThreatMiner to see if there're malware associated with the malicious IP. There was one malware sample with MD5 hash `c99131e0169171935c5ac32615ed6261`. Its SHA256 hash was `9709473ab351387aab9e816eff3910b9f28a7a70202e250ed46dba8f820f34a8`.



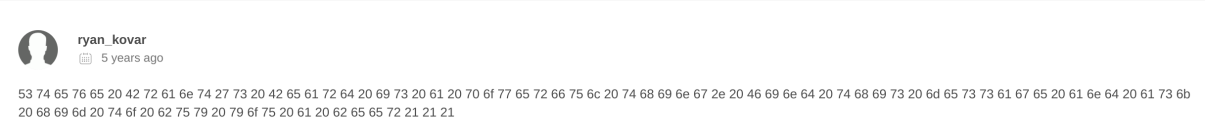


- **Answer: 9709473ab351387aab9e816eff3910b9f28a7a70202e250ed46dba8f820f34a8**

Question 13 - What is the special hex code associated with the customized malware discussed in question 12? (Hint: It's not in

Splunk)

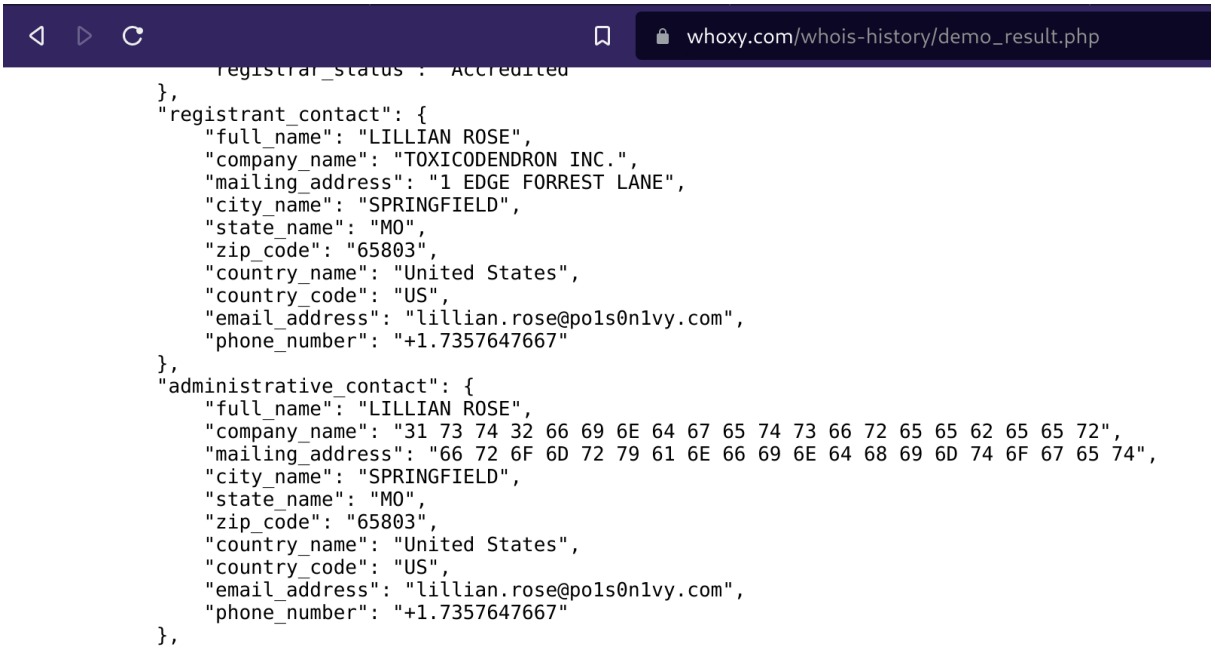
- After digging through VirusTotal, I found the hex code associated in the community section. Coverted to ASCII, the code says “Steve Brant’s Beard is a powerful thing. Find this message and ask him to buy you a beer!!!”.



- **Answer - 53 74 65 76 65 20 42 72 61 6e 74 27 73 20 42 65 61 72 64 20 69 73 20 61 20 70 6f 77 65 72 66 75 6c 20 74 68 69 6e 67 2e 20 46 69 6e 64 20 74 68 69 73 20 6d 65 73 73 61 67 65 20 61 6e 64 20 61 73 6b 20 68 69 6d 20 74 6f 20 62 75 79 20 79 6f 75 20 61 20 62 65 65 72 21 21 21**

Question 14 - One of Po1s0n1vy’s staged domains has some disjointed “unique” whois information. Concatenate the two codes together and submit them as a single answer.

- I tried to look through the whois information of the domains discovered with ThreatCrowd in Question 8, but was not able to find the answer. I searche the whois history using `https://www.whoxy.com/`.
- I discovered unusual `company_name` and `mailing_address` field values for `waynecorinc.com`.

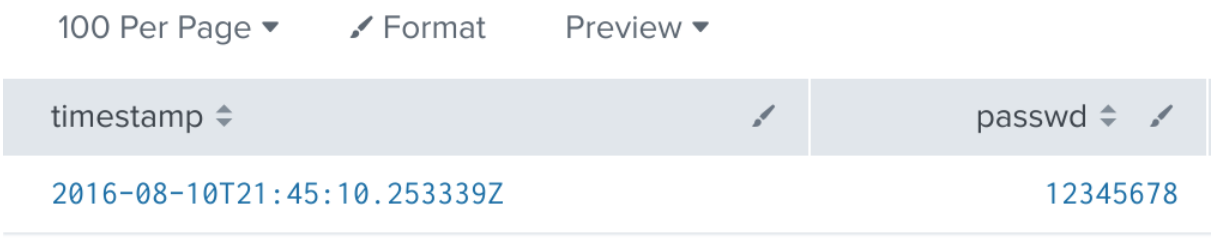


- **Answer: 31 73 74 32 66 69 6E 64 67 65 74 73 66 72 65 65 62 65 65 72 66 72 6F 6D 72 79 61 6E 66 69 6E 64 68 69 6D 74 6F 67 65 74**

Question 15 - What was the first brute force password used?

- Building off of Question 9, I extracted the `passwd` field from `form_data` field using regular expressions. The first brute force password used was `12345678`.

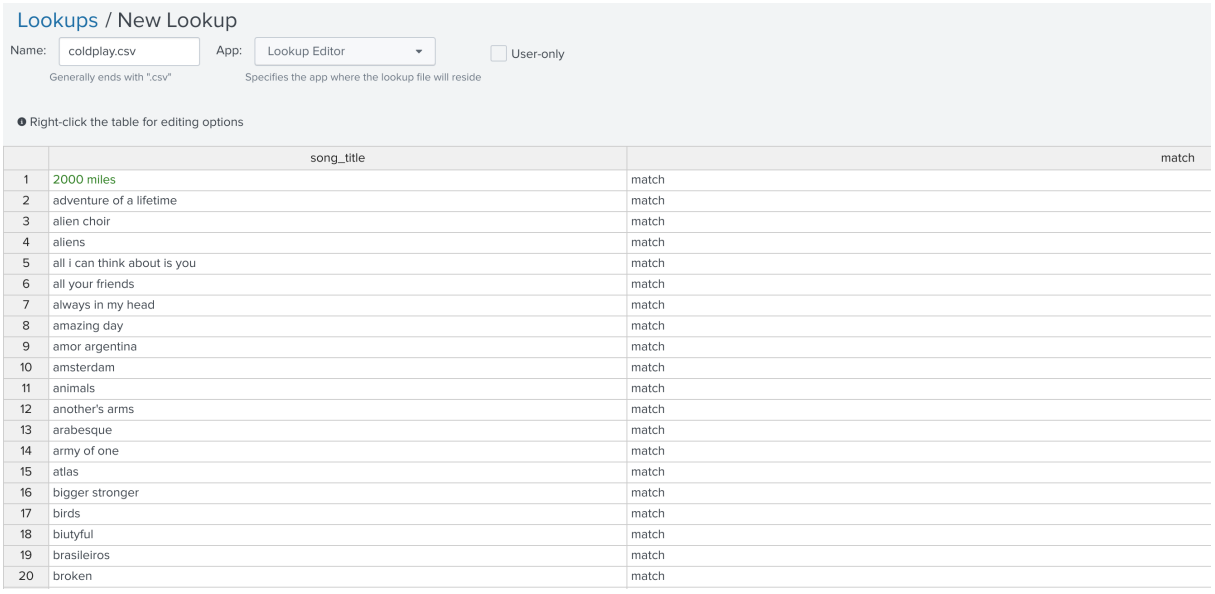
```
index="botsv1" sourcetype="stream:http" http_method=POST uri="*administrator/index.php*"
http_user_agent="Python-urllib/2.7"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| table timestamp passwd
| sort - timestamp desc
| head 1
```



- **Answer: 12345678**

Question 16 - One of the passwords in the brute force attack is James Brodsky's favorite Coldplay song. Hint: we are looking for a six-character word on this one. Which is it?

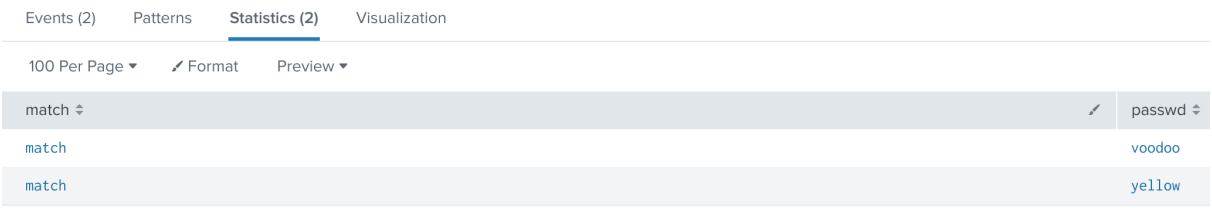
- I grabbed the list of songs in csv format and added the title `song_title` and `match`. I imported the CSV file into Splunk lookup editor.



	song_title	match
1	2000 miles	match
2	adventure of a lifetime	match
3	alien choir	match
4	aliens	match
5	all i can think about is you	match
6	all your friends	match
7	always in my head	match
8	amazing day	match
9	amor argentina	match
10	amsterdam	match
11	animals	match
12	another's arms	match
13	arabesque	match
14	army of one	match
15	atlas	match
16	bigger stronger	match
17	birds	match
18	biutyful	match
19	brasileiros	match
20	broken	match

- Building on top of the query from Question 15, I filtered for the 6 character passwords and did a lookup against the CSV file. There were two matches - `voodoo` and `yellow`. The question took `yellow` as the answer.
- Search query:

```
index="botsv1" sourcetype="stream:http" http_method=POST uri="*administrator/index.php*"
http_user_agent="Python-urllib/2.7"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| eval passwdlen=len(passwd)
| search passwdlen=6
| lookup coldplay.csv song_title as passwd output match
| search match=*
| table match passwd
```

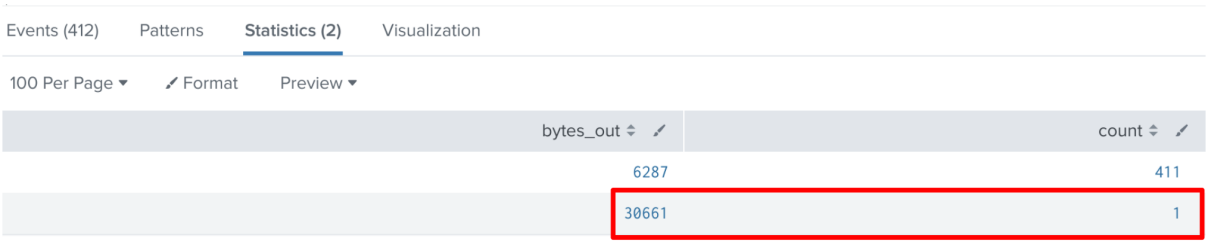


- **Answer: yellow**

Question 17 - What was the correct password for admin access to the content management system running "imreallynotbatman.com"?

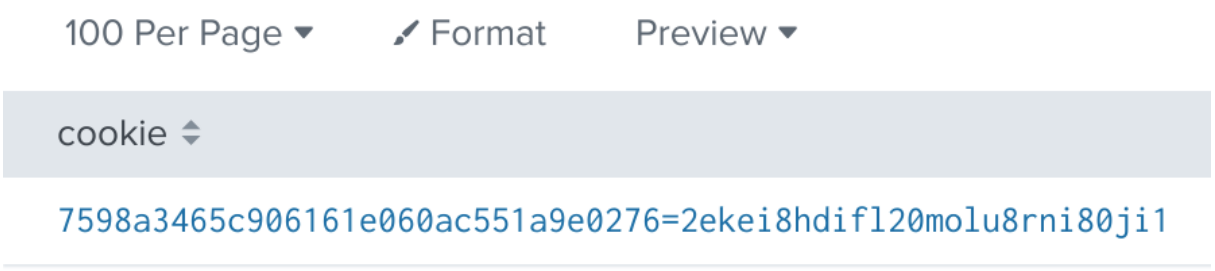
- The `POST` requests for brute forcing passwords were redirected, and `GET` requests were used to display the redirected pages. When I searched for the `GET` requests to `/joomla/administrator/index.php`, there were 824 results, which is double the amount of the `POST` requests. This is due to `FETCH`, which sent `GET` request twice.
- I filtered out the initial `GET` requests that do not contain `cookie` field. This leaves us with requests with session information.
- From the output, I observed `bytes_in` and `bytes_out` fields. All requests had `bytes_in` value of 223, but `bytes_out` had two distinct values - 30661 bytes for one request and 6287 bytes for the rest. This means that out of 412 brute force attempts, 1 returned a unique HTML page.
- Search query:

```
index="botsv1" sourcetype="stream:http" c_ip="23.22.63.114" http_method="GET"
uri="/joomla/administrator/index.php" cookie="*"
| stats count by bytes_out
| sort by count desc
```



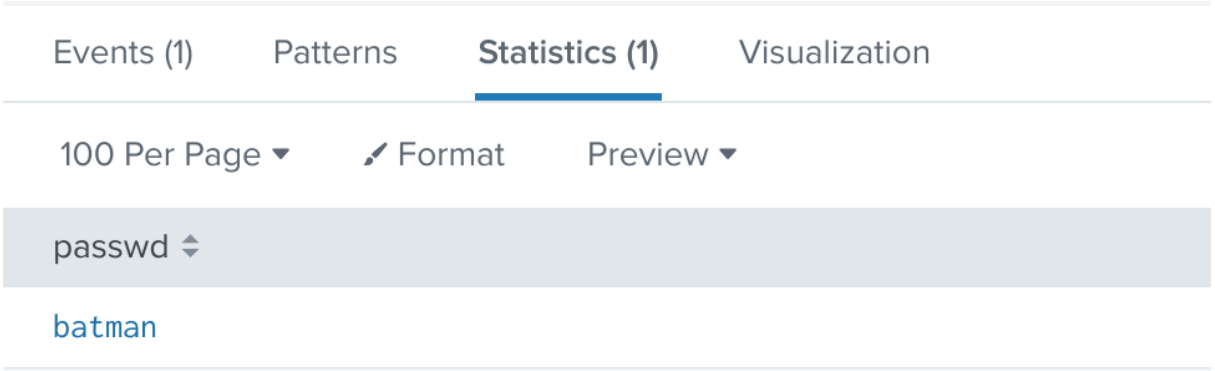
- As opposed to other requests, the `dest_content` of this request did not contain the message “Username and password do not match or you do not have an account yet.”, which suggested successful logon.
- I searched for the corresponding session cookie.
- Search query:

```
index="botsv1" sourcetype="stream:http" c_ip="23.22.63.114" http_method="GET"
uri="/joomla/administrator/index.php" cookie="*"
| search bytes_out=30661
| table cookie
```



- I then searched for matching sessions, and one of the events was a brute force `POST` request using the password “batman”.
- Search query:

```
index="botsv1" sourcetype="stream:http" c_ip="23.22.63.114"
uri="/joomla/administrator/index.php"
cookie="7598a3465c906161e060ac551a9e0276=2ekei8hdifl20molu8rni80ji1"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| search passwd="*"
| table passwd
```

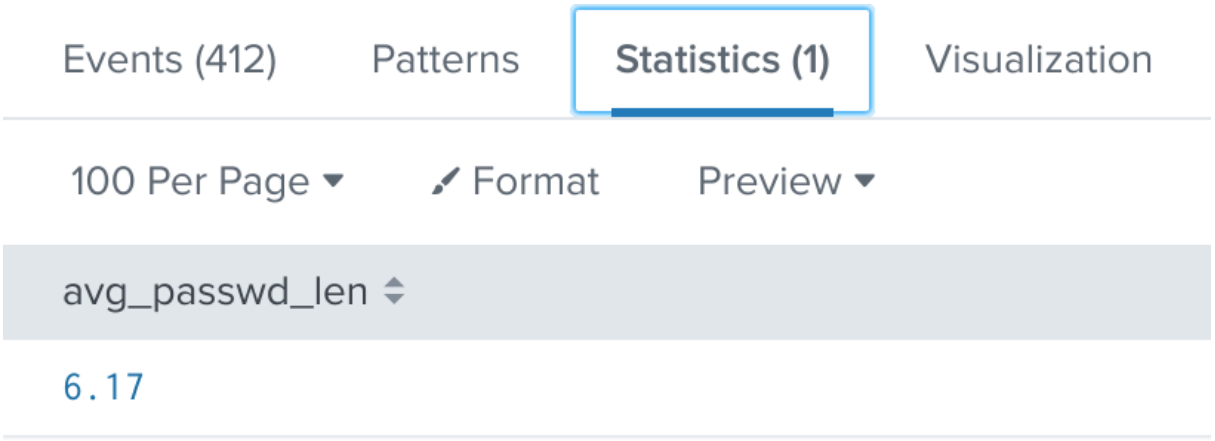


- **Answer: batman**

Question 18 - What was the average password length used in the password brute-forcing attempt? (Round to a closest whole integer. For example “5” not “5.23213”)

- Search query:

```
index="botsv1" sourcetype="stream:http" c_ip="23.22.63.114" http_method="POST"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| eval passwd_len = len(passwd)
| stats avg(passwd_len) as avg_passwd_len
| eval avg_passwd_len = round(avg_passwd_len,2)
```



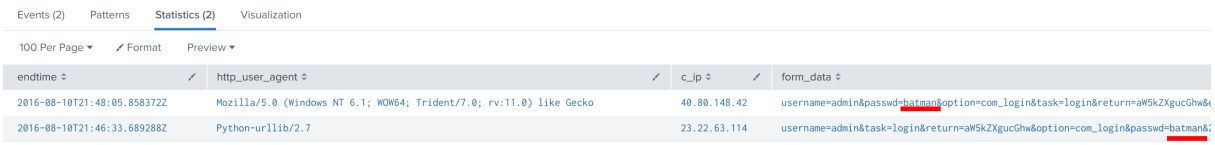
- **Answer: 6**

Question 19 - How many seconds elapsed between the brute force password scan identified the correct password and the compromised login? Round to 2 decimal places.

- For this question, I filtered for any requests containing the password “batman” in the `form_data` field.
- After brute forcing the admin password, the threat actor logged in for the second time with a browser from a different IP address.

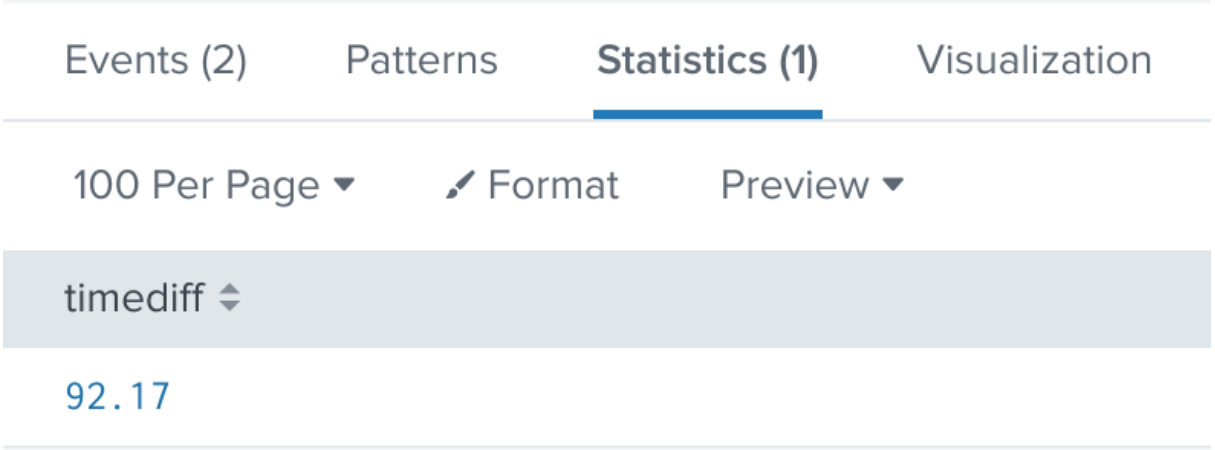
- Search query:

```
index="botsv1" sourcetype="stream:http"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| search passwd="batman"
| table endtime http_user_agent c_ip form_data
```



- I then stripped the timestamps into epoch and subtracted the two values. The time difference between the brute force and logon was 92.17 seconds.
- Search query:

```
index="botsv1" sourcetype="stream:http"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| search passwd="batman"
| eval striptime = strptime(endtime, "%Y-%m-%dT%H:%M:%S.%6QZ")
| stats max(striptime) as compromised_login min(striptime) as bruteforce_login
| eval timediff = round(compromised_login - bruteforce_login,2)
| fields timediff
```

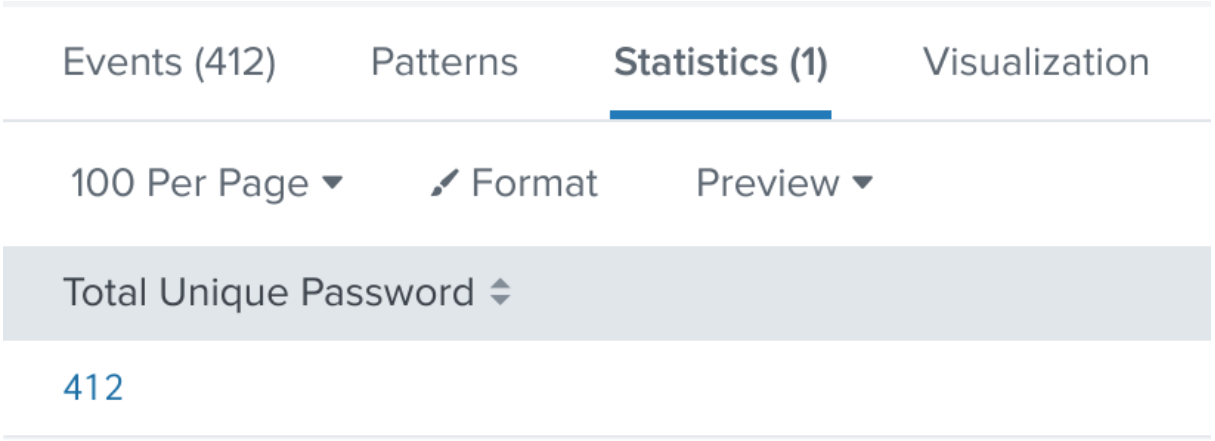


- **Answer: 92.17**

Question 20 - How many unique passwords were attempted in the brute force attempt?

- Search query:

```
index="botsv1" sourcetype="stream:http" c_ip="23.22.63.114" http_method="POST"
| rex field="form_data" "passwd=(?<passwd>.+?(?=&|$))"
| stats dc(passwd) as "Total Unique Password"
```

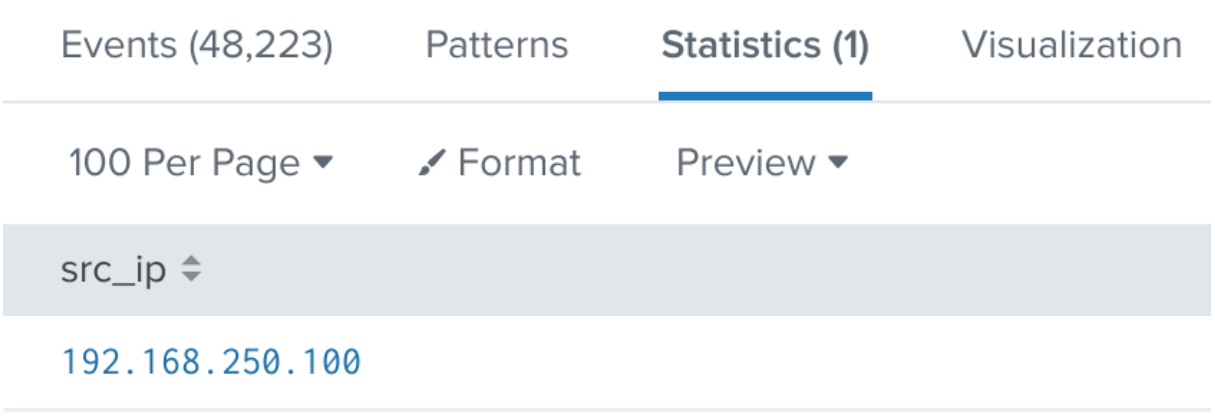


- **Answer: 412**

Question 21 - What was the most likely IP address of we8105desk in 24AUG2016?

- I searched for the event log after 8/24/16 with the user `bob.smith` and extracted the `src_ip` field. I excluded any loopback IP.
- Search query:

```
index="botsv1" host=we8105desk source="WinEventLog:Microsoft-Windows-Sysmon/Operational"
User="WAYNECORPINC\\bob.smith"
| search src_ip!=127*
| stats count by src_ip
```

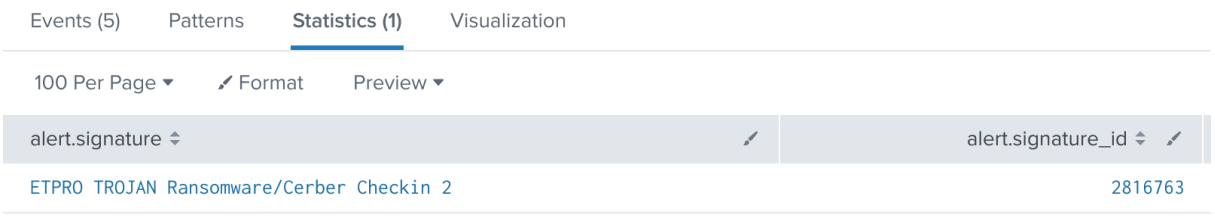


- **Answer: 192[.]168.250.100**

Question 22 - Amongst the Suricata signatures that detected the Cerber malware, which one alerted the fewest number of times? Submit ONLY the signature ID value as the answer. (No punctuation, just 7 integers.)

- I searched the suricata log for alert signature that contains the word “cerber”. I extracted the relevant alert with minimum count and queried for its signature ID.
- Search query:

```
index="botsv1" sourcetype="suricata"
| search alert.signature="*Cerber*"
| stats count by alert.signature alert.signature_id
| head 1
```

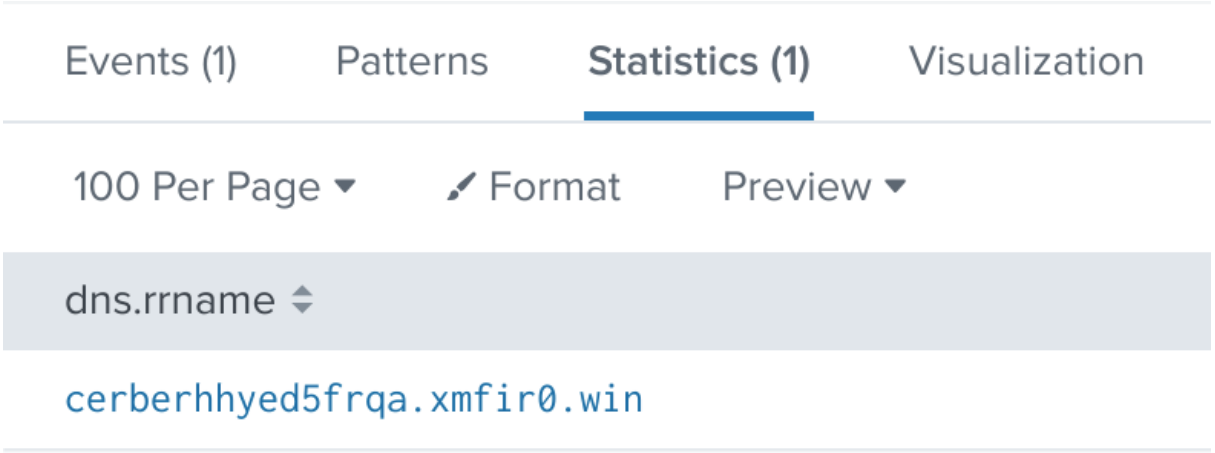


- **Answer: 2816763**

Question 23 - What fully qualified domain name (FQDN) makes the Cerber ransomware attempt to direct the user to at the end of its encryption phase?

- One of the alert signatures was “ETPRO TROJAN Ransomware/Cerber Onion Domain Lookup”. I searched the suricata log for the corresponding `flow_id` field.
- I extracted DNS A record from the traffic flow and discovered `cerberhhyed5frqa.xmfir0.win`.
- Search query:

```
index="botsv1" sourcetype="suricata"
| search flow_id=4097325782 dns.rrtype=A
| stats count by dns.rrname
```

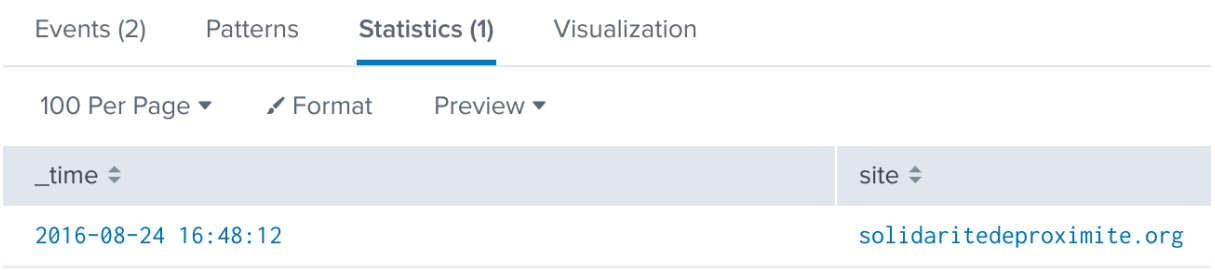


- **Answer: cerberhhyed5frqa[.]xmfir0.win**

Question 24 - What was the first suspicious domain visited by we8105desk in 24AUG2016?

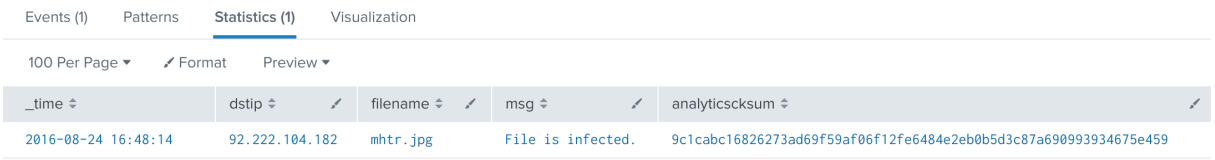
- First, I adjusted the timeline for Aug 24 2016 events. I then searched the Fortigate UTM log for traffic originating from `192.168.250.100`. I observed the application category and discovered that the host was participating in the botnet. I excluded the botnet traffic and NetBIOS traffic. I further exluded traffic to microsoft.com and bing.com. I was left with the traffics to `solidaritedeproximite.org` and `92.222.104.182`. Both requesting for uri `/mhtr.jpg`.
- Search query:

```
index="botsv1" sourcetype="fgt_utm" src="192.168.250.100" appcat!=Botnet
app!="NetBIOS.Name.Service"
| search site!="*microsoft*" site!="*bing*"
| table _time site
| sort - _time desc
| head 1
```

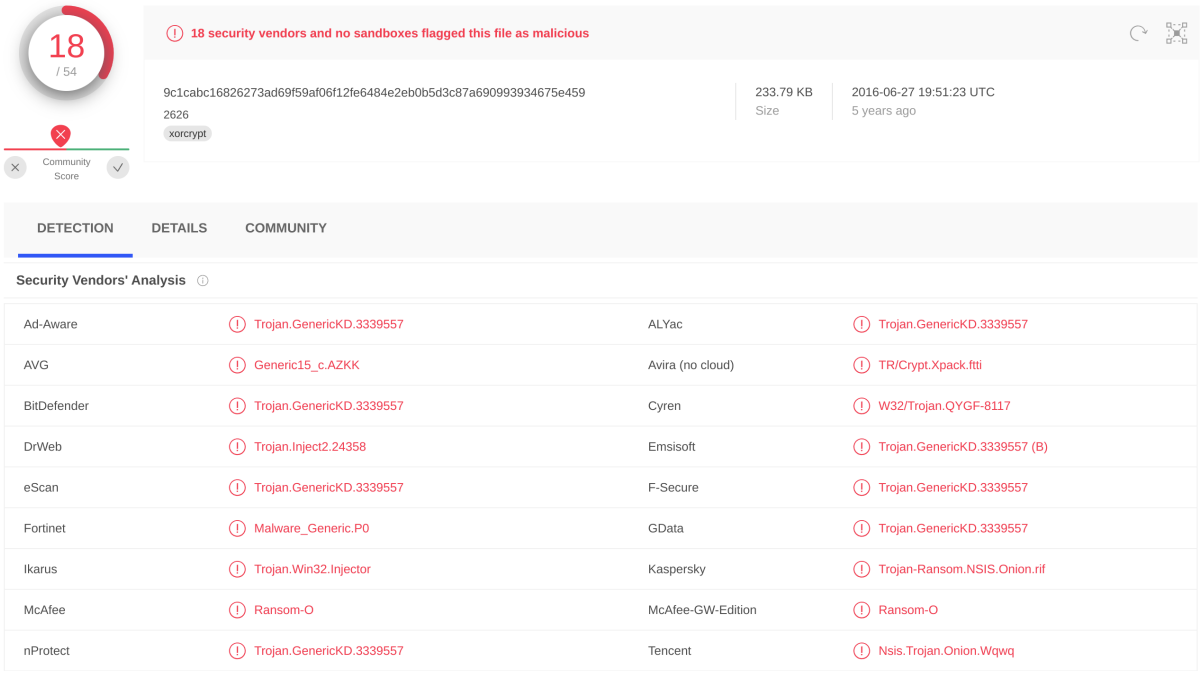


- I checked if Suricata or Fortinet UTM picked up on the file as being malicious. I found one event where Fortinet UTM labelled the file as infected.
- Search query:

```
index="botsv1" sourcetype=fgt_utm
| search filename=mhtr.jpg
| table _time dstip filename msg analyticscksum
```



- I checked VirusTotal for the hash, and it was determined that the file is the encryptor for the Cerber ransomware.

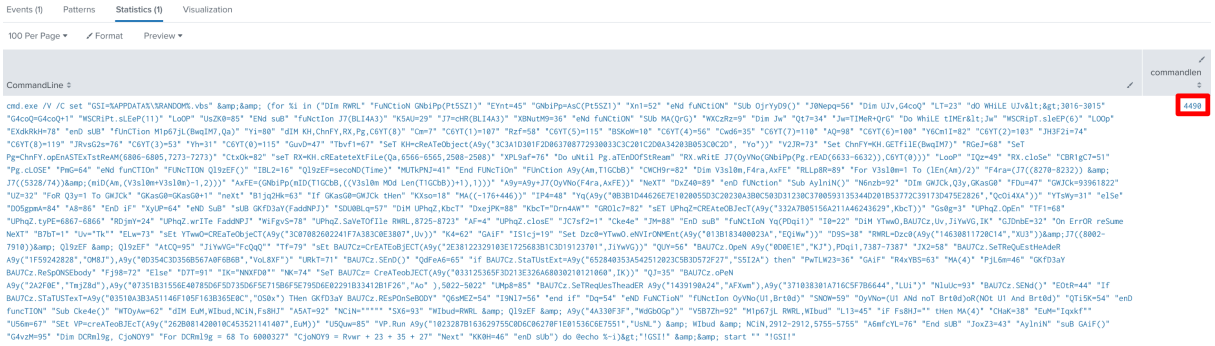


- **Answer: solidaritedeproximite[.]org**

Question 25 - During the initial Cerber infection a VB script is run. The entire script from this execution, pre-pended by the name of the launching .exe, can be found in a field in Splunk. What is the length in characters of the value of this field?

- I seach the Sysmon log for process creation and any field that contains .vbs extension.
- I then observed the parent processes and noticed `winword.exe`. I was confident that the script was run from the user opening a malicious document.
- The query returned a long command that contained a vbscript. I calculated the length of the field.
- Search query:

```
index="botsv1" source="wineventlog:microsoft-windows-sysmon/operational" EventCode=1 vbs
ParentImage="C:\\Program Files (x86)\\Microsoft Office\\Office14\\WINWORD.EXE"
| eval commandlen = len(CommandLine)
| table CommandLine commandlen
```

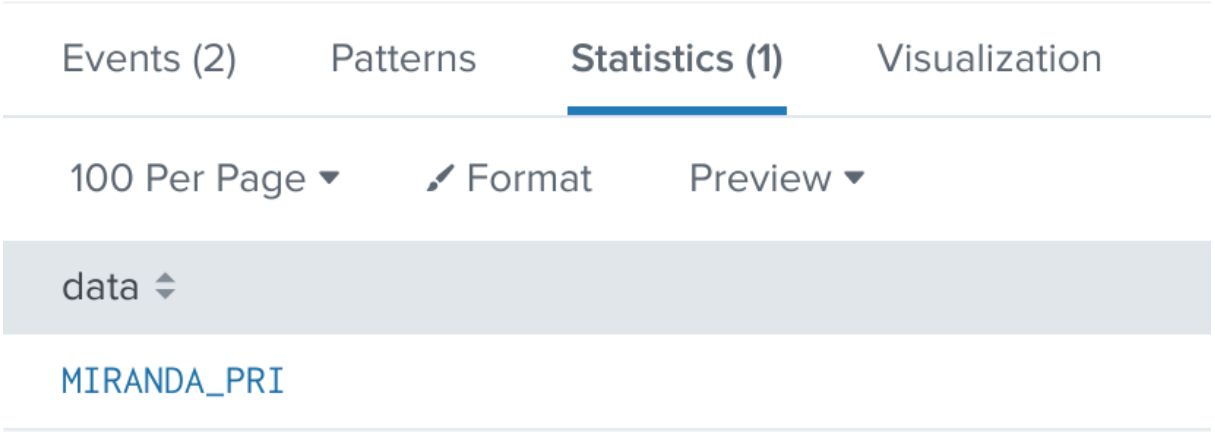


- **Answer: 4490**

Question 26 - What is the name of the USB key inserted by Bob Smith?

- I searched the Splunk WinRegistry for any record of USB devices inserted. I assumed that the user plugged in the key on his desktop `we8105desk`. USB history is usually stored in the registry under `USBSTOR` key. I queried the `friendlyname` object and found the USB name `MIRANDA_PRI`.
- Search query:

```
index="botsv1" source=WinRegistry dest="we8105desk" key_path="*usbstor*friendlyname*"
| stats count by data
```

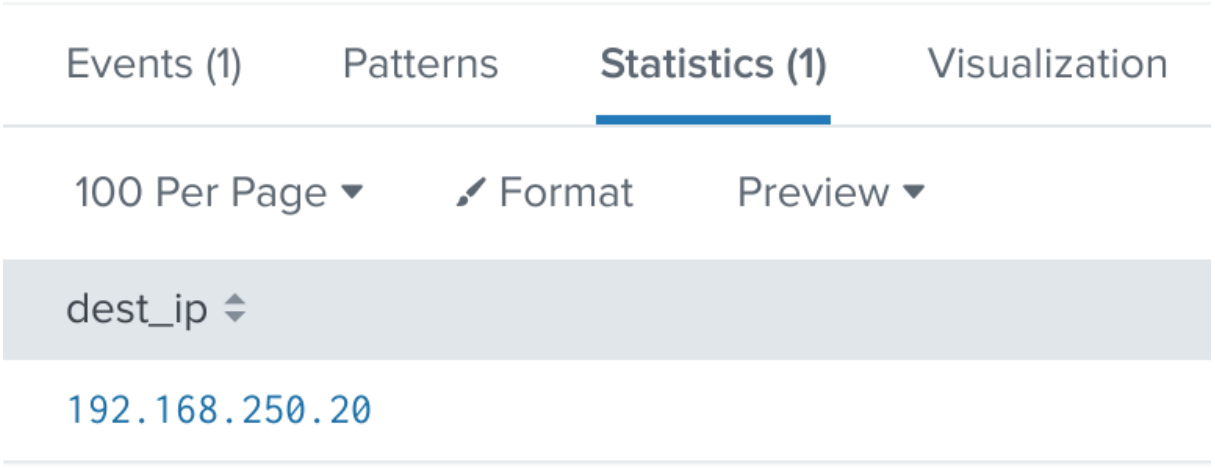


- **Answer: MIRANDA_PRI**

Question 27 - Bob Smith's workstation (we8105desk) was connected to a file server during the ransomware outbreak. What is the IP address of the file server?

- I searched the SMB stream log to find the file server IP address. I queried a logon from `bob.smith` and extracted the destination IP, which was `192.168.250.20`.
- Search query:

```
index="botsv1" sourcetype="stream:smb" login="bob.smith"
| table dest_ip
```

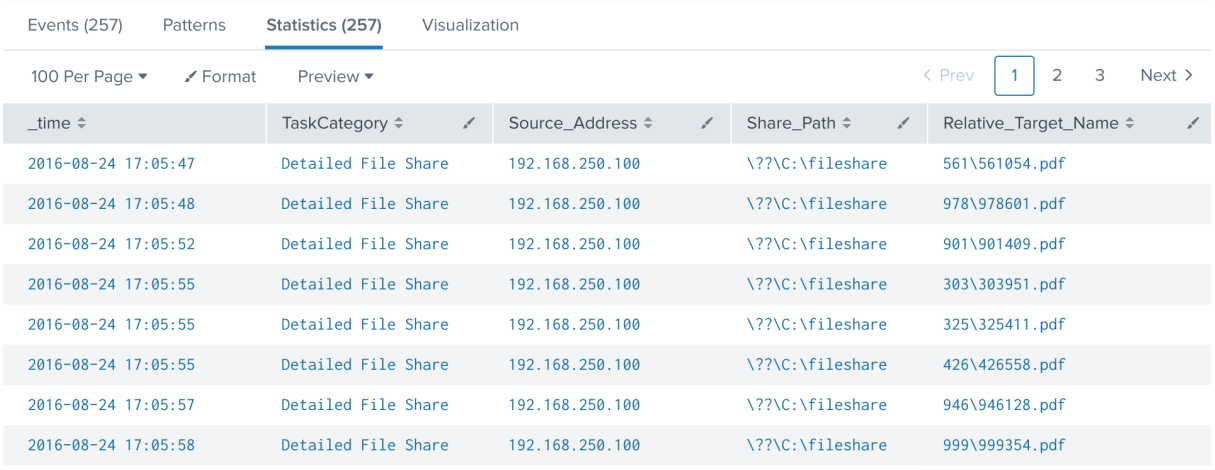


- **Answer: 192[.]168.250.20**

Question 28 - How many distinct PDFs did the ransomware encrypt on the remote file server?

- I searched the Windows Event Log to analyze the audit file system events. I set the host as the file server and searched for the `Event ID 5145`. This event is created when a network share object (file or folder) is accessed. I narrowed the search for write access to PDF.
- The query returned file access 257 events. I noticed that the time elapsed between the first access and the last was less than 10 minutes. Excessive file access within short period of time can be indicative of a ransomware attack. In this case, the ransomware encrypted 257 PDF files within 10 minutes.
- Search query:

```
index="botsv1" source="WinEventLog:*" host=we9041srv EventCode=5145 Relative_Target_Name="*.pdf"
WriteData
| table _time TaskCategory Source_Address Share_Path Relative_Target_Name
| sort by _time
```



Events (257) Patterns Statistics (257) Visualization

100 Per Page ▾ Format Preview ▾ < Prev 1 2 3 Next >

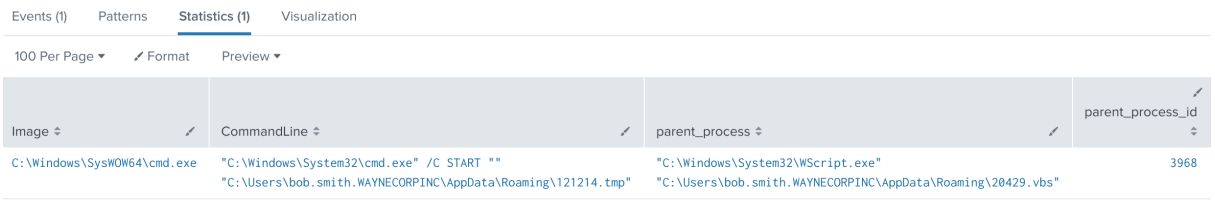
_time	TaskCategory	Source_Address	Share_Path	Relative_Target_Name
2016-08-24 17:05:47	Detailed File Share	192.168.250.100	\??\C:\fileshare	561\561054.pdf
2016-08-24 17:05:48	Detailed File Share	192.168.250.100	\??\C:\fileshare	978\978601.pdf
2016-08-24 17:05:52	Detailed File Share	192.168.250.100	\??\C:\fileshare	901\901409.pdf
2016-08-24 17:05:55	Detailed File Share	192.168.250.100	\??\C:\fileshare	303\303951.pdf
2016-08-24 17:05:55	Detailed File Share	192.168.250.100	\??\C:\fileshare	325\325411.pdf
2016-08-24 17:05:55	Detailed File Share	192.168.250.100	\??\C:\fileshare	426\426558.pdf
2016-08-24 17:05:57	Detailed File Share	192.168.250.100	\??\C:\fileshare	946\946128.pdf
2016-08-24 17:05:58	Detailed File Share	192.168.250.100	\??\C:\fileshare	999\999354.pdf

- **Answer: 257**

Question 29 - The VBScript found in question 25 launches 121214.tmp. What is the ParentProcessId of this initial launch?

- I searched the Windows Event Log for the process creation of the file `121214.tmp` with a VBScript parent process.
- The parent process ID was 3968.
- Search query:

```
index="botsv1" source="wineventlog:microsoft-windows-sysmon/operational" EventCode=1
Image"*121214.tmp" parent_process="*vbs*"
| table Image CommandLine parent_process parent_process_id
```



Events (1) Patterns Statistics (1) Visualization

100 Per Page ▾ Format Preview ▾

Image	CommandLine	parent_process	parent_process_id
C:\Windows\SysWOW64\cmd.exe	"C:\Windows\System32\cmd.exe" /C START "" "C:\Users\bob.smith.WAYNECORPINC\AppData\Roaming\121214.tmp"	"C:\Windows\System32\WScript.exe" "C:\Users\bob.smith.WAYNECORPINC\AppData\Roaming\20429.vbs"	3968

- **Answer: 3968**

Question 30 - The Cerber ransomware encrypts files located in Bob Smith's Windows profile. How many .txt files does it encrypt?

- Following the tutorial from Splunk, I searched the Sysmon log to find new file creation events within short period of time. I searched for any process that created more than 10 text files within 1 second. I used a short time window, because it takes less time for a ransomware to encrypt text files which tend to be smaller in size.
- The query returned only one process `osk.exe`.
- Search query:

```
index="botsv1" sourcetype="xmlwineventlog:microsoft-windows-sysmon/operational"
EventDescription="File Create Time" TargetFilename="C:\\Users\\bob.smith.WAYNECORPINC\\*.txt"
| streamstats time_window=1s count(EventDescription) AS "new_files"
| search new_files>10
| stats count by Image process_id
```

Events (339) Patterns **Statistics (1)** Visualization

100 Per Page ▾ Format Preview ▾

Image	process_id	count
C:\Users\bob.smith.WAYNECORPINC\AppData\Roaming\{35ACA89F-933F-6A5D-2776-A3589FB99832}\osk.exe	3588	339

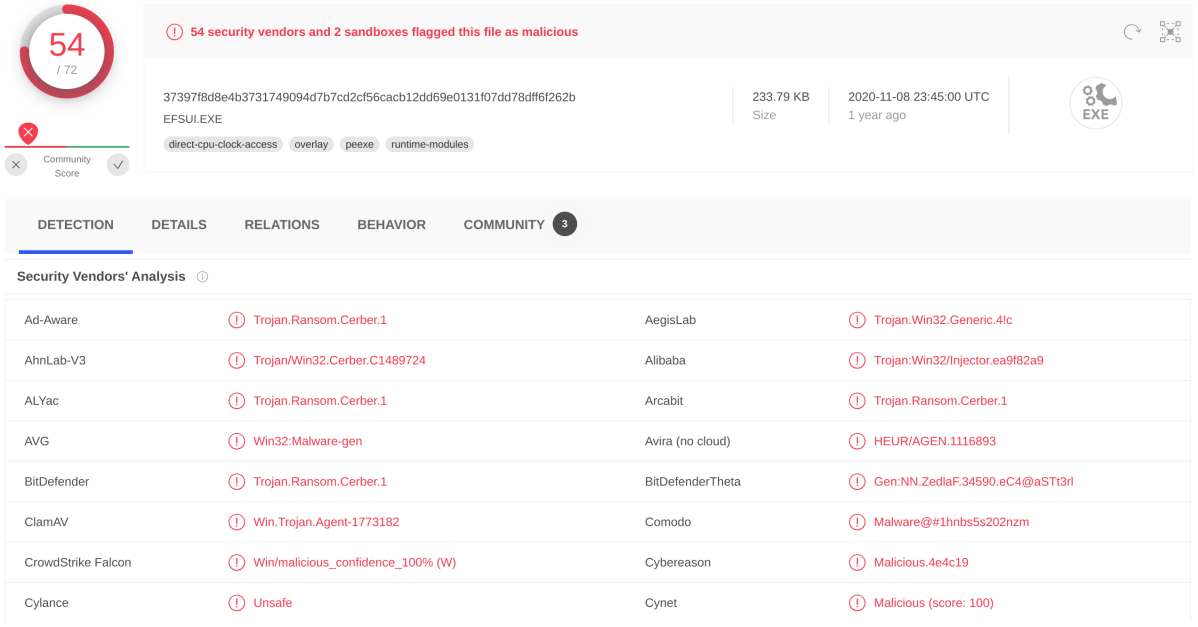
- I search the MD5 hash of the process using its process ID and searched VirusTotal. The process turned out to be Cerber ransomware.
- Search query:

```
index="botsv1" sourcetype="xmlwineventlog:microsoft-windows-sysmon/operational" EventCode=1
process_id=3588
| table MD5
```

Events (1) Patterns **Statistics (1)** Visualization

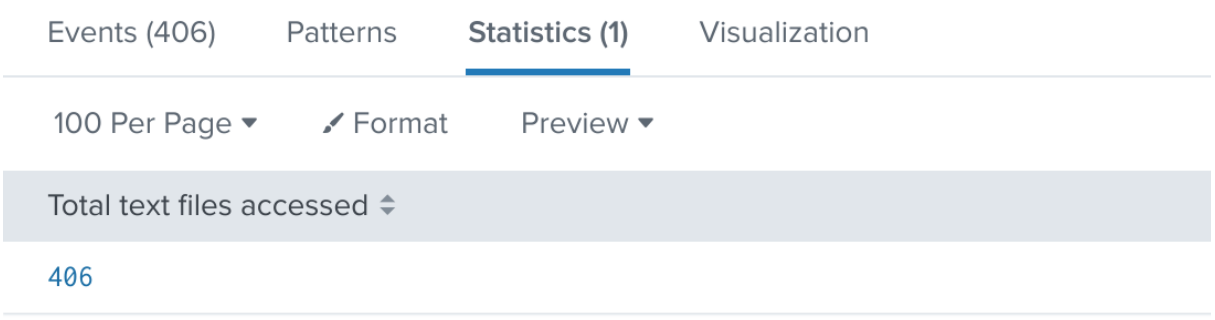
100 Per Page ▾ Format Preview ▾

MD5
EE0828A4E4C195D97313BFC7D4B531F1



- Based on the findings, I searched for the number of text files `osk.exe` accessed in Bob Smith's Windows profile.
- Cerber ransomware encrypted 406 text files.
- Search query:

```
index="botsv1" sourcetype="xmlwineventlog:microsoft-windows-sysmon/operational" process_id=3588
EventDescription="File Create Time" TargetFilename="C:\\Users\\bob.smith.WAYNECORPINC\\*.txt"
| stats count by TargetFilename
| eventstats sum(count) as "Total text files accessed"
| fields "Total text files accessed"
| head 1
```



- **Answer: 406**

Question 31 - The malware downloads a file that contains the Cerber ransomware crypto code. What is the name of that file?

- As discovered in Question 24, the encryptor file was `mhtr.jpg`.
- **Answer: mhtr.jpg**

Question 32 - Now that you know the name of the ransomware's encryptor file, what obfuscation technique does it likely use?

- The malicious executable was embedded within a jpg file. This is a steganography technique.
- **Answer: steganography**