COMP90073 Security Analytics, Semester 2 2020

Project 1: Detecting cyberattacks in network traffic data

Name: Diego Aranda Villarreal

Student ID: 992038

September 8th, 2020

Table of Contents

1 Introduction

The use of internet technologies by individuals, private companies and public service organizations has become under constant threat of potential intruders and anomalies while performing their operations. These attackers might use different techniques to penetrate networks and systems, which have been evolving over time and new patterns of attack have emerged. The characteristics and nature of such attacks can be related to variations in the volume of network traffic and packets, the use and focus of different ports, protocols, among other features. The purpose of this report is to use Splunk to analyse network traffic from a given dataset and to assess the different consequences and countermeasures to be potentially applied in the real world.

2 Overview of the dataset

2.1 Ingesting the dataset into PCAP Analyzer

Firstly, both Splunk software and PCAP Analyzer application were installed. Then, the .pcap file provided for this analysis (55.6 MB) was ingested in the software by creating the file location in the Windows OS and placing the file in it. Once the script of the software run automatically, the file was appropriately ingested in Splunk as .csv (84 MB).

2.2 Dataset

The dataset represents the network traffic comprised of 323,154 events, where the start event date was 2020-06-19 00:55:19.383 and the last event date was 2020-06-19 01:02:09.742, which represents around 7 [min] of network traffic data.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" | stats
    earliest(_time) as starttime, latest(_time) as endtime | eval start = strftime(starttime, "%Y-%m-%d %H:%M
    :%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M:%S.%Q") | table start, finish
```

Figure 1 – Searching for Start and Finish time of the events

2.3 High level view of dataset

We performed an exploratory analysis on the data by using basic features of Splunk and graphical views. The dashboard function of PCAP Analyzer was used to provide a summary of the data.

Figure 2 showed that the most used protocol was TCP with 229,430 transactions, which accounted for around 71% of all events. The second place was for DNS with 80,747 events (25%). The third place was for HTTP with 2,550 events (1%).

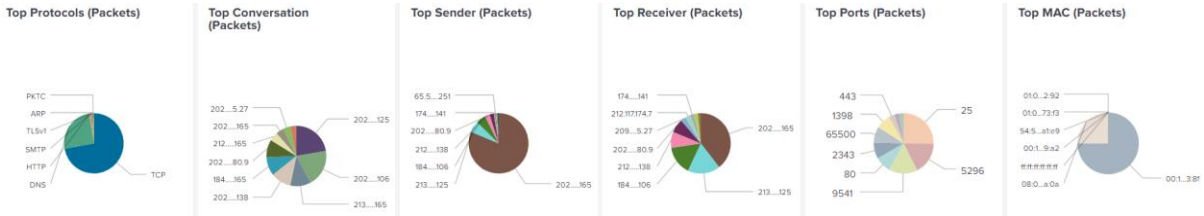


Figure 2 – Top protocols involved [packets]

Figure 3 showed that there might be a pattern of attack for the period between 00:55:30 AM – 00:56:30 AM and the period between 1:00:30 AM – 1:01:30 AM. The protocols with the most traffic for those periods correspond to HTTP, IRC, SMTP and TCP protocols, for which peaks of 2 [GB] of traffic were found.

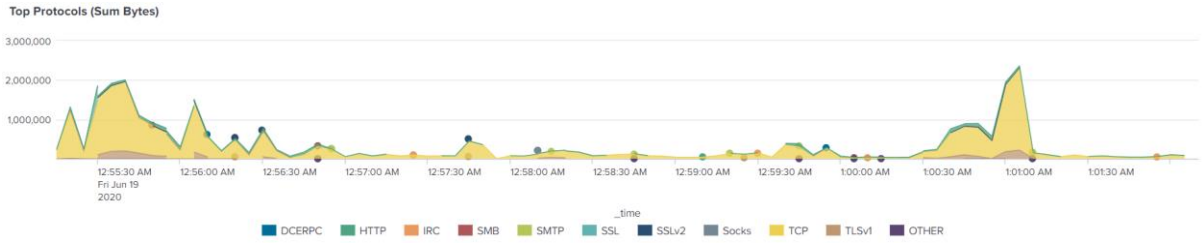


Figure 3 – Top protocols involved [bytes]

Moreover, Figure 4 that the host 202.166.84.165 was involved in several conversations with other addresses in the same periods of time as above.

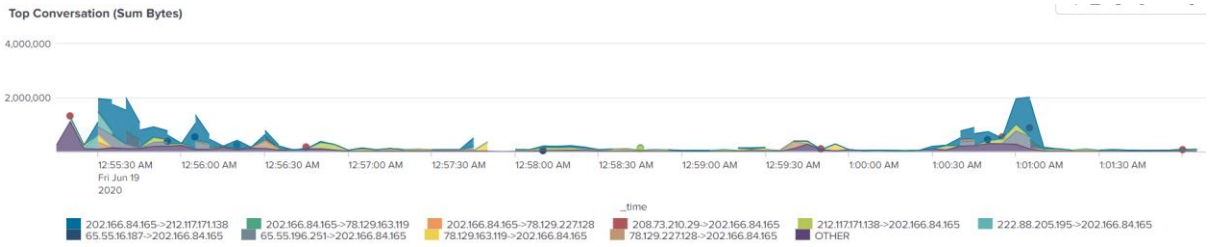


Figure 4 – Top Conversations between IPs [bytes]

It can be observed in Figure 5 and Figure 6 that 202.166.84.165 appeared as both top sender and top receiver, following the same pattern as stated above for the same periods of time.

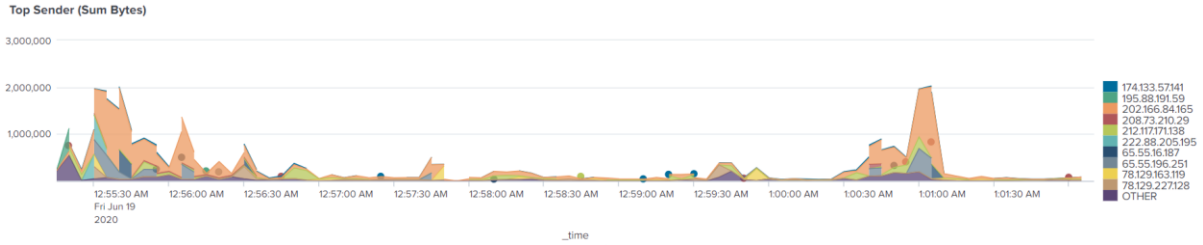


Figure 5 – Top Sender IPs [bytes]

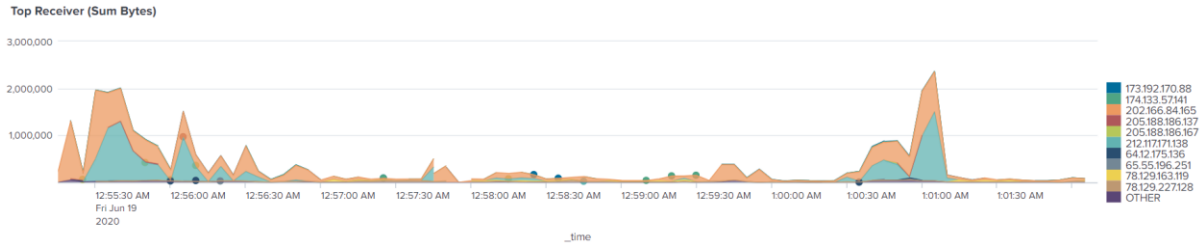


Figure 6 – Top Receiver IPs [bytes]

In addition, it was also seen that ports 587 and 80 were among the most used, which could be related to SMTP and HTTP protocols.

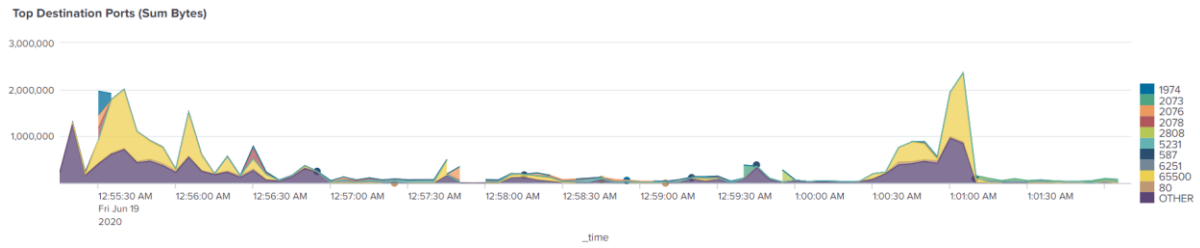


Figure 7 – Top Destination Ports [bytes]

3 Attacks

Table 1 provides a summary of the attacks related to our analysis. All events share the same date AEST Jun-19 2020. As expected, the machine with IP 202.165.84.165 was involved in al four attacks detected. The following sections will provide an evidence on how the attacks were detected.

Attack Summary	Timestamp				
Type	Start	Finish	Attacked services	Attacker(s)	Victim(s)
Botnet Command and Control	2020-06-19 00:55:21.400	2020-06-19 00:58:00.085	TCP services	31.192.109.167	202.166.84.165
SPAM	2020-06-19 00:55:23.278	2020-06-19 01:01:00.887	TCP services	202.166.84.165	205.188.186.137
	2020-06-19 00:55:28.415	2020-06-19 01:00:54.303	TCP services	202.166.84.165	205.188.186.167
	2020-06-19 00:55:24.632	2020-06-19 01:01:12.900	TCP services	202.166.84.165	64.12.168.40
	2020-06-19 00:55:26.189	2020-06-19 01:01:03.374	TCP services	202.166.84.165	64.12.175.136
	2020-06-19 01:01:03.772	2020-06-19 01:01:03.805	TCP services	202.166.84.165	98.136.185.95
ClickFraud	2020-06-19 00:55:22.906	2020-06-19 01:01:08.208	TCP services	202.166.84.165	98.126.71.122
IRC	2020-06-19 00:56:32.831	2020-06-19 00:56:32.831	TCP services	202.166.84.165	184.106.213.57
	2020-06-19 00:55:21.813	2020-06-19 00:55:21.813	TCP services	202.166.84.165	200.171.4.222
	2020-06-19 00:56:01.401	2020-06-19 00:59:50.226	TCP services	202.166.84.165	202.112.126.218
	2020-06-19 00:55:24.130	2020-06-19 00:55:24.130	TCP services	202.166.84.165	211.157.110.34
	2020-06-19 00:55:40.161	2020-06-19 00:55:40.161	TCP services	202.166.84.165	217.34.4.225
	2020-06-19 00:55:54.800	2020-06-19 00:55:54.800	TCP services	202.166.84.165	218.189.208.34
	2020-06-19 00:56:22.537	2020-06-19 00:59:56.192	TCP services	202.166.84.165	221.207.141.60
	2020-06-19 01:01:59.756	2020-06-19 01:01:59.756	TCP services	202.166.84.165	58.42.247.143
	2020-06-19 01:00:11.824	2020-06-19 01:00:11.824	TCP services	202.166.84.165	60.173.109.42
	2020-06-19 00:55:25.119	2020-06-19 01:00:47.803	TCP services	202.166.84.165	61.150.114.216
	2020-06-19 00:55:49.813	2020-06-19 00:56:43.757	TCP services	202.166.84.165	61.167.116.133
	2020-06-19 00:55:22.872	2020-06-19 00:56:37.880	TCP services	202.166.84.165	61.17.216.4
	2020-06-19 00:57:28.502	2020-06-19 01:00:41.386	TCP services	202.166.84.165	61.17.216.86
	2020-06-19 00:59:31.829	2020-06-19 00:59:31.829	TCP services	202.166.84.165	61.17.216.92
	2020-06-19 00:56:55.755	2020-06-19 00:56:55.755	TCP services	202.166.84.165	61.17.216.94
	2020-06-19 00:55:36.602	2020-06-19 00:55:36.602	TCP services	202.166.84.165	61.177.120.254
	2020-06-19 00:55:36.884	2020-06-19 00:55:36.884	TCP services	202.166.84.165	88.250.200.14

Table 1 – Summary of attacks

3.1 Botnet Command and Control

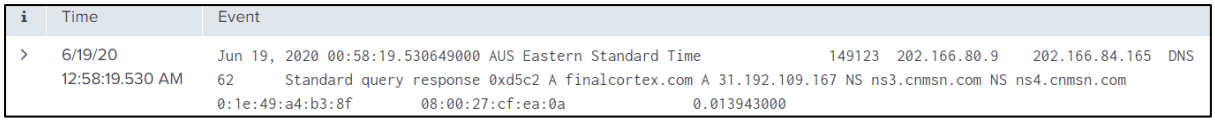
3.1.1 Evidence

There was a high traffic in HTTP protocol which suggested deeper research. We first looked out for the HTTP based requests to the C2 server “finalcortex.com” in order to get the destination IP address of such address:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" info = "*finalcortex.com*" AND info = "*response*"
```

Figure 8 – Searching for C2 destination IP address part 1

By inspecting the “info” field from Figure 9, the address trying to be reached by 202.166.84.165 corresponded to 31.192.109.167, the attacker.



i	Time	Event
>	6/19/20 12:58:19.530 AM	Jun 19, 2020 00:58:19.530649000 AUS Eastern Standard Time 149123 202.166.80.9 202.166.84.165 DNS 62 Standard query response 0xd5c2 A finalcortex.com A 31.192.109.167 NS ns3.cnmsn.com NS ns4.cnmsn.com 0:1e:49:a4:b3:8f 08:00:27:cf:ea:0a 0.013943000

Figure 9 – Searching for C2 destination IP address part 2

Later, the following statement was used in order to catch the events involved in the attack:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 31.192.109.167 AND protocol = "HTTP"
```

Figure 10 – Searching for events related to Command and Control attack

There were 54 events, for which we investigated deeper in order to assess which of the events were related to GET and POST methods. The GET method may be used for machines to test connectivity with destination IP addresses, and the POST method may be used for attempting to send data to a server to submit malicious data (Johari, 2020).

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 31.192.109.167 AND protocol = "HTTP" AND info = "*POST*" |
    stats earliest(_time) as starttime, latest(_time) as endtime by dst_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M:%S.%Q") | eval finish = strftime(endtime, "%Y
    -%m-%d %H:%M:%S.%Q") | eval victim = dst_ip | table victim, start, finish
```

Figure 11 – Searching for POST related events

The above sentence resulted in **53 events related to POST method**. The starting time of the attack may also be seen in Figure 12:

victim	start	finish
31.192.109.167	2020-06-19 00:55:21.400	2020-06-19 00:58:00.085

Figure 12 – Start and End time of Command and Control attack

3.1.2 Field extraction

We investigated the URI strings used by the attacker in the POST method. The URI string might be subject to vulnerabilities and the attacker might use it to gain access to the application layer by the means of malicious code (McFeters, 2008). The URI was extracted by using the function called Extract New Fields, for which the result generated by the regex functionality was "^(?:[^ \n]*){8}(?P<URI>[^]+)".

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 31.192.109.167 AND protocol = "HTTP" AND info = "*POST*" |
    dedup URI | table URI
```

Figure 13 – Searching for URI string of Command and Control attack

URI
/snapbn/gate.php

Figure 14 – URI string for Command and Control attack

A similar process was followed for obtaining the URI related to the GET method, which resulted to be "/snapbn/ip.php".

3.1.3 Summary

The following Table 2 shows a summary of our findings:

Attacker	Victim	Method	HTTP C2 requests	URI	Start	End
31.192.109.167	202.166.84.165	POST	53	/snapbn/gate.php	00:55:21.400	00:58:00.085
		GET	1	/snapbn/ip.php	00:55:21.316	-

Table 2 – Summary of Command and Control attack

Finally, we noticed that all the requests were directed to the port 80. A statement similar to Figure 14 was used to get the destination port.

3.1.4 Attack Narrative

At 00:55:21.279, **victim 202.166.84.165** made the first attempt to connect to "**finalcortex.com**" through DNS protocol directed to 202.166.80.9. It received a response from the DNS server at 00:55:21.309 and the IP address corresponded to **31.192.109.167**, the **attacker**. Once connection was established, the bot used the **HTTP protocol** to send a message to 31.192.109.167 containing the GET method in order to test connectivity at 00:55:21.316. Once connectivity was positively tested, the infected system attempted to send **53 POST HTTP requests** to the C2 server, all containing the **URI = /snapbn/gate.php**, the **first event** happened at **00:55:21.400**, and all received an OK response from the attacker. The **last attack** was at **00:58:00.085**. All transactions occurred at **port 80**.

3.2 SPAM

3.2.1 Evidence

There was a high traffic related to the SMTP protocol that suggested more investigation. We considered to search for the SMTP protocol alongside with the RCPT string.

The following statement was used to investigate the events related to SPAM:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "SMTP"  AND info = "*RCPT*"
```

Figure 15 – Investigating SPAM attack

The result contained 214 events, each one containing an email address in the "info" field. Next, we run the following statement to get the IP address of the attacker, alongside with the start and end time of the attack.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "SMTP"  AND info = "*RCPT*" | stats earliest(_time) as
    starttime, latest(_time) as endtime by src_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M:%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M:%S.%Q") |
    eval ip = src_ip | table ip, start, finish
```

Figure 16 – Searching for IP address of attacker

ip	start	finish
202.166.84.165	2020-06-19 00:55:23.278	2020-06-19 01:01:12.900

Figure 17 – Start and end time of SPAM attack

3.2.2 Field extraction

We performed an investigation on the email recipients of the SPAM attack. During a SPAM campaign the targeted emails can be vulnerable to be infected. If so, malicious emails can be sent on their behalf (Zhou, 2020). We utilized the PCAP Analyzer Extract New Fields function, for which the result generated by the regex functionality was "^(?:[^ \n]*){10}(?P<rcptemail>[^\t]+)". Figure 18 provides the statement used for obtaining the email addresses, and Appendix 1 contains the whole **214 list of targeted emails**.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "SMTP"  AND info = "*RCPT*" | dedup rcptemail | table
    rcptemail
```

Figure – 18 Searching for email addresses of SPAM attack

3.2.3 Summary

We run the following statements to get the first and last email recipients including the timestamps, and the IP addresses of the servers targeted.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "SMTP"  AND info = "*RCPT*" | stats earliest(_time) as
    starttime, latest(_time) as endtime, earliest(rcptemail) as startemail, latest(rcptemail) as finishemail, by src_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M
    :%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M:%S.%Q") | eval ip = src_ip | table ip, start, startemail, finish, finishemail
```

Figure 19 – Searching for first and last recipient of SPAM attack

ip	start	startemail	finish	finishemail
202.166.84.165	2020-06-19 00:55:23.278	<nickandsonia@comcast.net>	2020-06-19 01:01:12.900	<jberman1@gmail.com>

Figure 20 – First and last email recipients of SPAM attack

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "SMTP"  AND info = "*RCPT*" | stats earliest(_time) as
    starttime, latest(_time) as endtime by dst_ip, src_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M:%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M:%S
    .%Q") | eval victim = dst_ip | eval attacker = src_ip | table attacker, victim, start, finish
```

Figure 21 – Searching for IP addresses targeted by SPAM attack

attacker	victim	start	finish
202.166.84.165	205.188.186.137	2020-06-19 00:55:23.278	2020-06-19 01:01:00.887
202.166.84.165	205.188.186.167	2020-06-19 00:55:28.415	2020-06-19 01:00:54.303
202.166.84.165	64.12.168.40	2020-06-19 00:55:24.632	2020-06-19 01:01:12.900
202.166.84.165	64.12.175.136	2020-06-19 00:55:26.189	2020-06-19 01:01:03.374
202.166.84.165	98.136.185.95	2020-06-19 01:01:03.772	2020-06-19 01:01:03.805

Figure 22 – IP addresses targeted by SPAM attack

Finally, we noticed that all the requests were directed to the port 587. A statement similar to Figure 18 was used in order to get the destination port.

3.2.4 Attack Narrative

At 00:55:23.224, **attacker 202.166.84.165** made the first attempt to connect to “smtp.aol.com” through DNS protocol to 202.166.80.9. It received a response from the DNS server at 00:55:23.227 and the IP address corresponded to 205.188.186.137, address that contained the first set of victims. Then, the bot attempted to authenticate in the email system which achieved by 00:55:23.268. Next, at **00:55:23.278** sent an email to the **first victim** <**nickandsonia@comcast.net**> using the **SMTP protocol**. This same process was repeated for 5 IP addresses in total, with a total of **214 email** events distributed unevenly among the addresses. The **last victim** <**jberman1@gmail.com**> was contacted by **01:01:12.900**. All transactions occurred at **port 587**.

3.3 ClickFraud

3.3.1 Evidence

There was a high traffic in HTTP protocol which suggested further research. We first searched for HTTP requests to the server “www.generalamuse.com” in order to get its destination IP address. The following statement was used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" info = "*generalamuse*" AND info = "*response*"
```

Figure 23 – Searching for target IP address for ClickFraud attack part 1

The result of the above indicated that the victim's IP corresponded to 98.126.71.122.

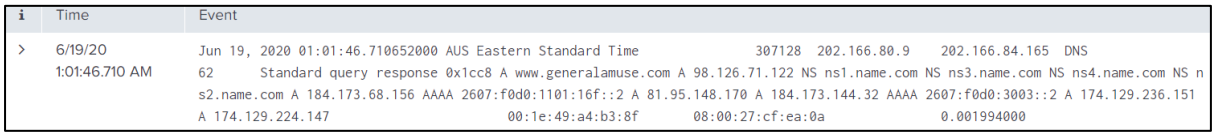


Figure 24 – Searching for target IP address for ClickFraud attack part 2

Next, we used the following statement in order to get the attacker's IP:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 98.126.71.122 AND protocol = "HTTP"
```

Figure 25 – Searching for attacker IP in ClickFraud attack part 1

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 98.126.71.122 AND protocol = "HTTP" | dedup src_ip | table
  src_ip
```

Figure 26 – Searching for attacker IP in ClickFraud attack part 2

The attacker's IP was 202.166.84.165. The result of the above provided us with **38 requests related to ClickFraud attack**.

3.3.2 Field extraction

We investigated the URI strings used by the attacker in the ClickFraud attack context. We used the same extracted field from Command and Control attack.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 98.126.71.122 AND protocol = "HTTP" | dedup URI | table URI
```

Figure 27 – Searching for URI string of ClickFraud attack

Figure 28 – URI string for Command and Control attack

3.3.3 Summary

We run the following statement to get start and end times of the attack, including both victim and attacker IPs.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" dst_ip = 98.126.71.122 AND protocol = "HTTP" | stats earliest(_time)
  as starttime, latest(_time) as endtime by dst_ip, src_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M:%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M
  :%S.%Q") | eval victim = dst_ip | eval attacker = src_ip | table victim, attacker, start, finish
```

Figure 29 – Searching for Start and End time of ClickFraud attack

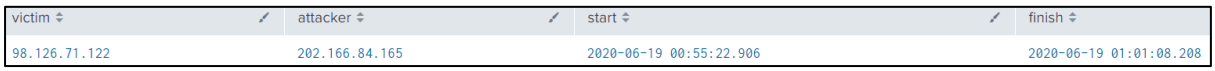


Figure 30 – Summary of Start and End time of ClickFraud attack

Finally, we noticed that all the requests were directed to the port 80. A statement similar to Figure 27 was used in order to get the destination port.

3.3.4 Attack Narrative

At 00:55:22.906, **attacker 202.166.84.165** made the first attempt to connect to "**www.generalamuse.com**" through DNS protocol to 202.166.80.9. It received a response from the DNS server at 00:55:22.898 and the IP address corresponded to **98.126.71.122**, the **victim**. Once

connection was established, the bot used the **HTTP protocol** to send the **first attack** at **00:55:22.906** to the victim containing the GET method alongside the **URI = /gen.php**. There was a total of **38 events** directed to the same address. The **last attack** occurred at **01:01:08.208**. All transactions occurred at **port 80**.

3.4 IRC

3.4.1 Evidence

We were interested the IRC POST requests which attackers might leverage in order to deliver malicious code to their targets, where the IRC server can be directly connected to the botmaster (Gu et al., 2008). We performed the following sentence to confirm that the attacker's IP was 202.166.84.165:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "IRC" AND info = "*POST*" | dedup src_ip | table src_ip
```

Figure 31 – Searching attacker's IP in IRC attack

To calculate the IRC servers, we performed the following statement, which resulted in **17 servers** (list can be seen in Table 1):

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv" protocol = "IRC" AND info = "*POST*" | dedup dst_ip | table dst_ip
```

Figure 32 – Searching for destination IPs of IRC attack

The following statement was used to create the IRC section of Table 1, which shows the start and end time for each targeted IP. The result of this statement resulted in **31 POST requests**.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv"  protocol = "IRC" AND info = "*POST*" | stats earliest(_time) as
    starttime, latest(_time) as endtime by dst_ip, src_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M:%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M:%S
    .%Q") | eval attacker = src_ip | eval victim = dst_ip | table attacker, victim, start, finish
```

Figure 33 – Searching for Start and End time of IRC attack targeted IPs

3.4.2 Summary

The following statement provides a summary of the IRC attack including the attacker's IP, start and end time of the attack:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap.csv"  protocol = "IRC" AND info = "*POST*" | stats earliest(_time) as
    starttime, latest(_time) as endtime by src_ip | eval start = strftime(starttime, "%Y-%m-%d %H:%M:%S.%Q") | eval finish = strftime(endtime, "%Y-%m-%d %H:%M:%S.%Q") |
    eval attacker = src_ip | table attacker, start, finish
```

Figure 34 – Searching for summary for IRC attack

attacker	start	finish
202.166.84.165	2020-06-19 00:55:21.813	2020-06-19 01:01:59.756

Figure 35 – Summary of IRC attack

3.4.3 Attack Narrative

At 00:55:21.809, **attacker 202.166.84.165** made the first attempt to connect to **200.171.4.222**, the **first victim**. Once connection was established, the bot used the **IRC protocol** to send the **first attack** at **00:55:21.813** to the victim containing the POST method. There was a total of **31 events** directed to **17 IP addresses**. The **last attack** occurred at **01:01:59.756** directed to **58.42.247.143**. All transactions occurred at **port 6667**.

4 Consequences

The following consequences are presented by considering the attacks' potential impacts over confidentiality, integrity and availability.

- **Command and Control (HTTP and IRC based)**: Once attacker establishes a Command and Control channel, they will be able to access the targeted system remotely, which might affect the **confidentiality** of data, as this attack can be launched for espionage. Moreover, the attacker might manipulate the data at their will, violating the **integrity** principle, and they might also turn the victim's system down affecting the **availability** (Gardiner, 2014).
- **SPAM**: The spam attack can be related to impacts on **availability** in two ways. Firstly, the attacker might bring down the resources of the targeted server with a massive amount of spam email, which might saturate the bandwidth of the network. Secondly, from a user perspective, all the targeted email recipients might perceive a loss in the availability of their email services, thus not being able to receive anything besides spam (Sumra et al., 2014).
- **ClickFraud**: This attack might have an impact on the **integrity** of the web content shown in the victim's browser system once the malware is installed. Moreover, other perpetrators might use this attack to raise the costs of competitor's ads, for which they fill the ads with clicks, which might impact **availability** of the system (Juels, 2007).

5 Patterns of attack

The following patterns of attack were taken from our previous analysis according to the main features used to find the evidence of the attacks, in addition to the extracted fields. The specific values of each feature can be seen in each Attack narrative section.

- Command and Control: src_ip + dst_ip + dst_port + protocol + URI
- SPAM: src_ip + dst_port + protocol
- ClickFraud: src_ip + dst_ip + dst_port + protocol + URI
- IRC: src_ip + dst_pot + protocol

Other sources of data could be gathered if more data was available. For C2 attack, either HTTP or IRC based, packet features could be used for detecting the attack, such as "length in bytes, number of packets, flow duration" and "average bytes per packet" (Beigi, 2014). For SPAM attack, focus could be set on the subject's words of the email, the header's "hour of day", the "number of all URLs" and "payload" features (Alqatawna, 2015). For ClickFraud attack, "clicks per publisher", "downloads per publisher" and variations in "clicks in time" could be measured too (Liu, 2009).

6 Countermeasures

Countermeasures	C2	SPAM	ClickFraud	IRC
Access management system(*)	X			X
Information encryption(*)	X			X
Segregation of duties(*)	X		X	X
Approval checkpoints(*)	X		X	X
Scheduled updates(*)	X	X	X	X
Incident management system(*)	X	X	X	X
Disaster recovery plan(*)	X	X	X	X
Web proxy(*)	X			X
DNS security(*)	X			X
IDS(*)	X	X	X	X
IPS(*)	X	X	X	X
HIPS(*)	X	X	X	X
Email security system(*)		X		
Antivirus system(*)		X	X	
SETA program(*)(**)	X	X	X	X
Honeypot/Honeynet(**)	X	X	X	X
SMTP proxy (***)		X		

(*) (Zhou, 2020)
(**) (Liu, 2009)
(***) (ORACLE, 2010)

Table 3 – Countermeasures per attack

Conclusion

We considered for this case to be an example of a botnet architecture, a network of infected machines that can be remotely controlled by Command and Control channels (Zhou, 2020). We considered that the botnet used IRC and HTTP channels, establishing communication through at least one known server. Once connected, the botmaster was able to infect the targeted machines with malicious payload (Liu, 2009).

Once infected, 202.166.84.165 (bot) was set up to start the botnet propagation. The bot can be used to send commands to be executed by victims (IRC – push-based) and to send requests for victims to download malware (HTTP – pull-based) in order to recruit more bots (Gu et al., 2008). Both of these attacks can be related to the Command and Control section of the Cyber Kill Chain (CKC), where intruders may gain access and manipulate the victim's system remotely (Zhou, 2020)

Bots can also be used for spreading massive SPAM and as a ClickFraud method (Liu, 2009), both cases for this scenario. In the case of SPAM, the bot was used to send unsolicited email messages to a large set of recipients as a campaign to gain profits and/or to recruit more bots (Gu et al., 2008). SPAM attack can be related to the delivery step of the CKC, where the bot may transmit a virus to the targeted systems (Zhou, 2020).

Regarding ClickFraud, the bot may be used to install advertisement programs targeting the browser of the system for increasing the click through rate. Thus, the victim's system will perform clicks on links periodically, counting as valid clicks (Liu, 2009). ClickFraud attack can be associated to the installation stage of the CKC, where malware gets installed in the targeted system.

Finally, we noticed that our analysis was constrained by the amount of data, so there could be even more findings that could have been gathered about other sources of attack.

References

Alqatawna, J, Faris, H, Jaradat, K, Al-Zewairi, M & Adwan, O 2015, 'Improving Knowledge Based Spam Detection Methods: The Effect of Malicious Related Features in Imbalance Data Distribution', *International Journal of Communications, Network and System Sciences*, vol. 8, pp. 118-129.

Gardiner, J, Cova, M & Nagaraja, S 2014, *Command & Control: Understanding denying and detecting*, University of Birmingham, viewed 7th September 2020, < https://arxiv.org/ftp/arxiv/papers/1408/1408.1136.pdf>.

Gu, G, Zhang, J & Lee, W 2008, 'BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic' *Proceedings of the 15th Annual Network and Distributed System Security Symposium*, viewed 7th September 2020, < https://corescholar.libraries.wright.edu/cse/7/>.

Johari, A 2020, *GET vs POST: What is the difference between GET and POST method?*, edureka!, viewed 7th September 2020, < https://www.edureka.co/blog/get-and-post-method/>.

Juels, A, Stamm, S & Jakobsson, M 2007, 'Combating Click Fraud via Premium Clicks', *16th USENIX Security Symposium*, pp. 17-26.

Kantardzic, M, Walgampaya, C, Wenerstrom, B, Lozitskiy, O, Higgins, S & King, D 2008, 'Improving Click Fraud Detection by Real Time Data Fusion', *2008 IEEE International Symposium on Signal Processing and Information Technology*, vol. 2008, pp. 69-74.

Liu, J, Xiao, Y, Ghaboosi, K, Deng, H & Zhang, J 2009, 'Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures', *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, pp. 1-11.

McFeters, N, Rios, B & Carter, R 2008, *URI Use and Abuse*, Black Hat Briefings, viewed 7th September 2020, <https://www.blackhat.com/presentations/bh-dc-08/McFeters-Rios-Carter/Presentation/bh-dc-08-mcfeters-rios-carter.pdf>.

ORACLE 2010, *SMTP Proxy*, ORACLE, viewed 7th September 2020, < https://docs.oracle.com/cd/E19047-01/sunscreen32/806-6347/6jfa0g88r/index.html>.

Sumra, I, Hasbullah, H & AbManan, J 2014, 'Attacks on Security Goals (Confidentiality, Integrity, Availability) in VANET: A Survey', *Vehicular Ad-hoc Networks for Smart Cities*, vol. 306, pp. 51-61.

Zhou, V 2020, Teaching Sessions 1, 4, 6 and 7, COMP90073 Security Analytics, learning materials on Canvas, The University of Melbourne, viewed 7th September 2020.

Appendix 1

<jberman1@gmail.com>	<fredd@bhconcepts.com>	<terry@ampower.com.au>
<home@jlauer.de>	<mark@hgidahohomes.com>	<brendanebr@hotmail.com>
<ducesout64@yahoo.com>	<dkhughes@woh.rr.com>	<racerjeffrey@aol.com>
<dupnockt@yahoo.com>	<claudioghion@tiscali.it>	<derekrainford@yahoo.co.uk>
<brwneyesnikki@yahoo.com>	<yigal@aol.com>	<flybd@yahoo.com>
<rbdagondon2004@yahoo.com>	<cblackpdr@yahoo.com>	<dbrownbear4@aol.com>
<lanie60416@yahoo.com>	<wutznit4me@gmail.com>	<manewman@another.com>
<bear315@msn.com>	<clemrgraham@yahoo.com>	<mb246810@yahoo.com>
<lynnrobin24@sbcglobal.net>	<mail@ctbergman.com>	<stephenmorran@att.net>
<rhduke@gmail.com>	<degen123456789@yahoo.de>	<adelebonge@gmail.com>
<hi-rosinante@zeus.eonet.ne.jp>	<happilyymarried@aol.com>	<pixiefairy91@gmail.com>
<jawrady@yahoo.com>	<vgchuma1@yahoo.co.uk>	<rubios2@optonline.net>
<bbstrow@comcast.net>	<hinje1225@aol.com>	<gloria_w@comcast.net>
<mbwagne@gmail.com>	<ronjames56@aol.com>	<gmedlin0001@gmail.com>
<a454philip@aol.com>	<oscarv3@yahoo.com>	<donna@1dj.com>
<kuwatani@sbcglobal.net>	<romhanyi2@fibermail.hu>	<alan.curdy@btopenworld.com>
<larrypw@peoplepc.com>	<rickp@montcalm.edu>	<galactuseats@yahoo.com>
<khaledsyfulla@yahoo.com>	<kdiggy1@gmail.com>	<fryedaddy18@yahoo.com>
<dmalaby1@yahoo.com>	<dlpetersburg@gmail.com>	<lkdodd@earthlink.net>
<krfeldman@cox.net>	<trash@etifish.com>	<laurenmom@hotmail.com>
<yuuna09126@yahoo.co.jp>	<perrothead_18@yahoo.com>	<sjt1dcx@yahoo.com>
<rrsalvo@yahoo.com>	<i.miliuviene@gmail.com>	<tbross36@hotmail.com>
<german_isaac@yahoo.com>	<bmwrider12@hotmail.com>	<demperri@yahoo.com>
<jbyrd5884@verizon.net>	<john.hurren@ntlworld.com>	<cboy4u@aol.com>
<jay.space@gmail.com>	<turret_press_1949@yahoo.com>	<ppitts@weber.edu>
<gfredericks@verizon.net>	<brad@besh.com>	<ourkatie83@yahoo.co.uk>
<irene@dailymail.co.uk>	<gdjones1341@gmail.com>	<w.tatzel@web.de>
<dedodson1@yahoo.com>	<armond.thomas@yahoo.com>	<markrichter35@yahoo.com>
<sledbaby@hotmail.com>	<mbaker5813@aol.com>	<curbitstl@yahoo.com>
<artind@web.de>	<icykc@hotmail.com>	<sabrina.schiller@gmail.com>
<lars.westerlund@telia.com>	<imrank80@hotmail.com>	<jeffpestano@yahoo.com>
<xfmny912@yahoo.co.jp>	<sprinkguy704@yahoo.com>	<dadnkatenjack@yahoo.com>
<billmac@charter.net>	<bobcummins2@gmail.com>	<pracely1@aol.com>
<liaison@usa.com>	<faibreeze@yahoo.com>	<rjc@xplornet.com>
<dsmartgirl2002@yahoo.com>	<denny@mlode.com>	<desertrat0263@yahoo.com>
<nemanitawake@yahoo.com>	<curly421@aol.com>	<kristenlavigne@yahoo.com>

<david.duthie@mac.com>	<noriyuki11922910@yahoo.co.jp>	<d_fellingham1@tiscali.co.uk>
<mari-525@ngs2.cncm.ne.jp>	<mlawson@apachemills.com>	<jkkdcgillham@yahoo.com>
<dkrogers@hughes.net>	<ihab_naa@hotmail.com>	<renchtrnr@yahoo.com>
<brianlxndr@yahoo.com>	<james.erikson@sbcglobal.net>	<pmergenthal@aol.com>
<chaz@ichaz.com>	<terpsichore99@aol.com>	<meredithejohnson@comcast.net>
<stuart4511@bellsouth.net>	<jack_manu@o2.co.uk>	<ahm203040@yahoo.com>
<firstbond@bellsouth.net>	<eugeherald@aol.com>	<uduncan@deltagastro.net>
<twam4@aol.com>	<theelpanther@yahoo.com>	<rgroth@permatite.com>
<ramonbarron@mac.com>	<xschmitz@arcor.de>	<tmiralem@gmail.com>
<jennifer.hersey@umit.maine.edu>	<pthack1225@aol.com>	<wayne651@live.com>
<michelledowding1@yahoo.com>	<acacia.lodge@xtra.co.nz>	<ag@aligureli.com>
<mammbe@gmail.com>	<larryrock@rogers.com>	<khadidja_kadri@yahoo.fr>
<cuhum@aol.com>	<petejcamp@aol.com>	<sh_sokolovsky@yahoo.com>
<pedmansouri@yahoo.com>	<abbassi.hamza@yahoo.fr>	<robertsmit@lantic.net>
<robert@troubleshootersusa.com>	<gfarndale@blueyonder.co.uk>	<johnraywesley@att.net>
<june610@verizon.net>	<christian.gerdes@gmx.de>	<fandm3721@aol.com>
<djmjmason@comcast.net>	<gregalford@aol.com>	<richard1la@aol.com>
<billygrove@sbcglobal.net>	<frank.breitkreutz@gmx.de>	<s.kaxuxuena@parliament.gov.na>
<piotr.korniak@wp.pl>	<ingogr@online.no>	<mjosparky@aol.com>
<deborah.crumlish@sbcglobal.net>	<slrydaz23@yahoo.com>	<rriley8184@comcast.net>
<brian_anderson49@yahoo.com>	<ti613@embarqmail.com>	<wintervers@aol.com>
<ssullivan45@optonline.net>	<jl.rojo@free.fr>	<uhlenkot@cableone.net>
<bigg_cali@yahoo.com>	<nschaef@sbcglobal.net>	<klrector@aol.com>
<john006@bigpond.com>	<meikle@woosh.co.nz>	<smartypantz214@aol.com>
<gerrit.teters@gmail.com>	<peachnpresto@aol.com>	<timiten@gmail.com>
<re7man3@aim.com>	<alan.linney@virgin.net>	<ronaldflood@brmemc.net>
<noworries98109@yahoo.com>	<steijl@pt.lu>	<jckelly@cox.net>
<peppinocolucci@libero.it>	<robertscarlette@yahoo.com>	<rcox@brcom.net>
<lbrown020@triad.rr.com>	<redbob1955@aol.com>	<karaii@aol.com>
<hewitt2@charter.net>	<christopherppallen@yahoo.com>	<baudetylira@terra.cl>
<jordy@gondtc.com>	<bbmrs1103@yahoo.com>	<tbmccormick@me.com>
<tgreen81@cox.net>	<dbauer3671@sbcglobal.net>	<livlif77@aol.com>
<johnso748@aol.com>	<rose.34@gmail.com>	<bonde007@earthlink.net>
<niko162@yahoo.com>	<eboyle@neo.rr.com>	<gargantua1953@yahoo.fr>
<varetto@libello.com>	<powelltony7@aol.com>	<bobbarb22@aol.com>
<nickandsonia@comcast.net>		