COMP90073 Security Analytics, Semester 2 2020

Project 1 : Detecting Cyberattacks in network traffic data

Submitted by :
Name : Ribhav Shridhar
Student ID : 1037144
Username : shridharr

Introduction :

Cybercrime is big business. By many estimates, cybercrime is now a US$1 trillion-dollar industry. Every organization with digital assets is vulnerable to attack and the growing sophistication of cyber criminals and their evolving tactics only increases the chance of a security breach involving the theft of sensitive data. Effective cyber defence must withstand changes to adversaries' tactics and tools that traditional, non-integrated "best of breed" approaches cannot address. It must also protect against advanced known threats, as well as unknown threats, which can be challenging to address with legacy solutions.

In this report we are studying and analysing a packet capture file to identify cyberattacks by leveraging the analytics capabilities of Splunk while recognizing and analysing traffic patterns before and after the attack has taken place.
In this report we are also searching for evidences of cyberattacks and hunting the attack sources and their targets.

U.S. COMPANIES VULNERABLE TO ATTACK

"There are two kinds of big companies in the United States. There are those who've been hacked and those who don't know they've been hacked."

— FBI Director James Comey in the Washington Times, Nov. 3, 2014

Data Description & EDA

Data set

The dataset for project 1 includes a packet capture(pcap) file of network traffic for a particular network which was a victim of various kinds of cyber-attacks. The pcap file also contains network data before and after the malware infection.
The file contains 323,154 events in the form of network data.
The first and last events from the dataset are given below –
Each event consists of fields such as :

1. time of event
2. IP address of Destination Machine
3. IP address of Source Machine
4. Port number of Destination Machine
5. Port number of Source Machine
6. Information of the packet being sent or request

7. Protocol being used for communication

Ingesting the PCAP File

To ingest the PCAP file to Splunk, we had to edit some scripts for the “SplunkForPCAP” package.
These can be accessed from “*C:\Program Files\Splunk\etc\apps\SplunkForPCAP\bin*” in windows. Altering to Windows Batch files, “pcap2csv_1_11_x_1_12_x.bat” file, to hardcode your own file path for the pcap and where to import the csv generated.

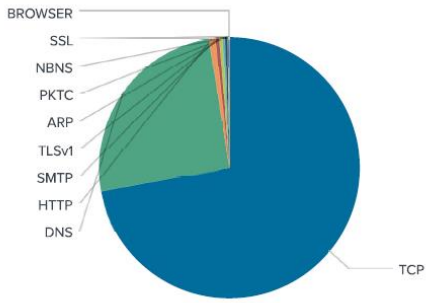
```
for /f "tokens=2 delims== " %%a in ('findstr "path" "%SPLUNK_HOME%\etc\apps\SplunkForPCAP\local\inputs.conf"') do (
for /F %%f in ('dir /b "%%a\*.pcap"') do "%programfiles%\Wireshark\tshark" -o tcp.calculate_timestamps:TRUE -r "C:\Users\hp\Desktop\Uni\Sem 4\Security
Analytics\Data\traffic-capture-sasm2.pcap" -T fields -e frame.time -e tcp.stream -e frame.number -e ip.src -e ip.dst -e _ws.col.Protocol -e tcp.srcport -e tcp.dstport -e tcp.len -e
tcp.window_size -e tcp.flags.syn -e tcp.flags.ack -e tcp.flags.push -e tcp.flags.fin -e tcp.flags.reset -e ip.ttl -e _ws.col.Info -e tcp.analysis.ack_rtt -e vlan.id -e eth.src -e
eth.dst -e http.time -e dns.time -e rpc.time -e smb2.time -e smb.time -e tcp.time_delta -e tcp.time_relative > "%SPLUNK_HOME%\etc\apps\SplunkForPCAP\PCAPcsv\Assignment1.csv"
move  "%%a\*.pcap" "%SPLUNK_HOME%\etc\apps\SplunkForPCAP\PCAPConverted\" >nul 2>&1
)
```

This issue is mostly due to “*SplunkForPCAP*” using *Tshark* version 2.

EDA

We have performed some Exploratory Data Analytics to analyse the data set and fetch its most important patterns and characteristics.
Some of the top metrics are summarized below –

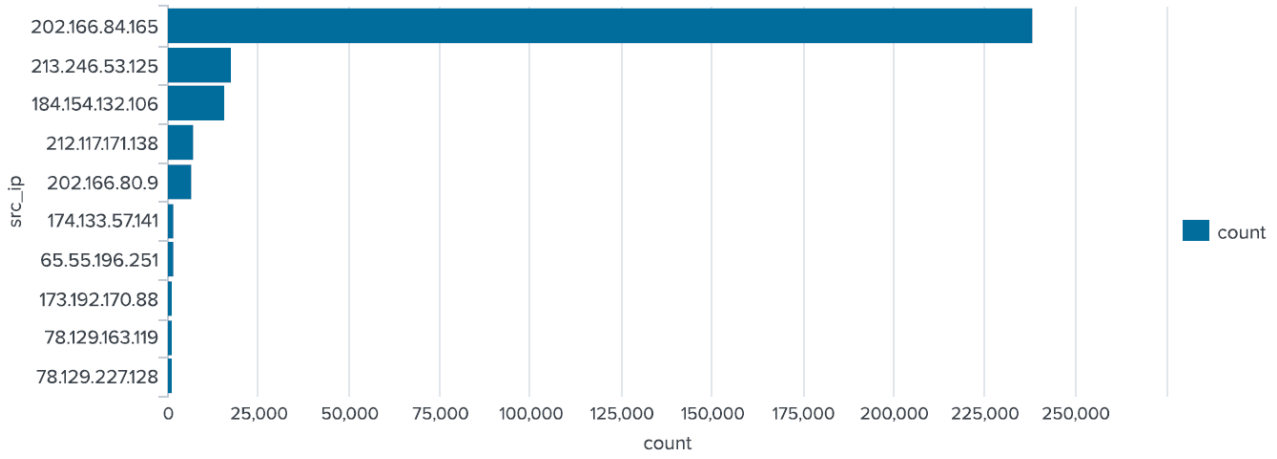
1. Top Protocol by number of events



Statistics :

protocol	count	percent
TCP	229430	72.058292
DNS	80747	25.360637
HTTP	2550	0.800892
SMTP	1562	0.490586
TLSv1	1332	0.418348
ARP	906	0.284552
PKTC	857	0.269163
NBNS	296	0.092966
SSL	229	0.071923
BROWSER	111	0.034862

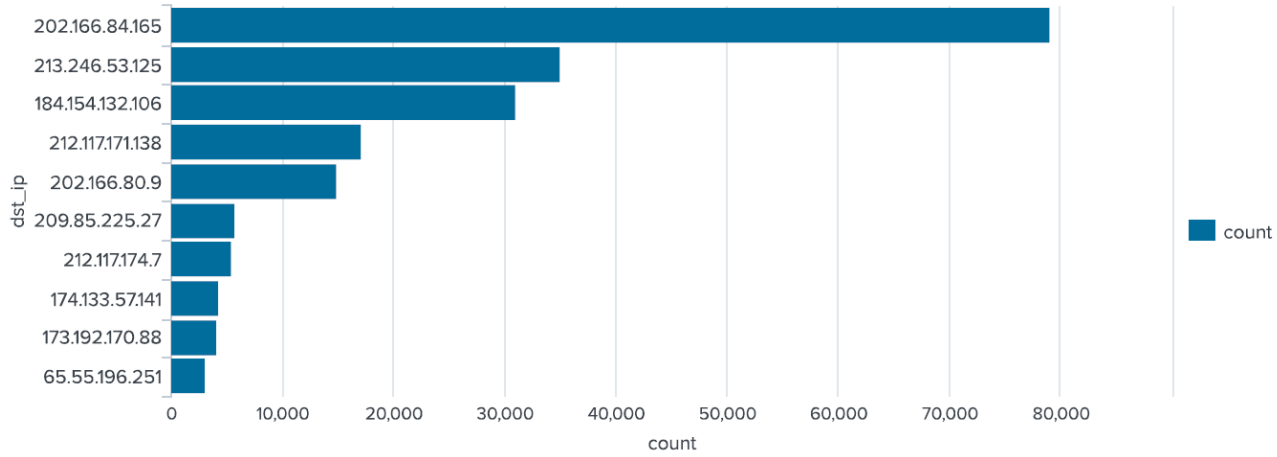
2. Top Senders by IP Addresses by number of events



Statistics :

src_ip	count	percent
202.166.84.165	238462	75.108744
213.246.53.125	17566	5.532790
184.154.132.106	15594	4.911666
212.117.171.138	7428	2.339609
202.166.80.9	6787	2.137712
174.133.57.141	1653	0.520648
65.55.196.251	1627	0.512459
173.192.170.88	1493	0.470253
78.129.163.119	1414	0.445370
78.129.227.128	1401	0.441275

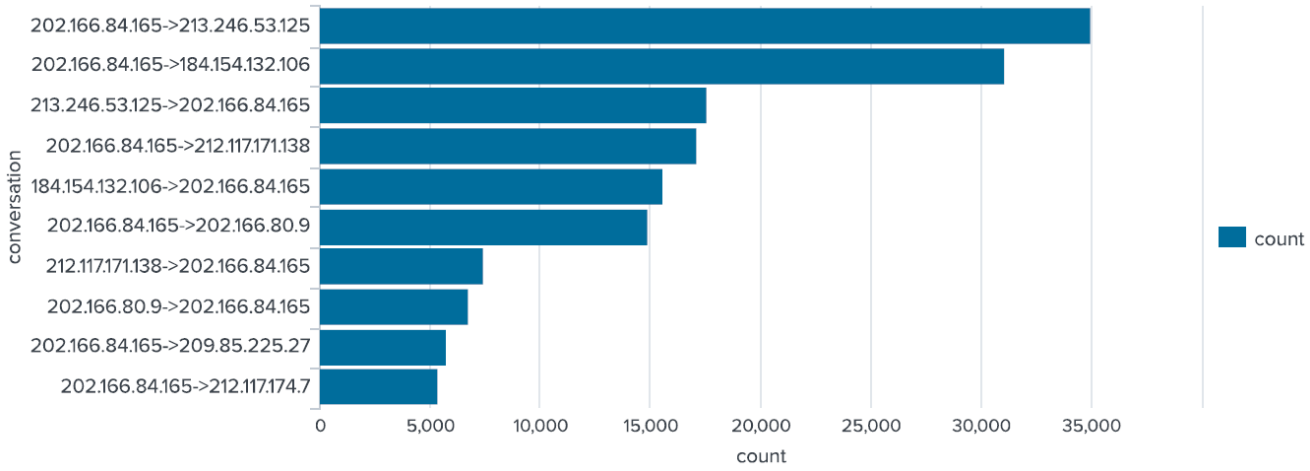
3. Top Receivers by IP Addresses by number of events



Statistics :

dst_ip	count	percent
202.166.84.165	79027	24.891256
213.246.53.125	35018	11.029673
184.154.132.106	31090	9.792465
212.117.171.138	17138	5.397982
202.166.80.9	14870	4.683627
209.85.225.27	5790	1.823685
212.117.174.7	5412	1.704626
174.133.57.141	4292	1.351858
173.192.170.88	4108	1.293903
65.55.196.251	3132	0.986491

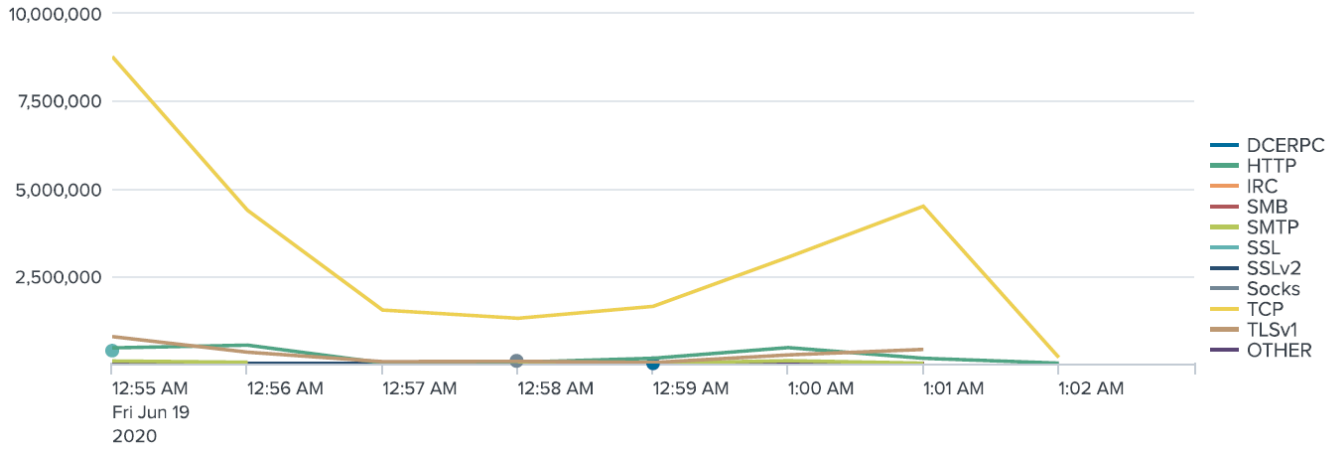
4. Top communication between two IP addresses by count of events



Statistics :

conversation	count	percent
202.166.84.165->213.246.53.125	35018	11.029673
202.166.84.165->184.154.132.106	31090	9.792465
213.246.53.125->202.166.84.165	17566	5.532790
202.166.84.165->212.117.171.138	17138	5.397982
184.154.132.106->202.166.84.165	15594	4.911666
202.166.84.165->202.166.80.9	14870	4.683627
212.117.171.138->202.166.84.165	7428	2.339609
202.166.80.9->202.166.84.165	6787	2.137712
202.166.84.165->209.85.225.27	5790	1.823685
202.166.84.165->212.117.174.7	5412	1.704626

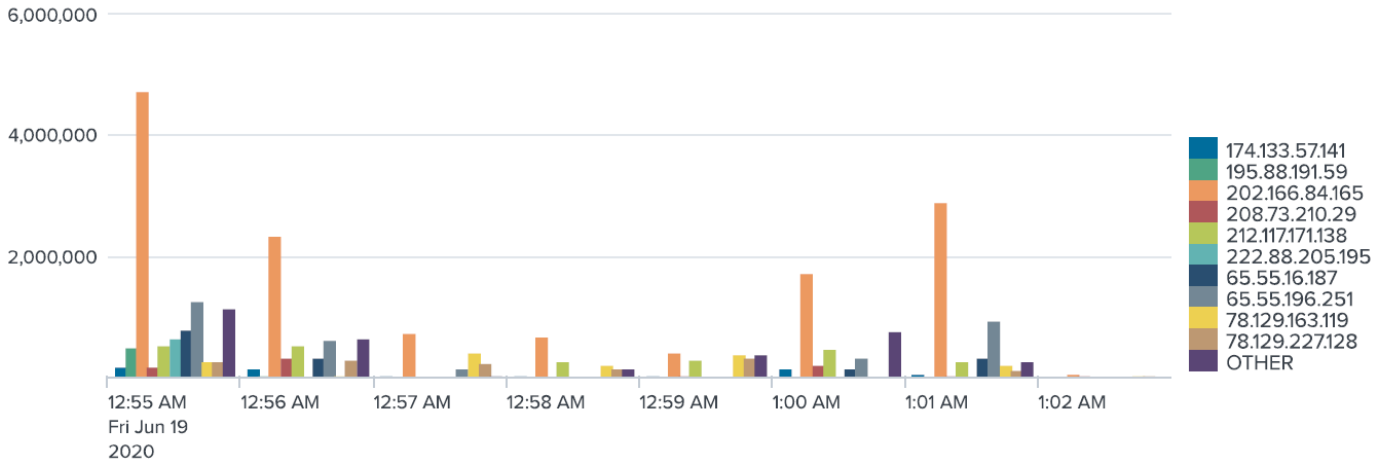
5. Top Protocol by Packet length



Statistics :

_time	DCERPC	HTTP	IRC	SMB	SMTP	SSL	SSLv2	Socks	TCP	TLSv1	OTHER
2020-06-19 00:55:00	18980	444676	6429	3036	71968	368316	23319		8768285	765922	7428
2020-06-19 00:56:00	4380	521591	6141	3036	37499		516		4380123	327109	876
2020-06-19 00:57:00		23831	563				129		1525148	54984	
2020-06-19 00:58:00		37328			833		129	77383	1290493	65089	10
2020-06-19 00:59:00	5840	154342	1679		12111				1629410	25220	5
2020-06-19 01:00:00		459119	1671	1518	78647		1419		3034012	243909	2389
2020-06-19 01:01:00		151488	597	1518	10622		645		4489398	400594	480
2020-06-19 01:02:00		7144							174676		

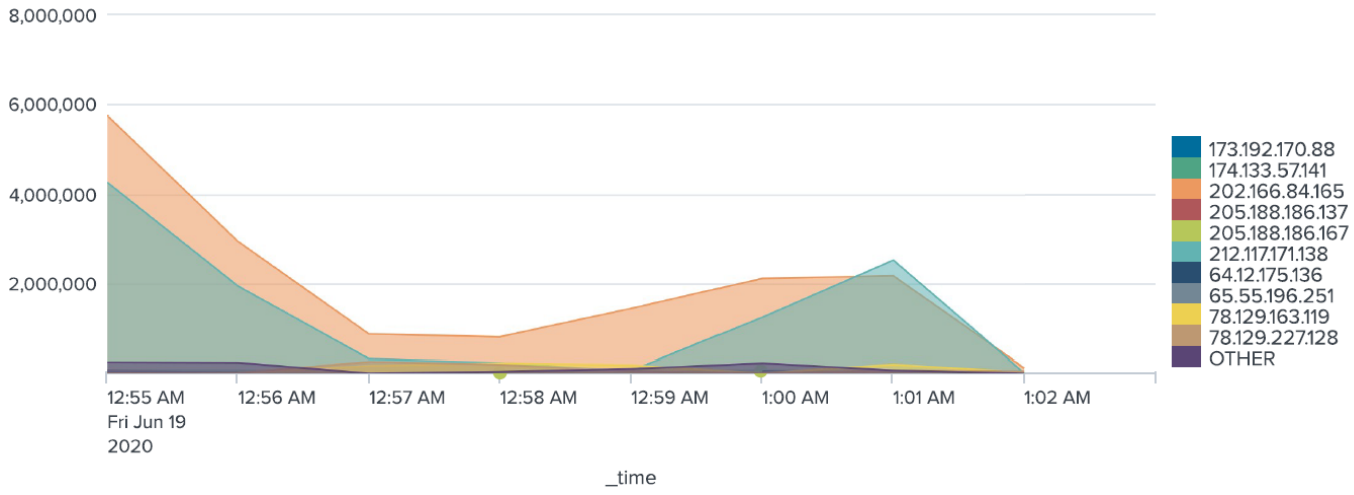
6. Top sender by Packet Length



Statistics :

_time	174.133.57.141	195.88.191.59	202.166.84.165	208.73.210.29	212.117.171.138	222.88.205.195	65.55.16.187	65.55.196.251	78.129.163.119	78.129.227.128	OTHER
2020-06-19 00:55:00	162083	494614	4732741	186283	522302	657904	791899	1257659	261309	261448	1150117
2020-06-19 00:56:00	148748	41805	2353304	309305	515533		312561	626344	39352	301544	632775
2020-06-19 00:57:00	17650		740782		1531			156277	424322	230175	33918
2020-06-19 00:58:00	16172		672923		267905		4376		209084	159388	141417
2020-06-19 00:59:00	37352		401527	32055	287599				388227	314710	367137
2020-06-19 01:00:00	153182		1722264	218412	479069		157506	330058	528	528	761137
2020-06-19 01:01:00	48229		2896620	30981	253887		312568	937509	192274	108088	275186
2020-06-19 01:02:00	3530		70304	30058	47				36226	35434	6221

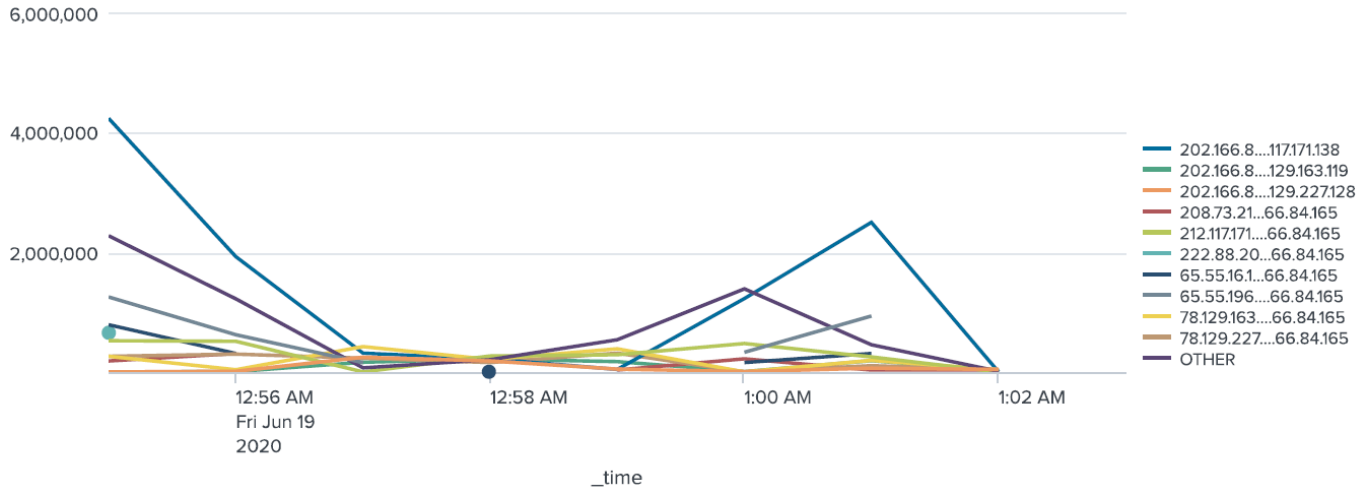
7. Top receiver by Packet Length



Statistics :

_time	173.192.170.88	174.133.57.141	202.166.84.165	205.188.186.137	205.188.186.167	212.117.171.138	64.12.175.136	65.55.196.251	78.129.163.119	78.129.227.128	OTHER
2020-06-19 00:55:00	17280	68020	5745618	73468	21784	4252192	28194	27204	1728	1728	241143
2020-06-19 00:56:00	40364	68736	2927967	2144	25816	1930674	21740	11104	8912	11768	232046
2020-06-19 00:57:00	3052	7160	863873			312626		2808	164664	247968	2504
2020-06-19 00:58:00	2404	7160	798342		144	223516			215552	185624	38523
2020-06-19 00:59:00	22136	16468	1427080			43396			175592	47848	96087
2020-06-19 01:00:00	63852	70168	2100420	8086	40102	1233330	44536	34896	4352	4224	218718
2020-06-19 01:01:00	21024	19332	2158722	8006		2509324	2528	15796	191520	62912	66178
2020-06-19 01:02:00	1280	1432	111516			28			21048	44728	1788

8. Top communication between 2 IP addresses by packet length



Statistics :

_time	202.166.84.165->212.117.171.138	202.166.84.165->78.129.163.119	202.166.84.165->78.129.227.128	208.73.210.29->202.166.84.165	212.117.171.138->202.166.84.165	222.88.205.195->202.166.84.165
2020-06-19 00:55:00	4252192	1728	1728	186283	522302	657904
2020-06-19 00:56:00	1930674	8912	11768	309305	515533	
2020-06-19 00:57:00	312626	164664	247968		1531	
2020-06-19 00:58:00	223516	215552	185624		267905	
2020-06-19 00:59:00	43396	175592	47848	32055	287599	
2020-06-19 01:00:00	1233330	4352	4224	218412	479069	
2020-06-19 01:01:00	2509324	191520	62912	30981	253887	
2020-06-19 01:02:00	28	21048	44728	30058	47	

Attacks:

1. Botnet Command and Control

Cyber-attacks have substantially risen in the recent years. Command and control, (C&C) is one of the most dangerous attacks and is executed using DNS.
The attack is started by infecting a machine/server that might be protected by a firewall. Hackers can achieve this in various ways - Using a phishing email that lures the user into following a link that leads to a dangerous website or downloading a file that executes a malicious software, making use of security faults in web browsers, or using other infected programs.

Botnet life cycle comprises of 5 different phases. These are –

- Initial Injection : This phase is the start of the Botnet life cycle. In this, an hacker looks to infect exposed hosts in the network with the help of spam messages, phishing emails, etc.
- Secondary Injection – In this phase, the infected server executes the script inserted by the attacker and installs it.
- Connection – In the connection phase, a link is formed between an infected machine and the C2 server for communicating instructions from the Hacker to the infected machine and also transmitting data from the infected machine to the Botmaster.
- Command and Control Server – Once the connection is formed, the Botmaster uses the C&C channel to send out instructions to the machine is the botnet.
- Update and maintenance – This phase contribute to maintaining the active machines in the botnet and keeping them updated.

Botnet life cycle

Phases		Instances	Resilience techniques
Injection & Spreading		-Distribution of malicious emails -Software vulnerabilities -Instant Messaging -P2P File sharing Network -Other Botnets	-Using trusted process -Trivial name-based obfuscation -Rootkit Techniques -Reduce Security rules -Reduce system capability -Installing antivirus software -Incorporated antidebugging & antivirtualization -Variant Spreading Techniques -Polymorphism & Metamorphism -Continuous bot upgrade
Command & Control	Model & Topology	-Centralized »Single Star »Multiserver Star »Hierarchical -Distributed »Random	-DNS techniques -Multiple URLs -Encryption Techniques -Dead drop -Variant C&C
	Application & Protocol	-IRC -HTTP -IM -P2P	
	Communication initiation	-Push Method -Pull Method	
	Communication direction	-Inbound -Bidirectional	
Botnet application		-DDoS attacks -Spamming & Spreading malwares -Espionage -Hosting malicious applications & activities	-Exposure limitation -Retaliation techniques -Camouflaged messages - Anonymization techniques

Methodology and pattern discovery

For the given dataset, after identifying that "*finalcortex.com*" as the C2 server, to do analysis on the data and the victims of the attack, we needed to obtain the IP address of the C2 server.
DNS based technique is one of the most popular to detect botnets. This technique involves the use of DNS information generated by a botnet to identify the anomalies. All the machines in the botnet start connection with the C2 server .To access the server, bots perform DNS queries to locate the server, which is hosted by a DNS supplier.
The following query was used to find the IP address of "*finalcortex.com*"

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" finalcortex.com protocol = dns
```

The following screenshot shows the output of the above query, giving the IP address to the C2 server.

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 00:1e:49:a4:b3:8f 08:00:27:cf:ea:0a 0.013943000 dst_ip = 202.166.84.165 info = Standard query response 0xd5c2 A finalcortex.com A 31.192.109.167 NS ns3.cnm... protocol = DNS src_ip = 202.166.80.9
>	6/19/20 12:58:19.517 AM	Jun 19, 2020 00:58:19.517704000 AUS Eastern Standard Time 149113 202.166.84.165 202.166.80.9 DNS 128 Standard query 0xd5c2 A finalcortex.com 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f dst_ip = 202.166.80.9 info = Standard query 0xd5c2 A finalcortex.com protocol = DNS src_ip = 202.166.84.165

The IP address obtained is 31.192.109.167.

We will be using this IP address as the attacker IP to find the bots in the bot army. That can be achieved by monitoring the HTTP requests between the two. The following query was used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = http dst_ip = 31.192.109.167 | dedup dst_ip | table src_ip,dst_ip
```

The output generated of the above query:

src_ip	dst_ip
202.166.84.165	31.192.109.167

Using the above query, it can be concluded that the new infected machine is 202.166.84.165.

All of the 54 events are between the two IP addresses only.

To get the URI of the events, the following query is utilized :

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = http dst_ip = 31.192.109.167 | rex "(?<URI>\/+\w+\/+\w+\.+\w+)"
```

The below screenshot shows 5 of the returned 54 events:

Time	Event
6/19/20 12:58:00.085 AM	Jun 19, 2020 00:58:00.085710000 AUS Eastern Standard Time 5807 137134 202.166.84.165 31.192.109.167 HTTP 1324 80 229 64240 0 1 1 0 0 128 POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000996000 0.007978000 URI = /snapbn/gate.php dst_ip = 31.192.109.167 dst_port = 80 info = POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) protocol = HTTP src_ip = 202.166.84.165 src_port = 1324
6/19/20 12:57:36.507 AM	Jun 19, 2020 00:57:36.507205000 AUS Eastern Standard Time 5538 123636 202.166.84.165 31.192.109.167 HTTP 1196 80 229 64240 0 1 1 0 0 128 POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000997000 0.009973000 URI = /snapbn/gate.php dst_ip = 31.192.109.167 dst_port = 80 info = POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) protocol = HTTP src_ip = 202.166.84.165 src_port = 1196
6/19/20 12:57:15.513 AM	Jun 19, 2020 00:57:15.513719000 AUS Eastern Standard Time 5295 110654 202.166.84.165 31.192.109.167 HTTP 1107 80 229 64240 0 1 1 0 0 128 POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000991000 0.008005000 URI = /snapbn/gate.php dst_ip = 31.192.109.167 dst_port = 80 info = POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) protocol = HTTP src_ip = 202.166.84.165 src_port = 1107
6/19/20 12:57:02.586 AM	Jun 19, 2020 00:57:02.586258000 AUS Eastern Standard Time 5010 101310 202.166.84.165 31.192.109.167 HTTP 4626 80 229 64240 0 1 1 0 0 128 POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000998000 0.006018000 URI = /snapbn/gate.php dst_ip = 31.192.109.167 dst_port = 80 info = POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) protocol = HTTP src_ip = 202.166.84.165 src_port = 4626
6/19/20 12:56:56.855 AM	Jun 19, 2020 00:56:56.855009000 AUS Eastern Standard Time 4773 96116 202.166.84.165 31.192.109.167 HTTP 3062 80 229 64240 0 1 1 0 0 128 POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000991000 0.009003000 URI = /snapbn/gate.php dst_ip = 31.192.109.167 dst_port = 80 info = POST /snapbn/gate.php HTTP/1.0 (application/x-www-form-urlencoded) protocol = HTTP src_ip = 202.166.84.165 src_port = 3062

There are two unique URIs obtained. Which indicate the resources compromised by the attacks.

URI
/snapbn/gate.php
/snapbn/ip.php

To get the start and end time of the attacks, the below query is used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = http dst_ip = 31.192.109.167 | sort _time | head 1 | table _time | append
    [ search source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = http dst_ip = 31.192.109.167 | sort _time | tail 1 | table
    _time ]
```

The output of the above is :

_time
2020-06-19 00:55:21.316
2020-06-19 00:58:00.085

Attack Narrative

Start Time	End Time	Attacker(s)	Victim(s)	Type of attack
2020-06-19 00:55:21.316	2020-06-19 00:58:00.085	31.192.109.167	202.166.84.165	Botnet

Countermeasures

After we know the IP address of the C2, it is easy to monitor the traffic coming from it.
The most simplistic approach would be, blocking any traffic coming from the C2 server. We can also redirect the traffic the malicious traffic to sinkholes. The sinkholes log malicious traffic, analyze it, and drop it afterwards such that it cannot reach the original destination it is intended for.
Another way is, if we know the C2 server location, and the botnet has a centralized structure, we can take down the C2 server.
The most viable countermeasure against botnets is probably to clean all infected machines and eliminate the bots installed.

2. SPAM

Spam is any type of unwanted, unsolicited digital communication, often an email or a text message, that gets sent out in large volumes. When hackers cannot embezzle data bandwidth from Internet Service Providers, they try to steal it from individual users, hacking computers and enslaving them in a zombie botnet. Software providers and developers provide resources creating email applications that attempt to filter out most the spam out.
Spam can be used to spread computer viruses, trojan horses or other malicious software. The main motive behind the attack can be identity theft. Some spam attacks try to exploit on human greed, while some attempt to take benefit of the victims' naivety with computer technology to trap them.

SMTP

The core email protocols do not have any mechanism for authentication, making it common for spam attacks. Simple Mail Transfer Protocol (SMTP) is a connection-oriented, text-based protocol using which a mail sender communicates with a mail receiver by issuing command strings. An SMTP session comprises of various commands :

1. MAIL command, to establish the return address.
2. **RCPT** command, to establish a recipient of the message. This command can be issued multiple times, one for each recipient. These addresses are also part of the envelope.
3. DATA to signal the beginning of the message text; the content of the message, as opposed to its envelope. It consists of a message header and a message body separated by an empty line.

Methodology and pattern discovery

Based on our research, search was performed on the data set using the RCPT keyword, and the botnet infected machine (IP Address 202.166.84.165). 214 unique events were obtained, each linked to a unique email address. These 214 events were associated to 5 IP addresses. The attacks were ranging from Jun 19, 2020 00:55:23.278082000 to Jun 19, 2020 01:01:12.900452000.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" src_ip="202.166.84.165" RCPT
```

The above search query was used to obtain the 214 events.

To identify the targeted email address, the below command was used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" RCPT | rex "(?<email_id>\w+\@\w+\.\w+)" | table src_ip, dst_ip, email_id
```

15 of the 214 results obtained can be seen below.

src_ip	dst_ip	email_id
202.166.84.165	205.188.186.137	dbauer3671@sbcglobal.net
202.166.84.165	205.188.186.137	d_fellingham1@tiscali.co
202.166.84.165	205.188.186.137	jkkdcgillham@yahoo.com
202.166.84.165	205.188.186.137	renchtrnr@yahoo.com
202.166.84.165	205.188.186.137	pmergenthal@aol.com
202.166.84.165	205.188.186.137	meredithejohnson@comcast.net
202.166.84.165	205.188.186.137	ahm203040@yahoo.com
202.166.84.165	205.188.186.137	uduncan@deltagastro.net
202.166.84.165	205.188.186.137	rgroth@permatite.com
202.166.84.165	205.188.186.137	tmiralem@gmail.com
202.166.84.165	64.12.168.40	wayne651@live.com
202.166.84.165	64.12.168.40	ag@aligureli.com
202.166.84.165	64.12.168.40	khadidja_kadri@yahoo.fr
202.166.84.165	64.12.168.40	sh_sokolovsky@yahoo.com
202.166.84.165	64.12.168.40	robertsmit@lantic.net

To get the start and end time of the attack along with the first and last email addresses attacked, we used the below mentioned query:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" RCPT | rex "(?<email_id>\w+\@\w+\.\w+)" | sort _time | head 1 | table _time, email_id
    | append [search source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" RCPT | rex "(?<email_id>\w+\@\w+\.\w+)" | sort _time | tail 1 |
    table _time, email_id]
```

We got the following results:

_time	email_id
2020-06-19 00:55:23.278	nickandsonia@comcast.net
2020-06-19 01:01:12.900	jberman1@gmail.com

To get the IP addresses associated with these events, the following command was used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" RCPT | dedup dst_ip | table dst_ip
```

Attack Narrative

Start Time	End Time	Attacker(s)	Victim(s)	Type of attack
Jun 19, 2020 00:55:23.278082000	Jun 19, 2020 01:01:12.900452000	202.166.84.165	64.12.168.40 98.136.185.95	SPAM

			64.12.175.136 205.188.186.137 205.188.186.167	

Countermeasures

As this attack is an extension of the botnet attack, the counter measures against botnets would help in countering the SPAM attacks.
Viable countermeasures for SPAM attacks can be, SPAM detection using machine learning, using graph based learning to detect SPAMS, using natural language processing techniques, or just using real time systems and online tools like "*PhishTank*".
Blocking communication from SMPT protocol can also mitigate this issue.

3. ClickFraud

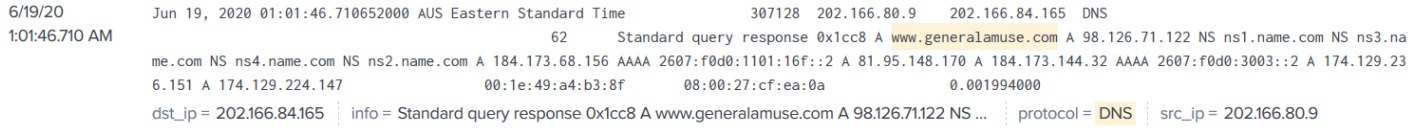
Click fraud is when a person or a machine imitates to be a genuine visitor on a webpage and clicks on an ad, a button, or some other type of hyperlink. The goal of click fraud is to trick a platform or service into thinking real users are interacting with a webpage, ad, or app.
Click fraud usually occurs on a large scale – each link is clicked many times, not just once, and usually multiple links are targeted. To automate this process, click fraudsters often use bots that "click" over and over. Bots comprise roughly 50% of all Internet traffic.

Methodology and pattern discovery

To find the number of ClickFraud requests to *www.generalamuse.com* we first need to obtain its IP address.
This can be done by performing search query with DNS protocol, given below:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = dns www.generalamuse.com
```

Output of the above is given below:



Studying the above output, it can be seen that 98.126.71.122 is the IP for *www.generalamuse.com.*
To get the number of ClickFraud request to the IP, we query using the HTTP protocol and destination IP as the IP obtained above. Given below:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" dst_ip = 98.126.71.122 protocol = http | rex "(?<uri>\w+\ \/\w+\.\w+)"
```

We get 38 events sending ClickFraud requests to *www.generalamuse.com.* 5 of the 38 events are given below:

Time	Event
6/19/20 1:01:08.208 AM	Jun 19, 2020 01:01:08.208700000 AUS Eastern Standard Time 11422 280953 202.166.84.165 98.126.71.122 HTTP 2286 80 299 64240 0 1 1 0 0 128 GET /gen.php HTTP/1.1 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000998000 0.009974000 dst_ip = 98.126.71.122 dst_port = 80 info = GET /gen.php HTTP/1.1 protocol = HTTP src_ip = 202.166.84.165 src_port = 2286 uri = /gen.php
6/19/20 1:01:03.616 AM	Jun 19, 2020 01:01:03.616781000 AUS Eastern Standard Time 11263 276638 202.166.84.165 98.126.71.122 HTTP 4845 80 299 64240 0 1 1 0 0 128 GET /gen.php HTTP/1.1 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000997000 0.029919000 dst_ip = 98.126.71.122 dst_port = 80 info = GET /gen.php HTTP/1.1 protocol = HTTP src_ip = 202.166.84.165 src_port = 4845 uri = /gen.php
6/19/20 1:01:03.595 AM	Jun 19, 2020 01:01:03.595838000 AUS Eastern Standard Time 11261 276621 202.166.84.165 98.126.71.122 HTTP 4811 80 311 64240 0 1 1 0 0 128 GET /gen.php HTTP/1.1 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000997000 0.012965000 dst_ip = 98.126.71.122 dst_port = 80 info = GET /gen.php HTTP/1.1 protocol = HTTP src_ip = 202.166.84.165 src_port = 4811 uri = /gen.php
6/19/20 1:00:50.715 AM	Jun 19, 2020 01:00:50.715461000 AUS Eastern Standard Time 10864 264724 202.166.84.165 98.126.71.122 HTTP 4823 80 299 64240 0 1 1 0 0 128 GET /gen.php HTTP/1.1 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000997000 0.020913000 dst_ip = 98.126.71.122 dst_port = 80 info = GET /gen.php HTTP/1.1 protocol = HTTP src_ip = 202.166.84.165 src_port = 4823 uri = /gen.php
6/19/20 1:00:50.371 AM	Jun 19, 2020 01:00:50.371407000 AUS Eastern Standard Time 10842 264418 202.166.84.165 98.126.71.122 HTTP 4152 80 299 64240 0 1 1 0 0 128 GET /gen.php HTTP/1.1 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000993000 0.009962000 dst_ip = 98.126.71.122 dst_port = 80 info = GET /gen.php HTTP/1.1 protocol = HTTP src_ip = 202.166.84.165 src_port = 4152 uri = /gen.php

To find the resources that have been or can be compromised from these attacks, it's important to analyze the URIs

uri
GET /gen.php

To get the start and end time of the attacks, the following query is used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" dst_ip = 98.126.71.122 protocol = http | rex "(?<uri>\/\w+\.\w+)" | sort _time | head 1 | table _time | append [ search source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = http dst_ip = 98.126.71.122 | sort _time | tail 1 | table _time ]
```

Output of the above command is given below:

_time
2020-06-19 00:55:22.906
2020-06-19 01:01:08.208

Attack Narrative

Start Time	End Time	Attacker(s)	Victim(s)	Type of attack
2020-06-19 00:55:22.906	2020-06-19 01:01:08.208	202.166.84.165	98.126.71.122	ClickFraud

Countermeasures

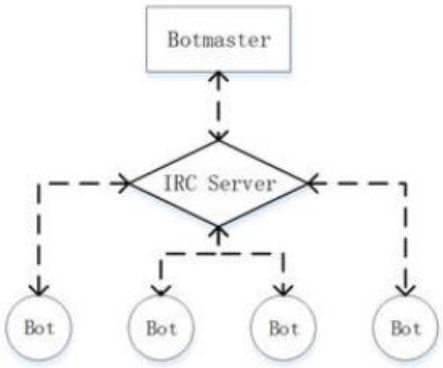
Similarly, as SPAM attack, we can mitigate this attack by countering the botnet attack,
Techniques like Detecting anomalous behaviors in server log patterns can help to mitigate ClickFraud attacks. Also, to prevent events created by bot systems for click fraud attacks, the simplest defense method is to efficiently differentiate such events from human-generated events and clean out them.

4. IRC

IRC (Internet Relay Chat) is a protocol for real-time text messaging among computers that are connected through the internet. It was created in 1988. It is majorly used for group conversations in chat rooms called "channels" although it also allows private messages between two users, transferring of files, and various other server-side and client-side instructions.

As IRC connections can be unencrypted and usually span lengthy time periods, they are an appealing target for DoS/DDoS attackers and hackers. Because of this, a careful security strategy is essential to ensure that an IRC network is not vulnerable to an attack.

These days, IRC is a common technique utilized by botnet owners to send commands to the different computers in their **botnet**. This is accomplished either on a particular channel, on a open IRC network, or on a separate IRC server. The IRC server containing the channel(s) that are used to control bots is referred to as a "**command and control**" or **C2** server.



Through the examining of activity cycle of the Botnet (such as ports and messages), the trends can be clearly identified, and these data flow can be simply cleaned out in the network defence. This type of Botnet has a little effect due of its small scale. However, because of its straightforward operating method and robust operability, it is deeply used by cyber-attackers.

Methodology and pattern discovery

To identify the IRC servers in our server logs, search using protocol was done. The same can also be achieved using the destination port number 6667, as IRC utilizes port 6667.

To find the IRC servers which might be part of a botnet attack, search using the infected machine as the source IP was done, and IRC protocol.

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = IRC src_ip = 202.166.84.165
```

The above query returned 31 events.

Below we can see 3 of the 31 events.

i	Time	Event
>	6/19/20 1:01:59.756 AM	Jun 19, 2020 01:01:59.756180000 AUS Eastern Standard Time 12374 317491 202.166.84.165 58.42.247.143 IRC 1317 6667 335 64240 0 1 1 0 0 128 Request (POST) (Accept:) (Accept-Language:) (CB2:) (Accept-Encoding:) (User-Agent:) (Host:) (Connection:) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000997000 0.019947000 dst_ip = 58.42.247.143 dst_port = 6667 info = Request (POST) (Accept:) (Accept-Language:) (CB2:) (Accept-Encoding:) (User-A... protocol = IRC src_ip = 202.166.84.165 src_port = 1317
>	6/19/20 1:00:47.803 AM	Jun 19, 2020 01:00:47.803864000 AUS Eastern Standard Time 10704 262122 202.166.84.165 61.150.114.216 IRC 2571 6667 282 64240 0 1 1 0 0 128 Request (POST) (Accept:) (Accept-Language:) (CB2:) (User-Agent:) (Host:) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000997000 0.061833000 dst_ip = 61.150.114.216 dst_port = 6667 info = Request (POST) (Accept:) (Accept-Language:) (CB2:) (User-Agent:) (Host:) protocol = IRC src_ip = 202.166.84.165 src_port = 2571
>	6/19/20 1:00:41.386 AM	Jun 19, 2020 01:00:41.386773000 AUS Eastern Standard Time 10359 256048 202.166.84.165 61.17.216.86 IRC 4507 6667 322 64240 0 1 1 0 0 128 Request (POST) (Accept-Language:) (CB2:) (Accept-Encoding:) (User-Agent:) (Host:) (Connection:) 08:00:27:cf:ea:0a 00:1e:49:a4:b3:8f 0.000998000 0.014960000 dst_ip = 61.17.216.86 dst_port = 6667 info = Request (POST) (Accept-Language:) (CB2:) (Accept-Encoding:) (User-Agent:) (Ho... protocol = IRC src_ip = 202.166.84.165 src_port = 4507

The POST request method requests that a web server accepts the data attached in the body of the request message. It is usually utilized when uploading a file.

To find out the unique IP address which received this request, the below query was used:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = IRC src_ip = 202.166.84.165 | dedup dst_ip | table src_ip, dst_ip
```

We received 17 unique IP addresses.

src_ip	dst_ip
202.166.84.165	58.42.247.143
202.166.84.165	61.150.114.216
202.166.84.165	61.17.216.86
202.166.84.165	60.173.109.42
202.166.84.165	221.207.141.60
202.166.84.165	202.112.126.218
202.166.84.165	61.17.216.92
202.166.84.165	61.17.216.94
202.166.84.165	61.167.116.133
202.166.84.165	61.17.216.4
202.166.84.165	184.106.213.57
202.166.84.165	218.189.208.34
202.166.84.165	217.34.4.225
202.166.84.165	88.250.200.14
202.166.84.165	61.177.120.254
202.166.84.165	211.157.110.34
202.166.84.165	200.171.4.222

We used the following command to get the start and end time of the attack:

```
source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = IRC src_ip = 202.166.84.165 | sort _time | head 1 | table _time | append
    [search source="c:\\program files\\splunk\\etc\\apps\\splunkforpcap\\pcapcsv\\traffic-capture-sasm2.pcap (1).csv" protocol = IRC src_ip = 202.166.84.165 |  sort _time | tail 1 | table
    _time]
```

_time
2020-06-19 00:55:21.813
2020-06-19 01:01:59.756

Attack Narrative

Start Time	End Time	Attacker(s)	Victim(s)	Type of attack
Jun 19, 2020 00:55:21.813824000	Jun 19, 2020 01:01:59.756180000	202.166.84.165	58.42.247.143 61.150.114.216 61.17.216.86 60.173.109.42 221.207.141.60 202.112.126.218 61.17.216.92 61.17.216.94	Botnet Attack (Via IRC server)

			61.167.116.133 61.17.216.4 184.106.213.57 218.189.208.34 217.34.4.225 88.250.200.14 61.177.120.254 211.157.110.34 200.171.4.222	

Countermeasures

Mitigating the botnet attack, should be the first point of contact to stop this attack.
Disrupting or disabling the botnet server, which is sending IRC requests, or block any traffic coming from that server.

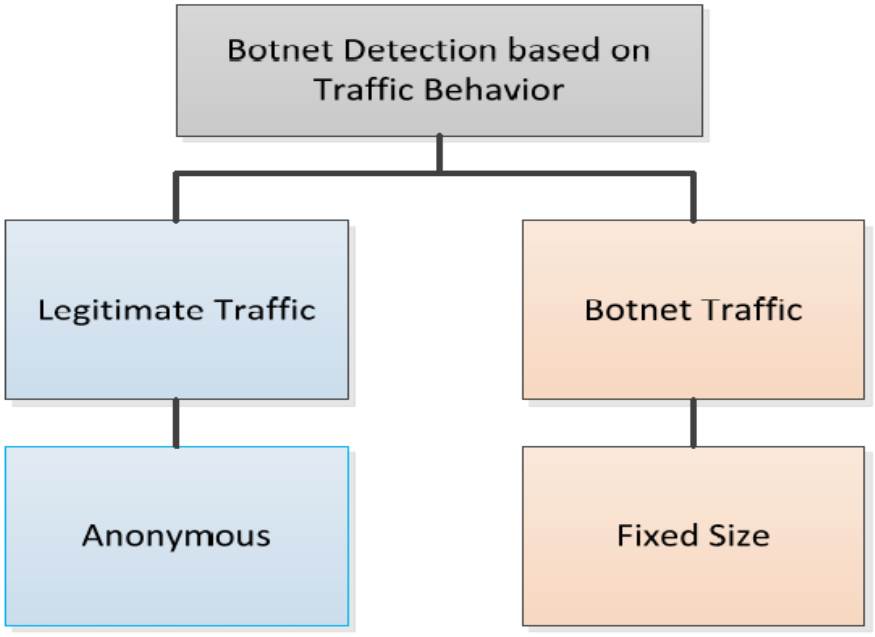
After performing analysis on each of the above attacks, it can be seen that the Victim from the Botnet attack, i.e. 202.166.84.165 is the attacker of the SPAM, ClickFraud, and the IRC attack to the website. This can also be confirmed by the timelines for each of the attacks. This also confirms the fact that, after getting infected, i.e. becoming a bot in the botnet, the machine 202.166.84.165 starts performing illegal operations like ClickFraud, IRC, etc.

Detection and Consequences

After performing searches on Splunk and doing further analysis, it can be concluded that this(all 4 attacks) is an extensive botnet attack, hosted by a C2 server.

DETECTION

In the botnet discovery method, there are a number of ways of determining it, example, based on DNS, Signature, Anomaly, and Machine Learning. The below figure represents botnet recognition based on the type of traffic or can be said as a way of recognition based on anomaly traffic. The volume of data communicated in the transmission between bots and botmasters tends to be fixed, in contrast to the size of the data transmitted normally within the network.



CONSEQUENCES

Botnets has both direct and indirect impacts on users. The most obvious impact is that an infected machine is no longer under the user's control. Most people today store extremely sensitive data (such as financial or legal details) on their personal devices, and such information can be highly vulnerable once the machine is infected.
If the device belongs to a company or government organization, losing control of it can put critical business functions or social services at risk. Evaluating this using the CIA triad, the *Confidentiality and Integrity* is impacted.
More indirectly, botnets can be used by their controllers to carry out other harmful actions, such as:

- Launching Distributed Denial of Service (DDoS) attacks on rival websites or services
  Availability of the service is impacted
- Distributing spam emails or malware
  Integrity of the service is impacted
- Malicious code distribution
  Confidentiality and Availability of the service is impacted.
- ClickFraud
  Integrity of the service is impacted

Conclusion

Increasing number of Internet users and its commercial character naturally bring in proportionate number of criminal minded people to the scene who pose potential threats to legitimate users, Internet infrastructure and timeliness of services offered by it. The aim of this report is to detect and document Internet threats so that general understanding about the malicious users and the malware is increased.
The main focus of this paper is Botnet which surrounds all other attacks in one way or the other.
Knowing C2 addresses can help protect against malicious activities and coordinated attacks. With that information, companies can block access, create alerts on their systems or investigate communications between C2s and samples.

Bibliography and Citations

- www.wikipedia.com
- Study of Botnets and their threats to Internet Security, J Qadri, 2009
- Breaking the Cyber Attack Lifecycle, Palo Alto Networks: Reinventing Enterprise Operations and Defense, March 2015
- A Survey Paper on Botnet Attacks and Defenses in Software Defined Networking, Rahmadani Hadianto and Tito Waluyo Purboyo
- Towards Evaluating Web Spam Threats and Countermeasures, Lina A. Abuwardih, 2018
- Command & Control - Understanding, Denying and Detecting, University of Birmingham, Joseph Gardiner, Marco Cova, Shishir Nagaraja, February 2014
- Proactive Botnet Countermeasures An Offensive Approach, Felix LEDER, Tillmann WERNER, and Peter MARTINI Institute of Computer Science IV, University of Bonn, Germany
- Botnets – The Silent Threat, David Barroso, November 2007