COMP90073-Security Analytics Assignment 1 Report

Detecting cyberattacks in network traffic data

Daniel Gil <Student Id: 905923>

Contents

1 Introduction

Individuals and organizations all around the world are relaying on network related technologies like cloud services to sustain their operations. It is an increased concern on how to protect these networks and data privacy from all kind of possible attacks and anomalies. Anomalies generated as a consequence of cyberattacks is considerably difficult due to the nature of the traffic, the volume, dimensionality and noisy of the data. The objective of the project is to perform analytics using Splunk to explore network traffic data from two days, one day have data recorded during a cyberattack and the other day has attack free network data.

This report is organized as follows: In Section: 2, it is presented an overview of the data including the main characteristics and insights obtained using general splunk analytics to discover suspicious patterns. Next, the section 3 will show a summary of attacks identified in the analysis including its narrative and methodology to find the evidence. From there, in section 4 it is shown the use of splunk analytics to extract features from network traffic data with the goal to model anomaly detection. The next section, 5, presents a case study using clustering algorithms to identify attacks and evaluates future improvements or limitations. Finally, conclusions are described in the section 5.2.1 to evaluate the steps performed and inaccuracies that could be made.

2 Exploratory Data Analysis

2.1 Dataset

The dataset denote network traffic from two days. In sources files, Day 1 represents the Attack Day whereas Day 2 is for the Normal Day. It is important to note that, when queries are referenced and results present both days, only one day query is described, to replicate the query for the other day it is necessary just to change the source to include the correct file.

```
index=main
| stats count as num_packets earliest(_time) AS starttime, latest(_time) as endtime  by source
| eval start=strftime(starttime,"%Y-%m-%d %H:%M:%S.%Q")
| eval finish=strftime(endtime,"%Y-%m-%d %H:%M:%S.%Q")
| eval duration=endtime-starttime
| sort -time
| table source, start, finish, duration, num_packets
```

Listing 1: Ports with more traffic

The import process was made automatically by splunk and some mistakes were considered for further queries. E.g. for day1_5 splunk loaded events with _time in 2019, then a filter by year is included in queries.

Since datasets present different behavior it is convenient to understand each one of them and compare against each other to identify abnormal behavior. Splunk has a dashboard that gives a summary of important data in dataset according to packets and flow.

2.2 Top metrics

Using splunk queries and dashboard a basic analysis can be performed based on the traffic by number of packets flowing between different hosts and size of the payload exchanged. In figure 2, during attack day TCP Protocol accounts for the majority of the traffic, followed by commonly used protocols like HTTP and DNS. In contrast, during normal day protocol HTTP has a different proportion being one of the less used. Some hosts appear to be predominant sharing packets, during normal day, conversations are led by 13.107.4.50, 192.168.10.15 and 192.168.10.14

Table 1: List of datasets

source	start	finish	duration	num_packets
day1_1.pcap.csv	2017-07-05 11:51:48.682	2017-07-05 11:55:10.762	202.079718	800000
day1_10.pcap.csv	2017-07-05 13:57:02.746	2017-07-05 14:03:48.149	405.403926	800000
day1_11.pcap.csv	2017-07-05 14:03:48.150	2017-07-05 14:47:02.265	2594.115381	800000
day1_12.pcap.csv	2017-07-05 14:47:02.265	2017-07-05 14:59:59.975	777.709571	187540
day1_13.pcap.csv	2017-07-05 11:42:42.084	2017-07-05 11:48:40.753	358.668639	800000
day1_14.pcap.csv	2017-07-05 11:48:40.753	2017-07-05 11:51:48.682	187.929402	800000
day1_2.pcap.csv	2017-07-05 11:55:10.762	2017-07-05 11:58:24.650	193.888132	800000
day1_3.pcap.csv	2017-07-05 11:58:24.650	2017-07-05 12:01:44.565	199.915017	800000
day1_4.pcap.csv	2017-07-05 12:01:44.565	2017-07-05 12:04:56.987	192.421521	800000
day1_5.pcap.csv	2017-07-05 12:04:56.987	2019-09-07 01:14:00.000	68562543.012745	800001
day1_6.pcap.csv	2017-07-05 12:16:55.590	2017-07-05 13:15:15.645	3500.055037	800000
day1_7.pcap.csv	2017-07-05 13:15:15.672	2017-07-05 13:45:54.559	1838.887838	800000
day1_8.pcap.csv	2017-07-05 13:45:54.559	2017-07-05 13:51:40.480	345.920627	800000
day1_9.pcap.csv	2017-07-05 13:51:40.491	2017-07-05 13:57:02.745	322.253761	800000
day2_1.pcap.csv	2017-07-03 12:28:24.847	2017-07-03 12:34:53.041	388.194044	800000
day2_10.pcap.csv	2017-07-03 12:19:43.227	2017-07-03 12:28:24.847	521.619748	800000
day2_2.pcap.csv	2017-07-03 12:34:53.041	2017-07-03 12:37:58.680	185.638605	800000
day2_3.pcap.csv	2017-07-03 12:37:58.680	2017-07-03 12:40:57.442	178.762204	800000
day2_4.pcap.csv	2017-07-03 12:40:57.442	2017-07-03 12:49:26.095	508.653073	800000
day2_5.pcap.csv	2017-07-03 12:49:26.095	2017-07-03 13:18:37.636	1751.540234	800000
day2_6.pcap.csv	2017-07-03 13:18:37.636	2017-07-03 14:07:37.609	2939.973629	800000
day2_7.pcap.csv	2017-07-03 14:07:37.609	2017-07-03 14:52:23.189	2685.579320	800003
day2_8.pcap.csv	2017-07-03 11:55:58.598	2017-07-03 12:16:41.882	1243.284236	800000
day2_9.pcap.csv	2017-07-03 12:16:41.882	2017-07-03 12:19:43.227	181.344768	800000

while the attack day is being dominated by traffic between 13.107.4.50, 192.168.10.15 and 172.16.0.1.

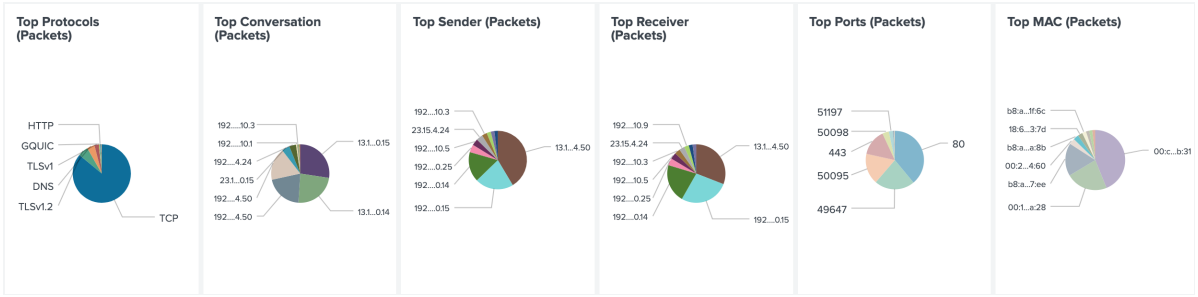

Figure 1: Top Protocols (packets) in Normal Day

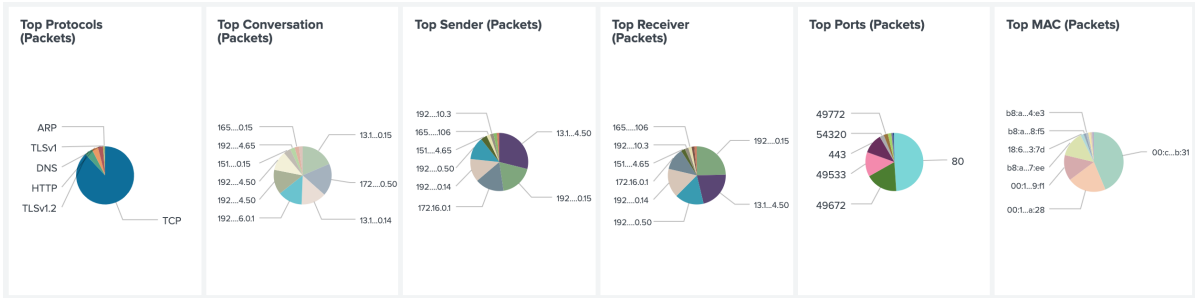

Figure 2: Top Protocols (packets) during Attack Day

Further analysis comparing the time frame it can be seen that host 172.16.0.1 was not even present in the Normal Day suggesting more investigation on these conversations, senders and receivers.

Network traffic has also been increased during the attack day for port 80, from 32% to 38% percent of the flow. This is aligned with previous observation that there is more relevance in protocol HTTP during the attack day. According to the data flow shown in 3 and 4, during the attack day we can see a different pattern in TCP and HTTP traffic from 1.40pm and 2:20pm.

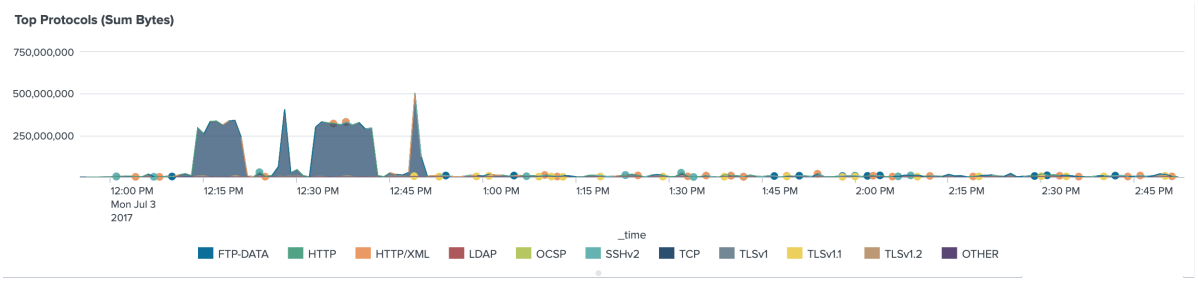

Figure 3: Top Protocols (bytes) in Normal Day

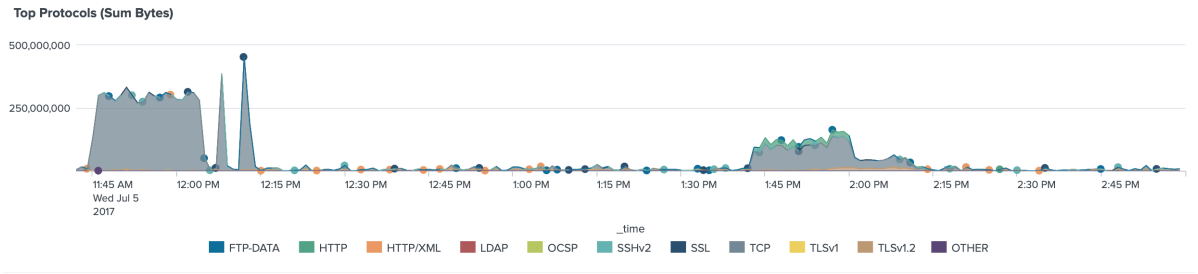

Figure 4: Top Protocols (bytes) during Attack Day

Traffic data shows important relevance for some hosts with significant amount of traffic and payload. In figure 6 and 5 we can see traffic dominated by conversations involving the same hosts 13.107.4.50, 192.168.10.14, 23.15.4.19 and 192.168.10.15, however, during the attack day additional conversation are discovered between new hosts: 192.168.10.50, 151.101.44.65 and 172.16.0.1 suggesting further investigation. In fact, data sent by the host 13.107.4.50 and received by host 172.16.0.1 is considerably bigger than many other hosts as shown in 8 compared to 7

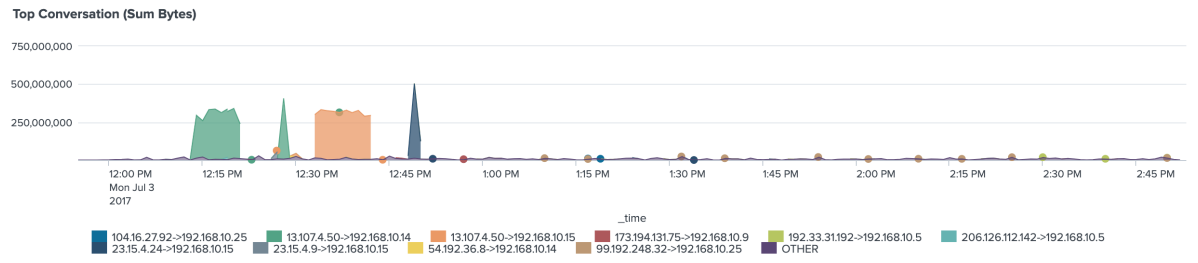

Figure 5: Top Conversations (bytes) in Normal Day

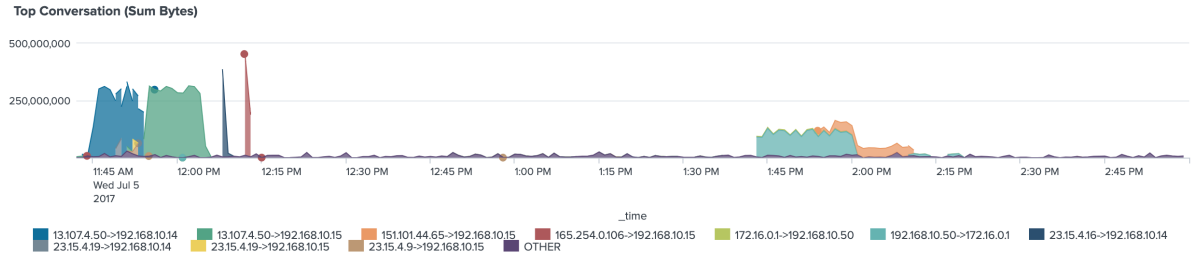

Figure 6: Top Conversations (bytes) during Attack Day

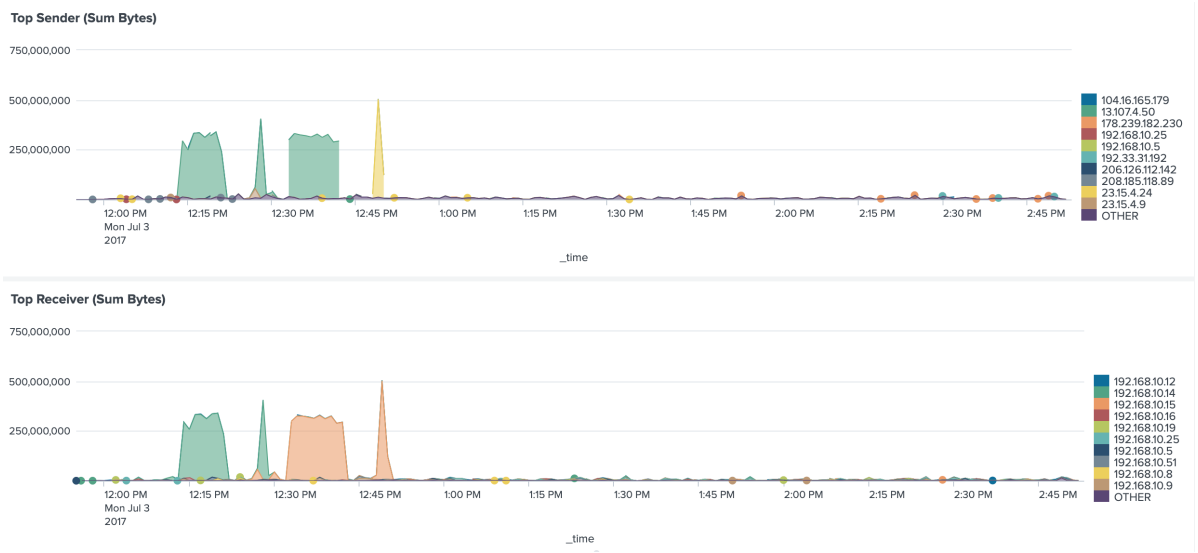

Figure 7: Top traffic (bytes) in Normal Day

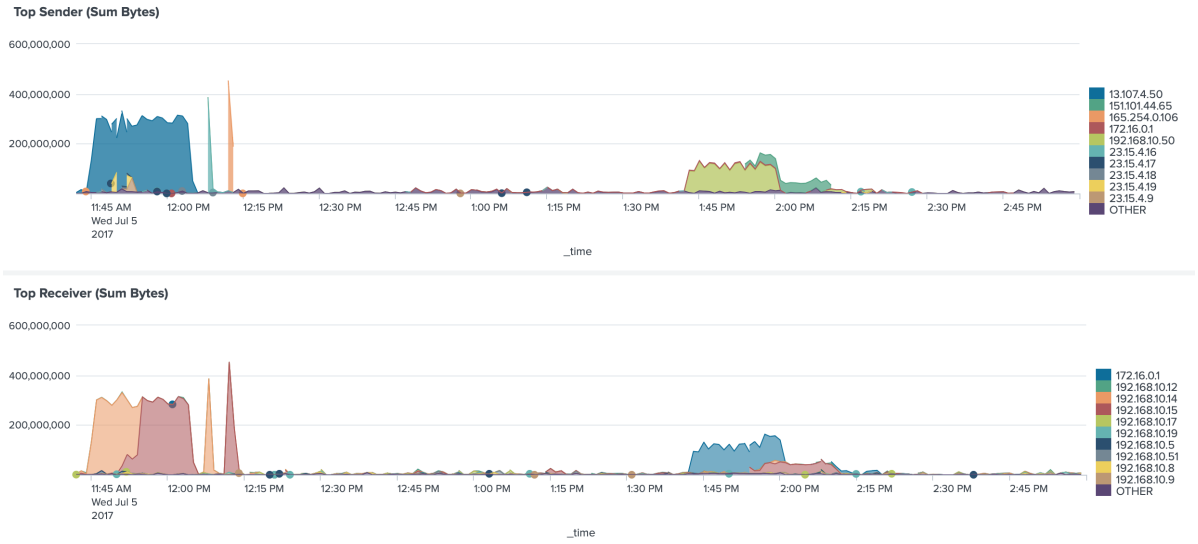

Figure 8: Top traffic (bytes) during Attack Day

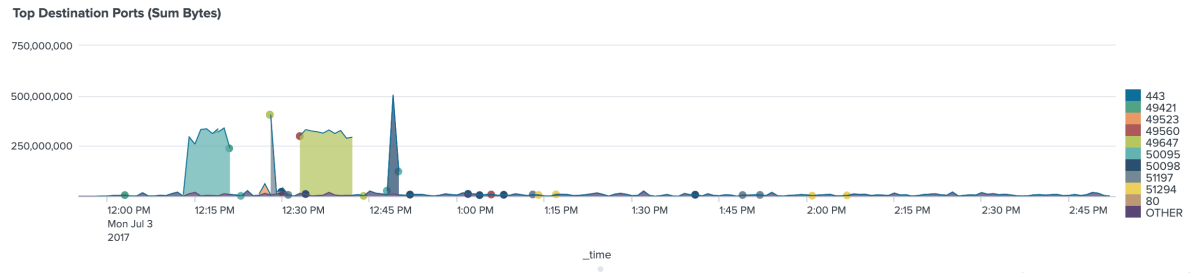

Figure 9: Top Ports (bytes) in Normal Day

During attack day we can see more interaction between new hosts which could be a new application or data transferred for example on the Internet, as suggested in figure 9 in contrast to 10 data exchanged in port 80 is significantly different during the attack day. Note that during the attack day it is shown an increase of data transferred to other ports, however, from figure 10

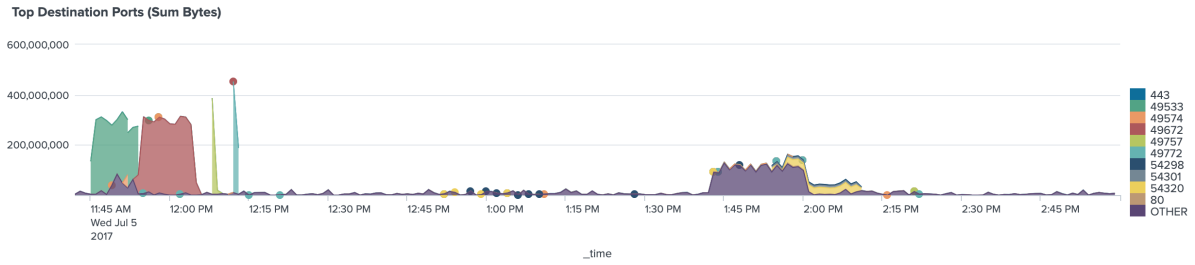

Figure 10: Top Ports (bytes) during Attack Day

cannot be identified the port number, suggesting further investigation as this peak can be caused by interaction or sum of many other ports. A comparison between the two days it is shown in figure 11 using splunk search 2, potential Reconnaissance Attacks or any other ongoing attack is taken during the day.

```
1 index=main date_year=2017 earliest=0
2 | eval attack_day=if(date_wday == "monday", "Normal Day","Attack Day" )
3 | chart count by attack_day,dst_port
4 | sort - count
```

Listing 2: Ports with more traffic

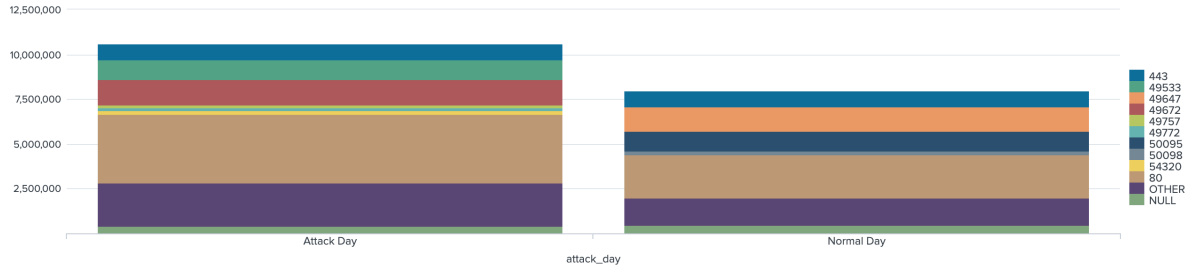

Figure 11: Port traffic comparison

3 Attacks

Table 2 summarizes attacks discovered by analysis.

Attack Summary				
Timestamp	**Type**	**Attacked Services**	**Attacker(s)**	**Victim(s)**
7/5/17 12:01 PM - 2:20 PM	DoS Attack (Volumetric)	Any TCP service	172.16.0.1	192.168.10.51
7/5/17 1:35 PM - 2:00 PM	SYN Flood	Any TCP service	172.16.0.1	192.168.10.51

Table 2: Summary of attacks

The next section will present a detailed description on how the attacks were identified.

3.1 Volumetric Attack

3.1.1 Methodology

From the analysis it was detected an increased traffic in protocols like TCP and HTTP suggesting further investigation. It is of the interest the conversations produced during that time, the ports interacting, protocols and data (bytes) involved in the conversation. 172.16.0.1

was detected as potential attacker due to anomalies in traffic during the attack day as it was part of abnormal conversations and a suspicious IP address. Interactions with other IPs like 192.168.10.50 and 192.168.10.51 were flagged to monitor since they appeared in the conversation. The main method used to detect the attacker was following a potential Cyber Kill Chain

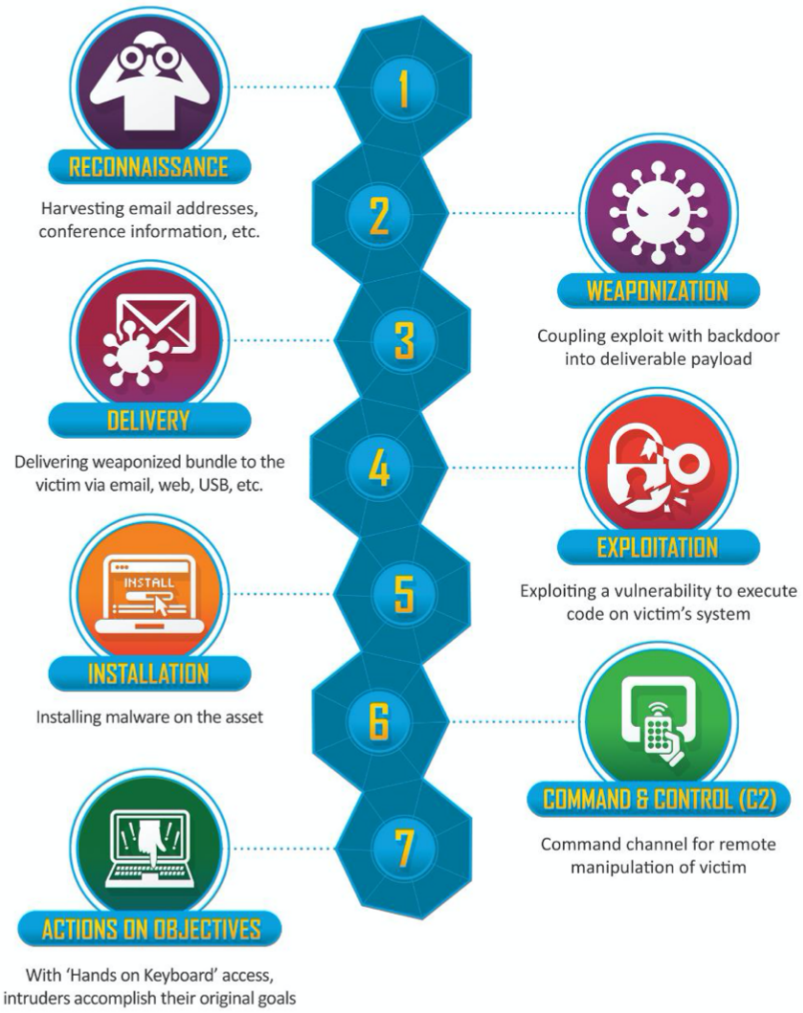

Figure 12: Cyber Kill Chain

shown in figure 12. To confirm a potential attack the sequence of events should follow the type of workflow described in the cyber kill chain.

3.1.1.1 Reconnaissance

Scanning is a way for attackers to discover hosts and ports available on the network to deliver a potential attack. Its often done in preparation of an attack or joining next phase on an attack. Splunk search 3 allows to visualize in figure 13 hosts sending packets to many ports in a particular small time window.

```
1 source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017
2 | timechart span=1m dc(dst_port) by src_ip usenull=f
```

Listing 3: Hosts interacting 1 minute time window with ports

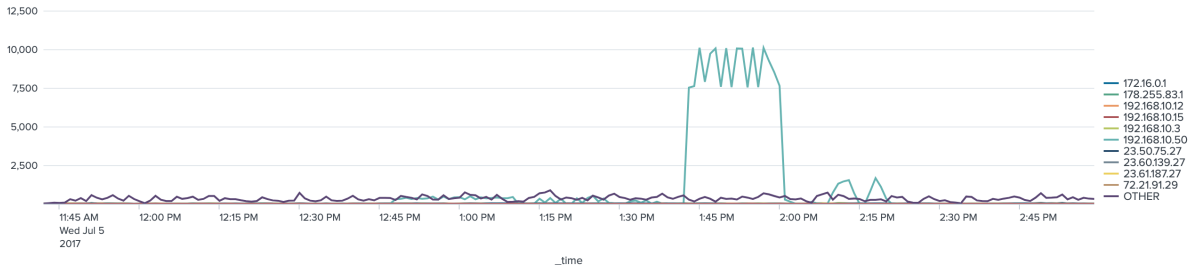

Figure 13: Port Scan

The host 192.168.10.50 represents abnormal behavior and it is worth to investigate more as it is replying to 14116 from requestes made by 172.16.0.1. The splunk search 4 shows in figure 14 not only ports but also different IPs this host is communicating with.

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017
| stats dc(dst_port) as num_dst_port dc(dst_ip) as num_dst_ip by src_ip
| sort - num_dst_port, num_dst_ip limit=10
```

Listing 4: Hosts interacting 1 minute time window with port and IP Address

Figure 14: Port Scan to different IPs

The host 172.16.0.1 appears now confirming initial assumptions as a potential attacker.

3.1.1.2 Weaponization

Next step is to investigate payload, the analysis can show if the attack was executed using a significant payload. using splunk search 5 it is shown in the figure 15 the top hosts sending significant payload using different ports. The host 192.168.10.50 is sending extremely high payload back to 172.16.10.1, in addition, it can be seen a significant traffic to many ports with this host and 192.168.10.51.

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017
| stats dc(dst_port) as dst_port count as packets sum(tcp_length) as total_bytes by src_ip,dst_ip
| sort -dst_port,total_bytes,packets
```

Listing 5: Payload sent by hosts

3.1.1.3 Delivery

Having three hosts in mind, a detailed investigation has to be done regarding the traffic flow in order to determine the delivery process. The figure 16 produced by splunk search 6 shows

src_ip	dst_ip	dst_port	packets	total_bytes
192.168.10.50	172.16.0.1	14116	1056926	1910023352
172.16.0.1	192.168.10.51	1243	20443	21002200
162.213.33.48	192.168.10.51	348	3954	1338134
162.208.20.178	192.168.10.15	303	1676	233476
178.255.83.1	192.168.10.8	157	766	111915
178.172.160.3	192.168.10.15	151	2891	3845305

Figure 15: Payload and traffic

number of packets shared while the figure 17 produced by splunk search 7 shows the total bytes transferred over time.

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017 "172.16.0.1" OR "192.168.10.51" OR "192.168.10.50"
| eval conversation=src_ip+"->"+dst_ip
| timechart span=1m count by conversation
```

Listing 6: Interaction traffic

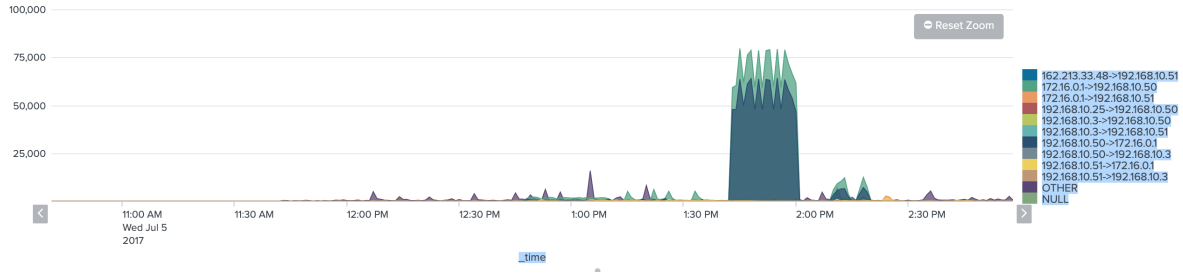

Figure 16: Interactions over time (packets) 172.16.0.1, 192.168.10.51 and 192.168.10.50

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017 "172.16.0.1" OR "192.168.10.51" OR "192.168.10.50"
| eval conversation=src_ip+"->"+dst_ip
| timechart span=1m count by conversation
```

Listing 7: Interaction payload

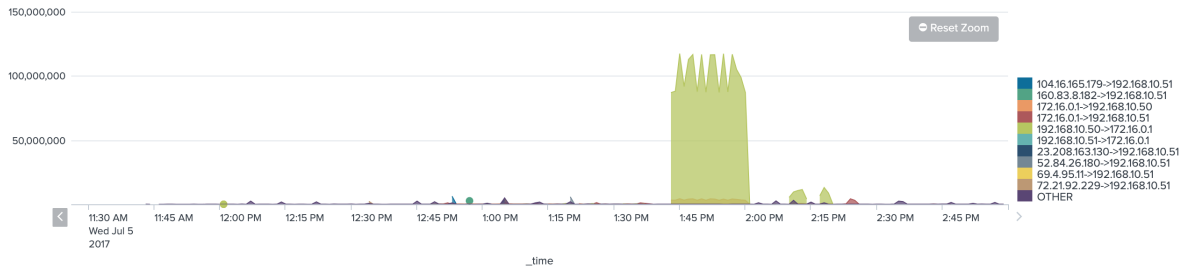

Figure 17: Interactions over time (bytes) 172.16.0.1, 192.168.10.51 and 192.168.10.50

It can be seen traffic and significant payload flowing from 172.16.0.1 to 192.168.10.50.

3.1.1.4 Exploitation

To determine exploitation of a vulnerability, data from normal and attack day can be compared to inspect abnormal behavior. To analyze the interaction it is necessary to observe all possible interactions between the hosts and protocol used. The splunk search 8 shows in figure 18 interactions and protocols used during the attack day while the splunk search 9 shows in figure 19 interactions and protocols used during the normal day. It can be seen the increasing traffic

during the attack day compared to normal day, in addition, the traffic between 192.168.10.50 and 192.168.10.51 seems to be normal as there is no significant deviation from the attack free day.

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv"  date_year=2017
(src_ip="172.16.0.1" OR src_ip="192.168.10.51" OR src_ip="192.168.10.50") AND
(dst_ip="172.16.0.1" OR dst_ip="192.168.10.51" OR dst_ip="192.168.10.50")
| eval conversation=src_ip+"->"+dst_ip
| stats count as num_packets, sum(tcp_length) as total_bytes by conversation,protocol
| sort -num_packets, total_bytes
```

Listing 8: Interaction traffic with protocols - Attack Day

conversation	protocol	num_packets	total_bytes
172.16.0.1->192.168.10.50	TCP	1216516	18379110
192.168.10.50->172.16.0.1	TCP	889296	1542875183
192.168.10.50->172.16.0.1	HTTP	167630	367148169
172.16.0.1->192.168.10.50	HTTP	167108	58546624
192.168.10.51->172.16.0.1	TCP	12166	193924
172.16.0.1->192.168.10.51	TCP	10405	4622242
172.16.0.1->192.168.10.51	TLSv1.2	10038	16379958
192.168.10.51->172.16.0.1	TLSv1.2	5200	2872048
192.168.10.50->192.168.10.51	TCP	1944	134755
192.168.10.51->192.168.10.50	TCP	1922	365330
192.168.10.51->172.16.0.1	TLSv1	1243	635181
192.168.10.50->192.168.10.51	SSHv2	477	129450
192.168.10.51->192.168.10.50	SSHv2	469	53964
192.168.10.50->192.168.10.51	FTP	194	5304
192.168.10.51->192.168.10.50	FTP	145	2126
192.168.10.51->192.168.10.50	FTP-DATA	113	177416
192.168.10.51->192.168.10.50	SSH	38	760
192.168.10.50->192.168.10.51	NBNS	6	
192.168.10.50->192.168.10.51	NBSS	5	20
192.168.10.50->192.168.10.51	SMB	3	373
192.168.10.51->192.168.10.50	SMB	3	382
192.168.10.51->192.168.10.50	NBNS	2	
192.168.10.51->192.168.10.50	NBSS	1	72

Figure 18: Interactions and protocols 172.16.0.1, 192.168.10.51 and 192.168.10.50

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day2_*.pcap.csv" date_year=2017
(src_ip="172.16.0.1" OR src_ip="192.168.10.51" OR src_ip="192.168.10.50") AND
(dst_ip="172.16.0.1" OR dst_ip="192.168.10.51" OR dst_ip="192.168.10.50")
| eval conversation=src_ip+"->"+dst_ip
| stats count as num_packets, sum(tcp_length) as total_bytes by conversation,protocol
| sort -num_packets, total_bytes
```

Listing 9: Interaction traffic with protocols - Normal Day

conversation	protocol	num_packets	total_bytes
192.168.10.50->192.168.10.51	TCP	2018	159740
192.168.10.51->192.168.10.50	TCP	1959	297716
192.168.10.50->192.168.10.51	SSHv2	572	156662
192.168.10.51->192.168.10.50	SSHv2	555	64340
192.168.10.50->192.168.10.51	FTP	144	3950
192.168.10.51->192.168.10.50	FTP	111	1626
192.168.10.51->192.168.10.50	FTP-DATA	82	131386
192.168.10.51->192.168.10.50	SSH	46	920

Figure 19: Interactions and protocols 172.16.0.1, 192.168.10.51 and 192.168.10.50

3.1.1.5 Installation, command and actions on objectives

Apart from the data transferred there is no evidence of any maleware installation as the dataset does not provide any operating system log. The attacker takes control of the victim.

3.1.2 Description and narrative

During a volumetric attack the attacker sends a significant amount of traffic with the goal of saturate the bandwidth and any resources of the victim with the aim of stop its services. In DoS (Denial of Service) and DDoS (Distributed Denial of Service) the attacker typically sends IP packets from a fake source address to disguise himself. A volumetric attack can be initated inside a network of hosts controlled by malware that allow attackers to control the host and generate the necessary traffic. These hosts are usually called bots that can be computers, servers or any other device connected to the network. The popularity of some unsecured IoT devices has increased potential risk in organizations for this kind of attacks. A DDos attack is a coordinated attack over a target with the aim of compromise its resources and produce a negative impact in the availability of its services. DDoS attacks can use significant amount of data transferred per second taking advantage of technology to control infected machines also called 'bots' [1]. These controlled machines can produce high amount of requests in a small time window form different connections [2] making the attack extremely hard to detect and prevent. Attackers usually follow the cycle: infect, coordinate, and attack [3]. Once the attacker takes control infecting other machines with malewares, this machines coordinate themselves to create an attack and establish a target. This bots now can analyze the network and decide when to start the attack based on different patterns launching high amount of requests to overload the target. If the attack is coordinated between different locations it becomes harder to mitigate, if the attack succeeds the victim can be flooded with traffic overloading its capacity and causing a denial of service. In the figure 20 it is shown a typical anatomy of a DDoS attack, usually the attack starts when the attacker takes control of other hosts called bots and deploy the necessary tools to generate large volume of traffic without being identified by its own IP address (IP Spoofing). Following that, the attacker use bots to send streams of attack to the victim.

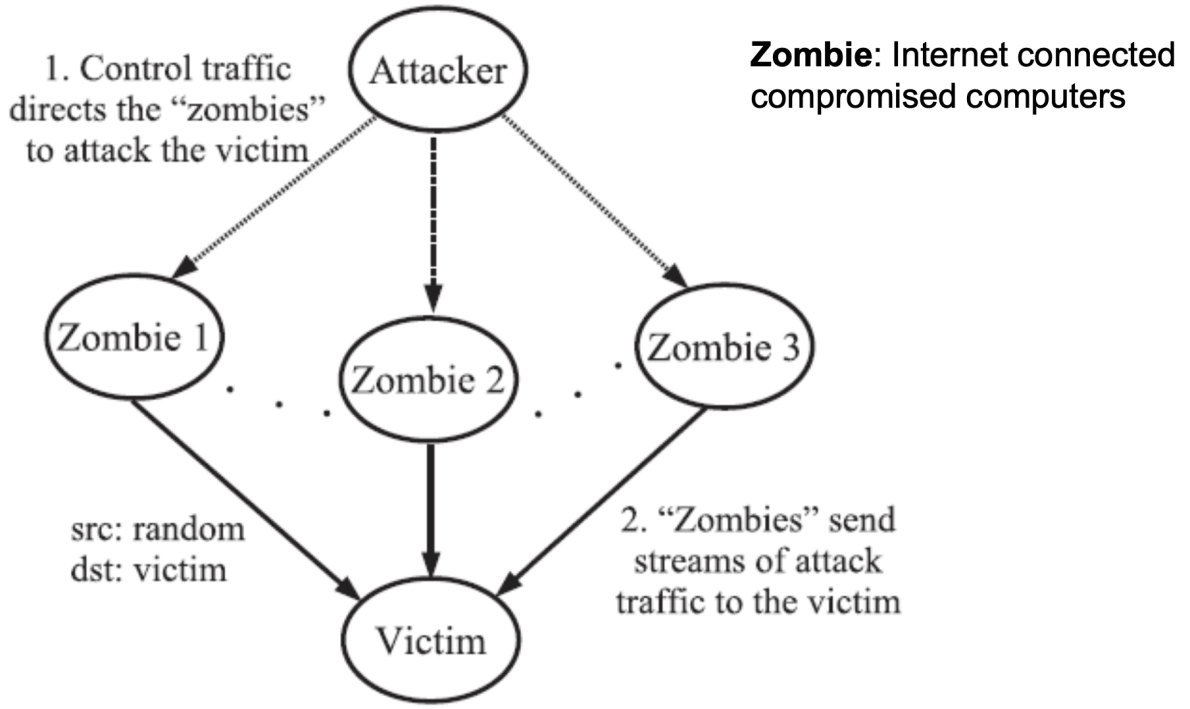

Figure 20: DDoS Distributed Denial of Service

The splunk search 10 summarizes the time line of the attack in the figure 22. The host

172.16.0.1 is probably a botnet or zombie used for the attacker to disguise itself, then, at 12.01pm initiates the attack sending traffic through 14116 ports directed to port 80 in host 192.168.10.50. A detailed traffic interaction is shown in 21, following the kill chain one can assume the attacker prepare the attack and at 1.37pm the kickoff occurred and a high transfer of data is seen in the network, when the attack finishes at 14.20pm, the victim sent around 1821MB of data.

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017 (src_ip="172.16.0.1" OR src_ip="192.168.10.50") AND (dst_ip="1
| eval conversation=src_ip+"->"+dst_ip
| eval mb=tcp_length/1024/1024
| stats dc(src_port) as dc_src_ports dc(dst_port) as dc_dst_ports sum(mb) as total_mb avg(mb) as avg_mb count as total_packets earliest(_time) AS
| eval start=strftime(starttime,"%Y-%m-%d %H:%M:%S.%Q")
| eval finish=strftime(endtime,"%Y-%m-%d %H:%M:%S.%Q")
| eval duration=(endtime-starttime)/60
| eval mbps=total_mb/duration
| sort -_time
| table conversation, duration, start, finish,dc_src_ports,dc_dst_ports,total_mb,avg_mb,mbps,total_packets
```

Listing 10: Interaction traffic with protocols - Normal

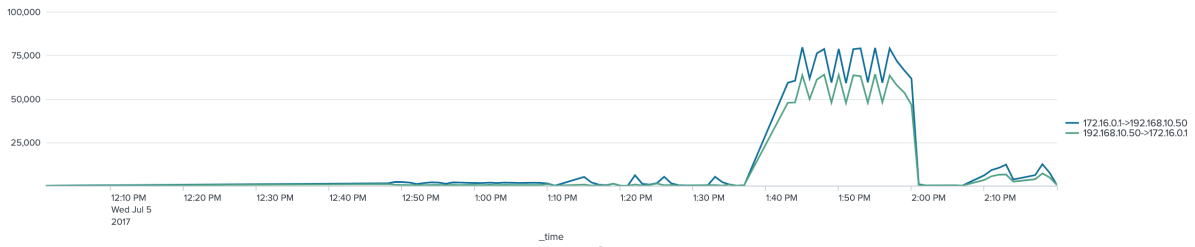

Figure 21: Timeline of the attack

conversation	duration	start	finish	starttime	endtime	dc_src_ports	dc_dst_ports	total_mb	avg_mb	mbps	total_packets
172.16.0.1->192.168.10.50	139.1397045	2017-07-05 12:01:00.533	2017-07-05 14:20:08.915	1499256060.533403	1499264408.915675	14116	1	73.36209678649902	0.0000530217001053024 66	0.5272549416	1383624
192.168.10.50->172.16.0.1	139.1397030	2017-07-05 12:01:00.533	2017-07-05 14:20:08.915	1499256060.533540	1499264408.915718	1	14116	1821.5402145385742	0.0017234321178006542	13.09144820	1056926

Figure 22: Timeline of the attack

3.2 SYN Flood Attack

3.2.1 Methodology

From the analysis it was detected an increased traffic in protocols like TCP and HTTP suggesting further investigation. First, for TCP connections we are interested to analyze the handshakes. According to the three-way handshake, a host will start the connection by sending SYN to the other host. Similarly, if this other host wants to reject the connection it will send an RST, or if it wants to continue then it will send ACK to establish the connection and continue sharing packets. During a SYN flood attack, the attacker send a significant number of SYN to create a connection with the victim, but it will never finish the handshake leaving the victim waiting for ACK flag. Two scenarios are shown in figure 23 and results from splunk queries are shown in 25. The new host 172.16.0.1 during the attack day is initiating most of the conversations with the victim 192.168.10.50

In the description section, the handshake analysis is presented to conclude this as a potential attack as it a significant number of unfinished handshakes.

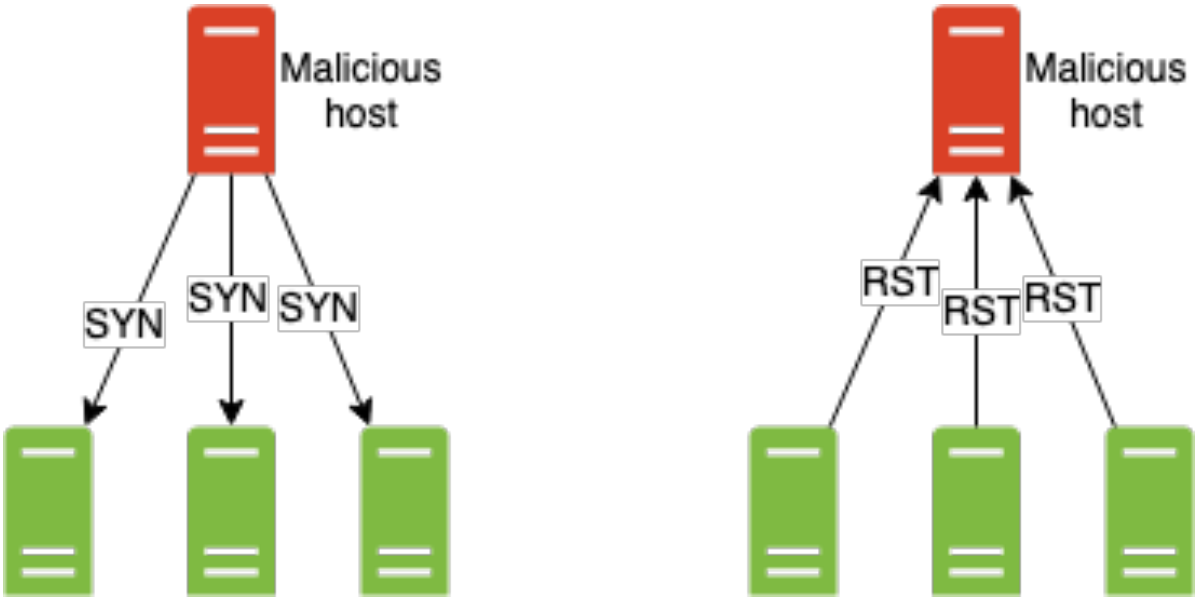

Figure 23: SYN Packets and RST packets traffic during Normal Day

dst_ip	src_ip	count_rst	count_ack	count_syn
192.168.10.50	172.16.0.1	293796	807324	217677
192.168.10.51	172.16.0.1	2362	8043	1243
172.16.0.1	192.168.10.50	2316	886980	172545
192.168.10.25	192.168.10.3	252	1332	240
192.168.10.50	192.168.10.3	200	2387	166
172.217.6.238	192.168.10.16	153	407	13
173.241.242.143	192.168.10.16	100	252	33
192.168.10.15	162.208.20.178	76	1181	303
192.168.10.8	192.168.10.3	76	514	102
192.168.10.17	192.168.10.3	70	369	62

Figure 24: SYN Packets during Attack Day

src_ip	dst_ip	count_syn	count_ack	count_rst
172.16.0.1	192.168.10.50	217677	807324	293796
192.168.10.50	172.16.0.1	172545	886980	2316
172.16.0.1	192.168.10.51	1243	8043	2362
192.168.10.51	172.16.0.1	1243	10918	5
192.168.10.51	162.213.33.48	480	1959	0
162.213.33.48	192.168.10.51	348	2310	0
162.208.20.178	192.168.10.15	303	1181	76
192.168.10.15	162.208.20.178	303	1348	4
192.168.10.3	192.168.10.25	240	1332	252
192.168.10.25	192.168.10.3	240	2136	0

Figure 25: SYN Packets during Normal Day

3.2.2 Description and narrative

A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a victim in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic[1]. Anomalies in handshakes can lead to detection of this attack. During the attack day, an increasing TCP traffic was detected, then if the traffic

[1] https://https://en.wikipedia.org/wiki/SYN_flood

is considered normal then handshakes should be completed successfully. A successful TCP connection should start with a SYN flag and then finish with ACK as shown in figure 26, the SYN flood attack exploits the handshake when the attacker does not respond with ACK after the victim host answer SYN with a SYN-ACK as shown in figure 27.

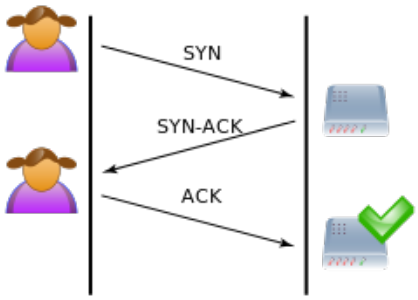

Figure 26: Succesful TCP handshake

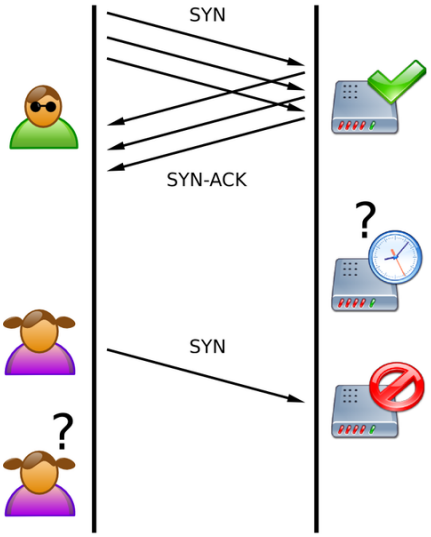

Figure 27: SYN Flood

In the last section we found suspicious SYN and RST flags between two hosts, to make sure there is something occurring anomalous with the handshakes one can analyze the handshakes not finishing successfully. The splunk query 11 generates the results 28 and 29, one can conclude anomalies in handshakes during the attack day leading to a SYN flood.

```
source="/opt/splunk/etc/apps/SplunkForPCAP/PCAPcsv/day1_*.pcap.csv" date_year=2017
| transaction maxspan=5s tcp_stream startswith="[SYN]"
| search "SYN, ACK" AND linecount>=2 AND NOT "[ACK]"
| eval src=mvindex(src_ip,0)
| eval dst=mvindex(src_ip,1)
| stats count earliest(_time) AS starttime, latest(_time) as endtime by src,dst
| eval start=strftime(starttime,"%Y-%m-%dT%H:%M:%S.%Q")
| eval finish=strftime(endtime,"%Y-%m-%dT%H:%M:%S.%Q")
| eval duration=endtime-starttime
| sort -count
| table src,dst,start,finish,duration,count
```

Listing 11: "Half open" traffic

4 Feature extraction

Feature engineering was driven by four categories:

src	dst	start	finish	duration	count
192.168.10.3	192.168.10.50	2017-07-03T11:57:51.913	2017-07-03T14:47:50.622	10198.708737	6
178.172.160.4	192.168.10.12	2017-07-03T12:52:05.500	2017-07-03T12:52:05.530	0.030564	5
172.217.7.2	192.168.10.8	2017-07-03T13:55:04.871	2017-07-03T13:55:04.878	0.00748	4
192.168.10.3	192.168.10.5	2017-07-03T12:08:11.400	2017-07-03T12:08:29.133	17.733126	4
144.2.1.1	192.168.10.12	2017-07-03T12:38:35.364	2017-07-03T12:38:35.379	0.014610	3

Figure 28: "Half-open" traffic during Normal Day

src	dst	start	finish	duration	count
172.16.0.1	192.168.10.50	2017-07-05T13:15:41.254	2017-07-05T14:12:27.844	3406.590335	145
192.168.10.15	72.21.91.70	2017-07-05T12:17:53.785	2017-07-05T12:17:53.827	0.042223	24
178.172.160.2	192.168.10.15	2017-07-05T14:23:17.772	2017-07-05T14:23:17.845	0.073229	6
104.88.21.130	192.168.10.8	2017-07-05T14:13:21.832	2017-07-05T14:13:21.835	0.002274	5
192.168.10.16	192.168.10.3	2017-07-05T13:42:13.317	2017-07-05T14:38:53.404	3400.086624	3

Figure 29: "Half-open" traffic during Attack Day

- **Basic features:** Collected from the packet header incuding protocol types, services, TCP flag, source and destination flags
- **Time based features:** An example could be number of data transferred during a time a window.
- **Connection-based features:** Features that are computed over an historical packet transmission, for example, the number of packets from source to destination.

The table shows the list of features used for anomaly detection.

5 Case study

It is considered an anomaly a data point that presents a deviation from the expected behavior [4]. It can be defined as a single event or a set of data points / events. Anomalies in network traffic data can show complex patterns which can occur because of malicious activities in the network, overloading of networking infrastructure, malfunctioning of network devices or compromises when setting network parameters.

Feature name	**Type**	**Description**
Basic features		
start	String	Time when the connection started
finish	String	Time when the connection finished
src_ip	String	Source IP
dst_ip	String	Destination IP
num_dst_port	Number	Number of ports in destination
total_bytes	Number	Total bytes in the conversation
num_packets	Number	Total number of packets in the conversation
mbps	Number	Number of bytes per second

K-means is an algorithm that can help to cluster data and identify data points or events that are far from the expected behaviour using a measure of distance.

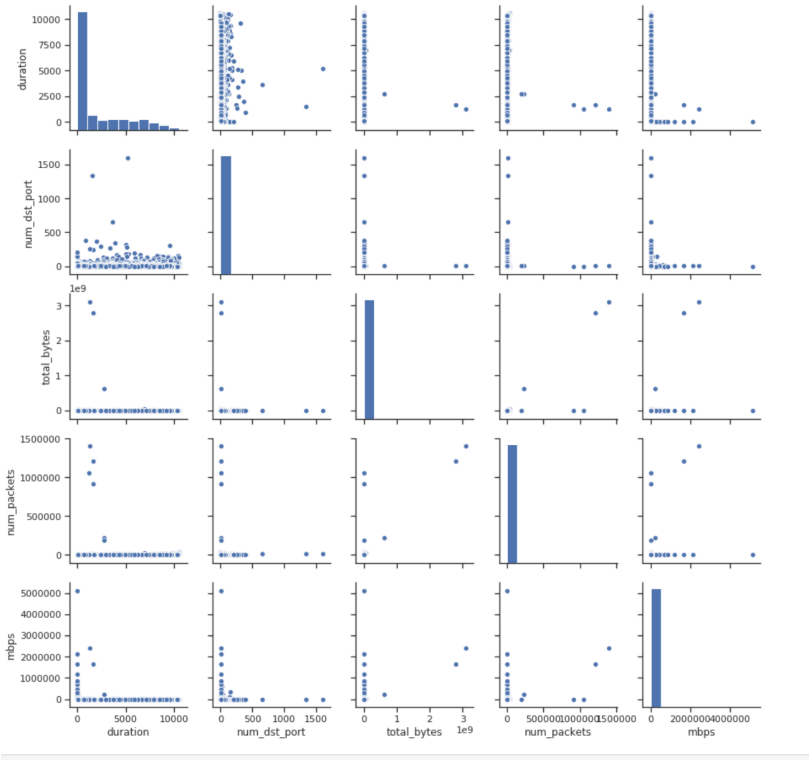

Figure 30: Scatter plot

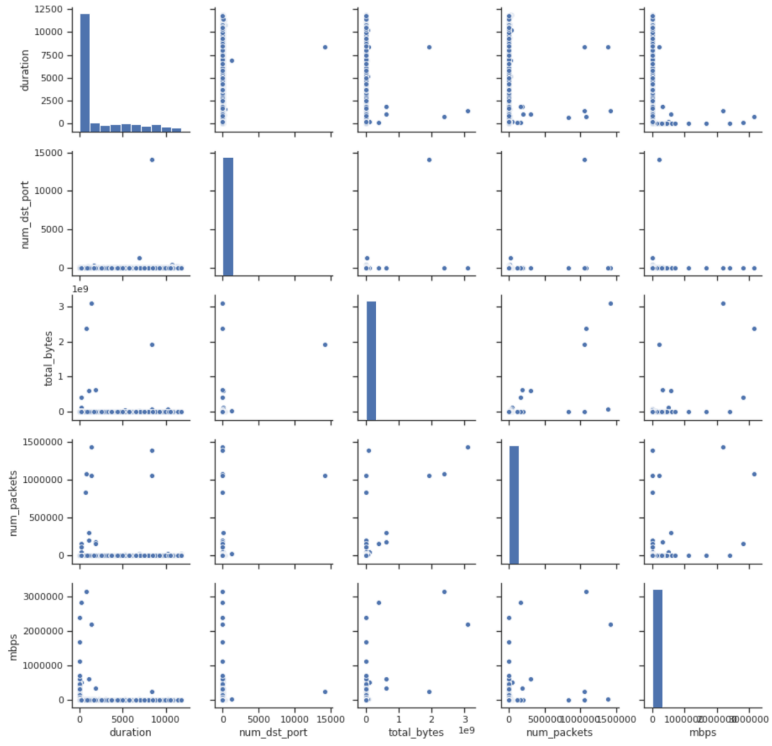

Figure 31: Scatter plot

	PC1	PC2	PC3
duration	-0.000654	0.772492	-0.633709
num_dst_port	0.297597	0.514425	0.656353
total_bytes	0.617046	-0.018323	-0.036169
num_packets	0.585354	0.033046	0.005810
mbps	0.433648	-0.370400	-0.407764

In this scenario, one can model 1-cluster to reflect the expected behaviour as fitted with normal day events and use the attack day to identify outliers from the normal day dataset considering that both attacks detected are related to the same attacker and victim. The figure 30 shows the relation between features, it can be seen as expected behaviour that the total number of destination ports are usually a small number, the total number of bytes is highly correlated with the number of packets.

5.1 Data preparation and modelling

For analysis purpose we can use a dimensionality reduction technique like PCA (Principal Component Analysis) to visualize the cluster analysis in three-dimensional space. The table illustrates the principal components and the weights assigned to each of the features, each principal component is a linear combination of the features.

5.2 Evaluation

The model is fitted with normal day data and tested with the attack day data.

Extreme values are identified for higher values in PC1 and higher values for PC3. This is, if we have traffic flowing between 2 hosts with a significant payload, requests are concentrated in small amount of time, and in addition we have an abnormal number of ports used, then this traffic is being classified as anomaly.

Details of the analysis can be found in the notebook sa.ipynb accessible from http://115.146.92.184:8888/lab (pwd:6260)

5.2.0.1 Additional sources

Although the dataset size is around 19GB it is probably hard to detect DDoS attacks with just few hours of traffic as we can have just some type of outliers allowing wrong conclusions. In addition, computer power is a limitation to develop clustering analysis or any machine learning algorithm with high dimensional data. In this report just few features were selected, but if we can combine this datasets with operating systems logs from selected machines and any other additional data from network appliances, then the analysis could be performed in depth.

5.2.0.2 Feature engineering

A typical feature selection procedure consists of three major steps: a) subset generation, b) subset evaluation and c) validation. This report includes only part a), evaluation and validation is out of the scope. Only basic features were considered, to build a robust model it is necessary to include time-based and connection-based features. Other features that could be relevant according to the attacks detected:

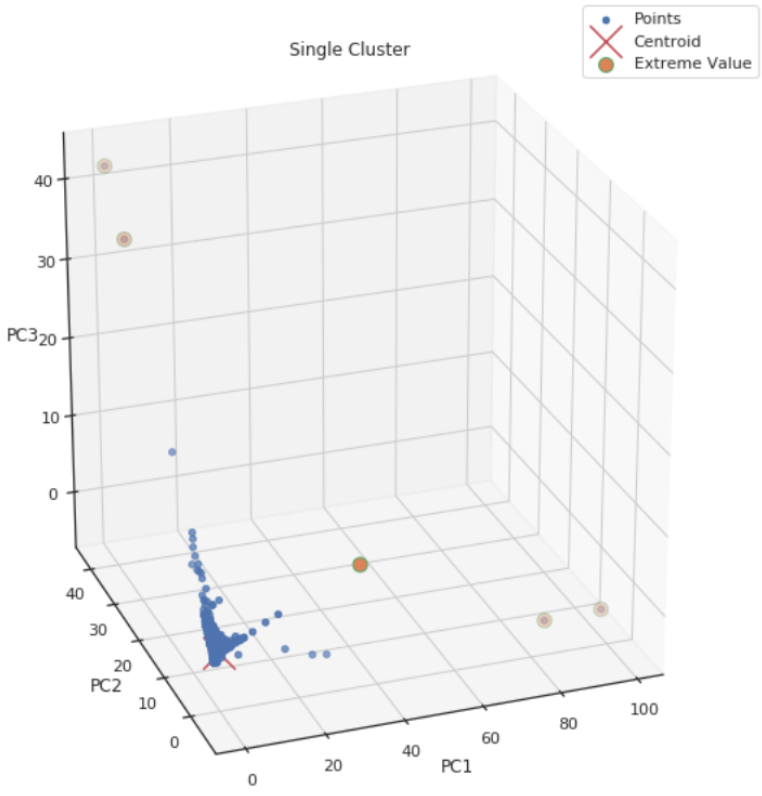

Figure 32: Cluster

- Basic: Protocol and TCP flags
- Connection-based: Number of flows to unique src/destination IPs in the last T seconds from the same source/destination and Number of flows from the source/destination to the same source/destination port in the last T seconds
- Time-based: Number of flows to unique destination IPs in the last N flows from the same source.

5.2.0.3 Limitations in clustering analysis

Malicious activities in a network can be described as point anomaly, contextual anomaly, and collective anomaly. K-means approach is not efficient to find and distinguish this kind of anomalies from other clusters. k-means has a tendency to make outliers a one-element cluster. Then the outliers have the smallest possible distance and will not be detected.

5.2.1 Conclusion

Anomaly detection is a complex task, however, many anomalies can be easily identified executing simple queries and analysing network traffic data. This report provides a good glimpse on how a data exploratory analysis can lead to intrusion detection using basic clustering algorithms. Although a robust and complex model can be built with data provided, a simple basic few features and a distance-based algorithm like k-means can help to identify outliers specially when anomalies represent extreme values that can be validated with the type of attack.

References

[1] M. Nogueira, "Non-parametric early warning signals from volumetric ddos attacks," *CoRR*, vol. abs/1609.09560, 2016. [Online]. Available: http://arxiv.org/abs/1609.09560

[2] P. Farina, E. Cambiaso, G. Papaleo, and M. Aiello, "Are mobile botnets a possible threat? the case of slowbot net," *Comput. Secur.*, vol. 58, no. C, pp. 268–283, May 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2016.02.005

[3] S. Taghavi Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE Communications Surveys amp Tutorials*, vol. 15, pp. 2046 – 2069, 11 2013.

[4] M. Bhuyan, D. K. Bhattacharyya, and J. Kalita, *Network Traffic Anomaly Detection and Prevention: Concepts, Techniques, and Tools*, 01 2017.