



**Abertay
University**

Network Forensics

Jack Gates

Student no: 1500763

CMP416: Digital Forensics 2

BSc Ethical Hacking Year 4

2018

CONTENTS

Introduction	3
Packet Analysis - Ann's Bad AIM	5
Timeline:	7
Packet Analysis - Ann's Rendezvous	8
Timeline:	10
Statistical Flow Analysis - The Curious Mr X.....	11
TimeLine	15
Preventing Further Compromise.....	15
Wireless Networks - HackMe, Inc.	16
Network IDS and Analysis - InterOptic Saves the Planet	18
Event Logs - L0ne Sh4rk's Revenge	20
Conclusion	22
References	23

INTRODUCTION

This document contains exercises involving network forensics. Each exercise includes an investigation tailored to a specific area in network forensics. The aim of these exercises is to give insight into network forensics and provide the ability to conduct a forensic investigation, involving evidence from certain sources of a network, and following the proper procedures defined in the computer misuse act. It also provides experience in planning these investigations within given scenarios and using a range of tools suitable for each task. The ability to critically evaluate and understand the results of the investigation will also be obtained from this practical work. There are five exercises in total, each focussed on a different topic within network forensics, although the first exercise can be broken down into two investigations.

Packet analysis is the first topic which will be covered in this investigation. Packets are all of the small amounts of data being sent over a network, and these are often analysed to understand the flow of packets which can allow events to be reconstructed in cases where network misuse may have taken place. This topic is covered over two investigations, the first being titled "Ann's Bad Aim". This exercise involves the suspected data leak, by an alleged undercover agent named Ann Dercover, of a secret recipe file and the relevant packet capture needs to be analysed in order to confirm or deny this. The second investigation is titled "Ann's Rendezvous" which takes place a while after the first investigation. In this scenario, Ann Dercover has been released on bail, but escaped. Her network activity had been monitored previously, so the packets will be analysed for evidence regarding her location. The two investigations that have been done for these scenarios will be demonstrated, showing how evidence was gathered in order to summarise any theories and a timeline of events will be created for each.

The next topic covered in this document is statistical flow analysis. Flow records are recorded by sensors and involve information regarding data flow, usually including IP addresses and ports of the source and destination, protocols, timestamps, and the size of any transferred data. These records are very useful in forensic cases and can be used in order to identify any compromised hosts, confirm any data leakages, or reveal the activity performed by given addresses over certain time periods. The scenario for this case, titled "The Curious Mr X.", is that a fugitive named Mr. X has infiltrated an Arctic Nuclear Fusion Research Facility (ANFRF), and pivots through the network. Fortunately, there is a Cisco ASA firewall which collects flow records from the internal, the internet, and DMZ subnets, and is also set up with a SPAN port that monitors the internal and DMZ in Argus format. These captured flow records will be analysed so that Mr. X's activity can be examined, so that the investigator can figure out what he found, and determine if there have been any data leaks performed by the attacker. The investigation will be demonstrated using evidence like in the previous topic, and a timeline will be created. Any mitigations that ANFRF staff can perform to prevent any further compromises will also be described.

The third topic is about the forensics involved in wireless networks. Wireless traffic is very common and should also be studied because it poses a different set of challenges for forensic investigators, as it uses different protocols for transferring packets which are normally encrypted, over the data-link layer. The scenario is titled "HackMe, Inc." and for this investigation, a notorious hacker that goes by the name 'InterOptic' is on the run from police and is suspected to be in the local area. Meanwhile in the same area, a system admin named Joe for a company named 'HackMe, Inc.' is discovering issues with his Wireless Access Point (WAP). He has been dropped and can no longer access it. He has contacted a team of investigators looking into the 'InterOptic' case as he thinks these events may be connected. Joe makes it clear that no one should have access to the WAP except himself, and he provides the MAC address of his device. He hands a packet capture of the event to the team of forensic investigators, hoping that his problem can be solved, and that InterOptic can be caught if they are

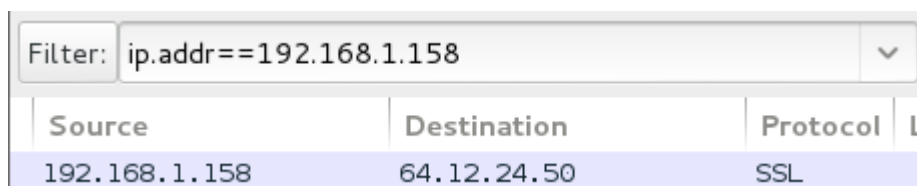
found to be involved. After this investigation is completed, a report will be written evaluating the investigation and demonstrating an understanding of the evidence, and what it means. The report will also summarise how the attack may have occurred.

The topic after this is focussed on Network Intrusion Detection systems and how they are analysed in network forensic investigations. These detection systems are used as a safety measure in networks, for detecting potentially malicious activity on a network and sending alerts. For the scenario titled "InterOptic Saves the Planet", a company called MacDaddy payment Processor has deployed a network intrusion detection system tool called 'snort'. This system records all anomalous activities in the form of alerts for the internal network subnet, the DMZ subnet, and the external network. The system had logged an alert stating that suspicious executable code was sent to the internal network from an external source. MacDaddy have hired an investigator to see if this alert is either true or false, and to further analyse other alerts to see if there may be other suspicious activity going on within the company network. The company have provided the investigator with all of the snort alerts that are relevant to the case, as well as the rules that the 'snort' system follows. The corresponding log file of all TCP communications have also been provided for an investigation into this alert. Once this investigation is finished, a report will be written to evaluate the investigation and demonstrate an understanding of the evidence, what it means, and to summarise the role of the NIDS.

The final topic that this investigation will cover is to do with the role of event logs in forensic investigations. These are useful if detailed packet captures are not available, which is normally the case in smaller networks, as they provide information about a system which could be useful during an investigation, for example, any privileged commands executed could be a useful event to log. For this scenario titled "LOne Sh4rk's Revenge", staff at a company called Bob's dry cleaners noticed that their DMZ server was suddenly hit with a wave of failed authentication attempts over secure shell. The staff believe that they may be under attack by a disgruntled customer known as "LOne Sh4rk". The security team have been collecting logs from the servers, workstations and the firewall to be sent to a server for storage, as they have been attacked previously and wanted to safeguard their network better as they store sensitive customer information, like credit card details. The suspicious failed authentication behaviour was recorded in the authorisation logs, and were handed over to a forensic investigator, along with the workstation, and the firewall logs. All of these will be analysed for evidence coinciding with this alert message for evidence of an attack. After this investigation is completed, a report will be written evaluating the investigation and demonstrating an understanding of the evidence, and what it means. If any compromises are made to the system then this will be highlighted. The report will also summarise the role of the log files provided in the investigation.

PACKET ANALYSIS - ANN'S BAD AIM

For the investigation, a copy of the IP address that Ann's system uses was provided. Using Wireshark that address was filtered for as seen in Figure 1. From doing this it can be seen that there is quite a lot of SSL and TCP traffic. One of the addresses that Ann is communicating with is '64.12.24.50'. Wireshark identified this as being an SSL packet, with a source port of 443, however this may not be correct so the IP address will be investigated further.



Source	Destination	Protocol
192.168.1.158	64.12.24.50	SSL

Figure 1 - Wireshark filter 1

From doing a WHOIS search on this IP address, it is found that the IP address belongs to AOL Inc in Figure 2. It can be assumed that this SSL traffic is associated with AIM (AOL Instant Messenger). As Wireshark assumed these packets were SSL, the packets cannot be read and will need to be decoded into AIM messaging traffic.

OrgName: AOL Inc.

Figure 2 - WHOIS search

This AIM traffic can be decoded using Wireshark, this is done by right clicking the SSL packet, and selecting "Decode as" and in the window select 'AIM' for traffic coming between ports 443, and ports 51128.

From following this TCP stream, the text contents of the AIM messages can be seen in Figure 3. Ann's system is communicating with a user named "Sec558user1", and they keep making reference to a secret recipe in their conversations, this could be the secret recipe that security staff were worried about Ann leaking. All messages making reference to the word "recipe" is extracted using the tool grep. The message "Here's the secret recipe... I just downloaded it from the file server. Just copy to a thumb drive and you're good to go >;-)" was found which highly suggests that suspicious activity is going on between the two addresses and a file may have been transferred as a file named "recipe.docx" is also seen in the output.

```
root@kali:~/Desktop/task1/tools# ngrep -I evidence01.pcap "recipe"
input: evidence01.pcap
match: recipe
#####
T 192.168.1.158:51128 -> 64.12.24.50:443 [AP]
*..a.....E4628778....Sec558user1.....Here's the
  secret recipe... I just downloaded it from the file server. Just co
  py to a thumb drive and you're good to go &gt;;-)...
#####
T 192.168.1.158:51128 -> 64.12.24.50:443 [AP]
*..c.z.....G7174647....Sec558user1.....R..7174647..F.CL...."DE
  ST.....F.....'.....recipe.docx.
#####
T 192.168.1.158:5190 -> 192.168.1.159:1272 [AP]
OFT2.....d.....
Cool FileXfer.....recipe.docx.
.....
#####
```

Figure 3 - Ngrep "recipe"

Since it is now known that there is suspicious activity going on between Ann's computer and an AOL AIM server, Wireshark can be used to filter the packets to only contain the communication involving these two systems. All packets involved with these IP addresses can then be exported to a separate '.pcap' file. Since AIM uses port 5190 to transfer files, a filter for all traffic using this external port can be applied in Wireshark. In Wireshark it can be seen that all file transfers over AIM happened between Ann's computer and IP address "192.168.1.159". The stream of 12264 bytes from Ann's computer to the mystery host, can be seen in Figure 4, and appears to contain an OFT2 file transfer judging by the headers in the packet after investigating it. This stream was extracted and saved as a raw data file.

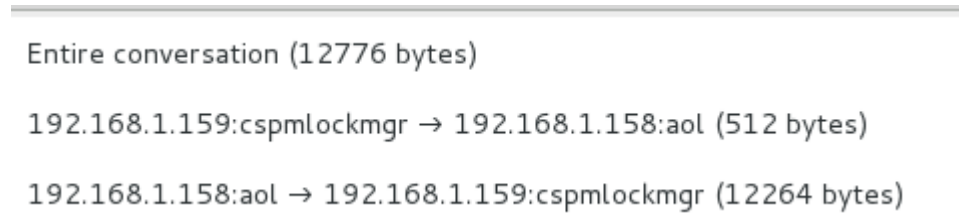


Figure 4 - conversation stream

The file is opened in a hex editor. Knowing the OFT2 packet header format, the size of the file is found to be the value '0x2EE8' which is 12008 bytes in decimal. Since it is known that the file is a 'docx' file, with a magic number of '50 4B', the magic number can be searched for and the offset copied. Using the offset and the file size as the length like in Figure 5, only the blocks making up the file will be selected. The hash value of the file is then checked like in Figure 6 to make sure that there have been no modifications of the file during analysis.

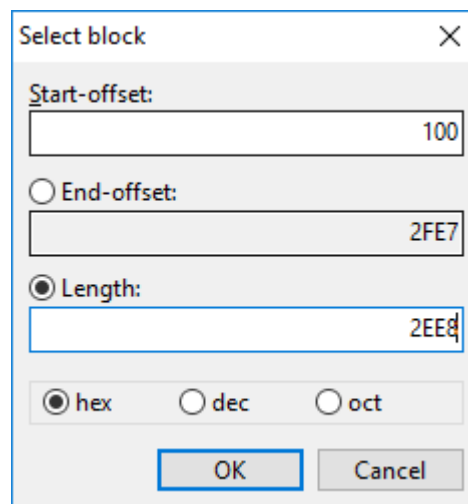


Figure 5 - selecting block

Algorithm	Checksum	Usage
MD-5	8350582774E1D4DBE1D61D64C89E0EA1	

Figure 6 - MD5 checksum

The selected blocks are then saved as a file and opened in a document viewer. The contents of the file resemble a secret recipe, and can be seen in Figure 7.

Recipe for Disaster:

1 serving

Ingredients:

4 cups sugar

2 cups water

In a medium saucepan, bring the water to a boil. Add sugar. Stir gently over low heat until sugar is fully dissolved. Remove the saucepan from heat. Allow to cool completely. Pour into gas tank. Repeat as necessary.

Figure 7 - Recipe for Disaster

Timeline:

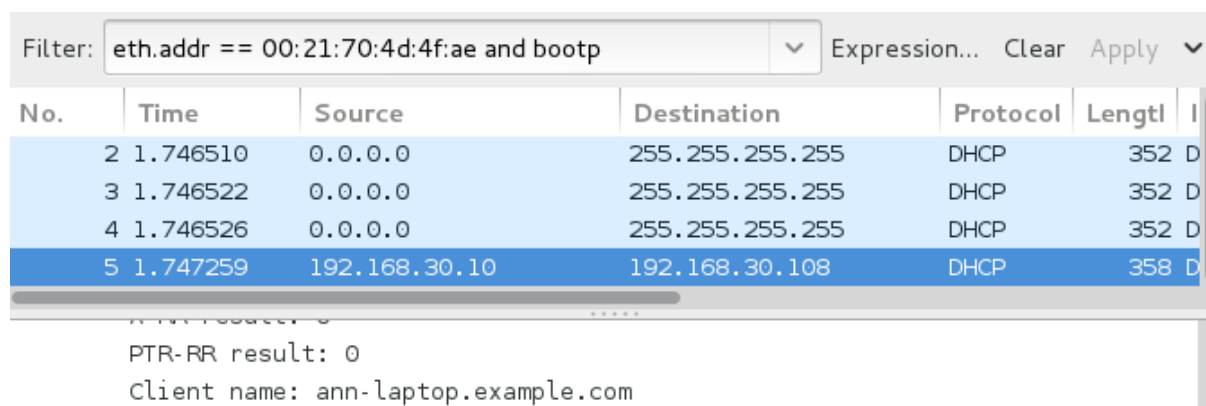
Aug 13, 2009

- 06:58:34.184977000** Discussions between Ann's computer and an AOL server start. From the messages it is seen that the AOL IM user Ann is communicating with has the name 'Sec558user1'.
- 06:58:01.610263000** a file is transferred from Ann's email to 'Sec558user1' titled 'recipe.docx'. Carving this file confirms that a secret recipe has been transferred.
- 06:58:26.641088000** 'sec558user1' sends a message to Ann stating that he "can't wait to sell it on ebay".

PACKET ANALYSIS - ANN'S RENDEVOUS

Ann is on the run, and her location is unknown. Looking at the protocol hierarchy gives a lot of useful information for this investigation into Ann's whereabouts. The bootstrap protocol is used, which could potentially provide a link from the provided MAC address to an IP address. It also shows that the three most used protocols are DNS, SMTP, and IMAP, which suggest that email communications could have maybe been transferred over the network as that is what these protocols are commonly seen working in conjunction for.

If Wireshark is filtered for the bootstrap protocol and Ann's mac address like in Figure 8, then DHCP packets are found. There are 3 DHCP requests followed by an DHCP ACK. The DHCP ACK has a destination IP of 192.168.30.108 for a system with the same MAC address as Ann's computer, as well as a client name suggesting ownership to ann. It is safe to assume that this is the IP address last assigned to Ann. More information from this packet can be found, such as the DNS server having an IP address of "10.30.30.20". It is also be seen that the IP lease time is an hour long, which is corroborated by the explicit renewal time value being half of this.



Filter: eth.addr == 00:21:70:4d:4f:ae and bootp

No.	Time	Source	Destination	Protocol	Length	Info
2	1.746510	0.0.0.0	255.255.255.255	DHCP	352	D
3	1.746522	0.0.0.0	255.255.255.255	DHCP	352	D
4	1.746526	0.0.0.0	255.255.255.255	DHCP	352	D
5	1.747259	192.168.30.10	192.168.30.108	DHCP	358	D

Packet 5 details:

- PTR-RR result: 0
- Client name: ann-laptop.example.com

Figure 8 - Wireshark filter 2

The tool 'Ngrep' is used on the evidence file and manages to extract seven packets containing the string "Ann Dercover". Several IP's are found, including the one belonging to Ann's machine. Ports that are commonly used for IMAP and SMTP were also found further cementing that the packets contain emails. Communications between several email addresses are found using the command in Figure 9, these addresses found are: sneakyg33ky@aol.com, interOpt1c@aol.com, d4rktangent@gmail.com, mistersekritx@aol.com.

```
root@kali:~/Desktop/task1/tools# ngrep "Ann Dercover" -N -t -q -I evidence-  
packet-analysis.pcap  
input: evidence-packet-analysis.pcap  
match: Ann Dercover
```

Figure 9 - Ngrep "Ann Dercover"

The timestamps found for these packets are then used to locate them on Wireshark in order to follow the TCP stream, and view the full communications. The timestamp of the first packet is "20:33:07:203874". From this TCP stream authentication details are found, as seen in figure 10, which are being sent from Ann's computer to the IP address "64.12.168.40" which belongs to interOpt1c@aol.com, encoded with base-64. This is converted to ASCII and reveal the authentication details to Ann's email. The username equals "sneakyg33ky" and the password equals "s00pers3kr1t".


```
AUTH LOGIN
334 VXNlcm5hbWU6
c25lYWt5ZzZmZa3k=
334 UGFzc3dvcmQ6
czAwcGVyczMrcjF0
235 2.7.0 Authentication successful
MAIL FROM: <sneakyg33ky@aol.com>
250 2.1.0 Ok
RCPT TO: <inter0pt1c@aol.com>
```

Figure 10 - Authentication details

The packet with timestamp “20:35:16:962873” was found on Wireshark and so the TCP stream is followed. This packet included communications with the email “mistersekritx@aol.com”. In this stream it is found that a document called “secretrendezvous.docx” was sent to this email as seen in Figure 11.

```
-----=_NextPart_000_00B8_01CC1497.244B3EB0
Content-Type: application/octet-stream;
.name="secretrendezvous.docx"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
.filename="secretrendezvous.docx"
```

Figure 11 – Email evidence

Using the hex editor once again the file was carved and the contents of the file are checked and can be seen in Figure 12. The file contained a specific location, containing the message “Meet me at the fountain near the rendezvous point. Address below. I’m bringing all the cash.” It can be assumed that Ann intends to meet with the email user mistersekritx@aol.com at this spot.

Meet me at the fountain near the rendezvous point. Address below. I’m bringing all the cash.

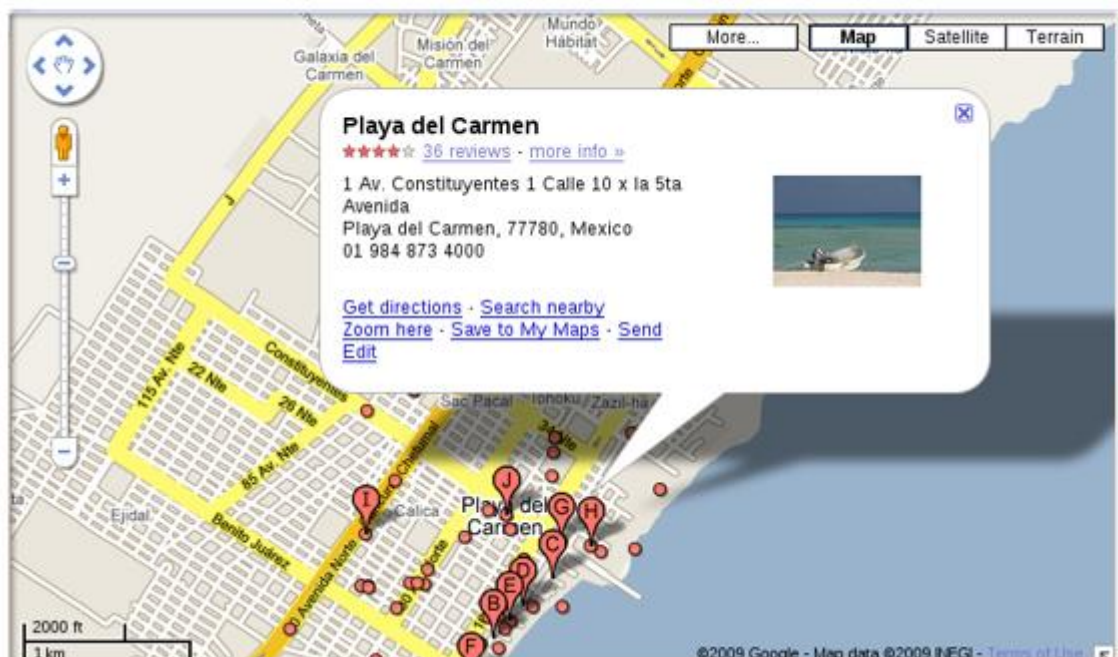


Figure 12 - secretrendezvous.docx

Timeline:

May 17, 2011

- 20:32:03.167145000** Ann's mac address was assigned the IP address 192.168.30.108.
- 20:33:07.203874** this IP address started sending messages under the email address sneakyg33ky@aol.com. The first email was sent to the address interOpt1c@aol.com. The contents of the message were "Hey, can you hook me up quick with that fake passport you were taking about?".
- 20:35:16.962873** an email was sent from sneakyg33ky@aol.com to mistersekritx@aol.com. The email contained a file called secretrendezvous.docx. Carving this file reveals potentially important evidence regarding Ann's location.

STATISTICAL FLOW ANALYSIS - THE CURIOUS MR X.

Since the attacking systems external IP address is known as “172.30.1.77” the flow records from the perimeters Cisco ASA firewall was scanned for records containing this IP address. From doing this it was found that 2173 packets were involved with this IP address as seen in Figure 13. A majority of these packets were sent from the attacking machine to address ‘172.30.1.231’, which is the DMZ’s external address, most of which were denied by the firewall. A variety of different ports were tried against the targeted IP, suggesting the attacker may have been performing a port sweep. It was found that the source port of the hacker switched between 44197 and 44198, possibly to throw off suspicion.

```
Summary: total flows: 2173, total bytes: 319944, total packets: 2173, avg
bps: 2207, avg pps: 1, avg bpp: 147
Time window: 2011-04-27 19:49:29 - 2011-04-27 20:15:36
Total flows processed: 14093, Records skipped: 0, Bytes read: 1763976
Sys: 0.048s flows/second: 293604.2 Wall: 0.524s flows/second: 26875.3
```

Figure 13 - flow record scan 1

During this scan, it was found that the firewall did not deny responses for traffic between the attacker and address ‘172.30.1.231’ on port 22, which there are several for. This can be seen in Figure 14 and suggests that the attackers port sweep successfully found an open port.

```
2011-04-27 19:52:30.111 TCP      172.30.1.77:54348      172.30.1.77:5434
8 ->      172.30.1.231:22      CREATE Ignore      0
```

Figure 14 - Found open port

This is confirmed when looking at the flow record data collected from the Internal and DMZ subnets via the SPAN port using an Argus listener. It should be known that there is a time skew of approximately 8 seconds between the Cisco ASA and the Argus listener. Looking for activity relating to the attacking systems IP address, it is seen that the attacker initiated a connection 3 times in Figure 15. The attacker must have figured out that the port is open then aborts the connection before the TCP handshake is complete. This could be so that the server assumes that there has been a communications error which matches behaviour relating to port scanning.

StartTime	Flgs	Proto	SrcAddr	Sport	Dir
DstAddr	Dport	TotPkts	TotBytes	State	
19:51:55.975054	e	6	172.30.1.77.44197	->	
10.30.30.20.22		3	182	sSR	
19:51:57.280064	e	6	172.30.1.77.44208	->	
10.30.30.20.22		3	182	sSR	
19:51:58.586575	e	6	172.30.1.77.44209	->	
10.30.30.20.22		3	182	sSR	

Figure 15 - 3 Reset connections

The port that was found is commonly used for SSH, which typically requires a log in. Transfers between the attacker and targeted system were consistently the same size, suggesting failed log in attempts. All of a sudden, the packet size changed shown in Figure 16, suggesting that the attacker possibly made a secure shell connection on the targeted system.

```

2011-04-27 20:00:35.612 TCP 172.30.1.77:56359
9 -> 172.30.1.231:22 DELETE Deleted 3755
2011-04-27 20:00:41.962 TCP 172.30.1.77:56361
1 -> 172.30.1.231:22 CREATE Ignore 0
2011-04-27 20:00:41.962 TCP 172.30.1.77:56360
0 -> 172.30.1.231:22 DELETE Deleted 3603

```

Figure 16 – Change in packet size 1

Looking back at the flow records again, it can be seen that a series of full TCP communications are made from the attacker to the host over port 22 every 6 seconds, which can be seen in Figure 17.

```

10.30.30.20.22 43 6609 sSEfF
19:52:44.304186 e 6 172.30.1.77.54349 ->
10.30.30.20.22 42 6543 sSEfF
19:52:50.637603 e 6 172.30.1.77.54350 ->
10.30.30.20.22 42 6543 sSEfF

```

Figure 17 – Full TCP communications

These packets could mean application data has been sent, and they continue until '20:00:44.258927', when two short completed connections are made. After these connections, several longer connections are made that do not finish transmission. This can be seen in Figure 18 and these connections go on until '20:15:55.326776'.

```

20:00:44.258927 e 6 172.30.1.77.56360 ->
10.30.30.20.22 38 6127 sSEfF
20:00:50.611823 e 6 172.30.1.77.56361 ->
10.30.30.20.22 35 5661 sSEfF
20:01:08.784786 e 6 172.30.1.77.56362 ->
10.30.30.20.22 669 147296 sSE

```

Figure 18 - Change in packet size 2

This corroborates with the idea that the attacker possibly made a secure shell connection on the targeted system as this behaviour matches with the usual patterns of a successful brute force attack against a secure shells password.

Looking at the flow records associated with '10.30.30.20' after the attacker accessed the machine, starting about '20:03:33.281463', it can be seen in Figure 19 that the host sent TCP SYN packets to ports 80 and 443 on the internal network and this continued until '20:03:49.599728'.

```

20:03:33.281463 e 6 10.30.30.20.32907 ->
10.30.30.10.80 1 74 s
20:03:33.282212 e 6 10.30.30.20.53778 ->
10.30.30.10.443 1 74 s
20:03:33.383155 e 6 10.30.30.20.53808 ->
10.30.30.10.443 1 74 s
20:03:44.420655 e 6 10.30.30.20.41339 ->
192.168.30.1.80 1 74 s

```

Figure 19 – Port scan 1

The systems that did respond to this port sweep and return reset packets to the attacker are “192.168.30.30” and “192.168.30.90” meaning that the systems exist and are alive. This is shown in Figure 20.

	StartTime	Dir	DstAddr	Dport	SrcPkts	DstPkts	Stat
e	20:03:47.426206	->	192.168.30.30.80		2	2	s
R	20:03:47.930667	->	192.168.30.90.80		3	3	s
R							

Figure 20 - Found systems 1

After the port sweep was successful, the attacker then started sending more SYN packets to the two systems that responded. The number of packets that were sent to the hosts turned out to be exactly 1000. The command entered in Figure 21 shows the specific packets sent during this time period. This indicates that a Nmap port scanner was used as Nmap by default scans the most common 1000 ports for each protocol as mentioned in this blog (Nmap.org, 2009).

```
root@kali:~/Desktop/SFA/argus-clients-3.0.6# ra -z -t 2011/04/27.20:03:49
+2s -nnr ../task2/tools/argus-collector.ra -s dport - ' host 10.30.30.
20 and dst host (192.168.30.30 or 192.168.30.90) ' | grep -v Dport | sort -
u | wc -l
1000
```

Figure 21 - Number of ports scanned

It was figured out that the attacker would have found that port 22 was open on both systems as well as port 514 on system ‘192.168.30.30’. The found systems are shown in Figure 22.

```
root@kali:~/Desktop/SFA/argus-clients-3.0.6# ra -z -t 2011/04/27.20:03:49
+2s -nnr ../task2/tools/argus-collector.ra -s 'host 10.30.30.20 and dst
host (192.168.30.30 or 192.168.30.90) and synack'
```

	StartTime	Flgs	Proto	SrcAddr	Sport	Dir
	DstAddr	Dport	TotPkts	TotBytes	State	
	20:03:49.604221	M *	6	10.30.30.20.46692	->	
192.168.30.90.22		8	560	sSER		
	20:03:49.604225	M *	6	10.30.30.20.36307	->	
192.168.30.30.22		8	560	sSER		
	20:03:49.651694	M *	6	10.30.30.20.39370	->	
192.168.30.30.514		8	560	sSER		

Figure 22 - Port scan 2

Starting at ‘20:04:09.818524’ and ending at ‘20:04:14.525608’, the attacker was most likely performing another port sweep targeting systems with an open 3389 port shown in Figure 23. This port is registered for Microsoft WBT Server, usually used for Windows Remote Desktop, which is most likely what the attacker was looking for.

20:04:09.818524	e	6	10.30.30.20.56212	->
192.168.30.1.3389		1	74	s
20:04:09.818527	e	6	10.30.30.20.56255	->
192.168.30.2.3389		1	74	s
20:04:09.818529	e	6	10.30.30.20.52168	->
192.168.30.3.3389		1	74	s

Figure 23 - Open WRD

The attacker established a TCP connection over port 3389 with three different systems, shown in Figure 24 to have addresses '192.168.30.100', '192.168.30.101', and '192.168.30.102'.

```
root@kali:~/Desktop/SFA/argus-clients-3.0.6# ra -z -t 2011/04/27.20:04:09
+6s -nnr ../task2/tools/argus-collector.ra - 'host 10.30.30.20 and net
192.168.30.0/24 and synack'
```

StartTime	Flgs	Proto	SrcAddr	Sport	Dir
20:04:13.846496	M *	6	10.30.30.20	41371	->
92.168.30.100.3389	8	560	sSER		1
20:04:13.846498	M *	6	10.30.30.20	33786	->
92.168.30.101.3389	8	568	sSER		1
20:04:13.846500	M *	6	10.30.30.20	57473	->
92.168.30.102.3389	8	568	sSER		1

Figure 24 - Found systems 2

After the port sweep there are more flows from '10.30.30.20' to '192.168.30.101' over port 3389 as seen in Figure 25, suggesting a connection from the compromised DMZ to the systems RDP.

92.168.30.101.3389	18	1260	sSEf*		
20:04:54.428013	M *	6	10.30.30.20	34189	->
92.168.30.101.3389	3060	852450	sSE		1
20:05:54.442780	M *	6	10.30.30.20	34189	->
92.168.30.101.3389	1954	285408	sSE		1
20:06:54.549496	M *	6	10.30.30.20	34189	->
92.168.30.101.3389	68	10032	sSE		1

Figure 25 - Connections to '192.168.30.101'

After scanning the system at '192.168.30.101', it was found that a connection over port 21 was made to the attacker's system at '172.30.1.77'. Port 21 is commonly used for FTP which is for transferring files. This implies that the system '192.168.30.101' may have sent a file under the hacker's control to the hacker's machine. All packets that are using FTP ports and are associated with this workstation were searched and can be seen in Figure 26.

port	State	StartTime	SrcAddr	Sport	Dir	DstAddr	Dp
	sSE	20:05:33.393949	192.168.30.101.1603		->	172.30.1.77.21	
	sSE	20:06:50.728909	192.168.30.101.1603		->	172.30.1.77.21	
2	sSEfF	20:06:50.731907	172.30.1.77.20		->	192.168.30.101.500	
3	sSEfF	20:07:03.952171	172.30.1.77.20		->	192.168.30.101.500	
	EfFR	20:11:14.429838	192.168.30.101.1603		<?>	172.30.1.77.21	

Figure 26 - Internal communications to attacker

Matching up the ports and times with the original scan it can be seen in Figure 27 that a large number of bytes were sent from the '192.168.30.101' system to the attacker's address over FTP, suggesting that a file was transferred from inside the network to the attacker.


```
20:07:03.952171 e 6 172.30.1.77.20
92.168.30.101.5003 30 17872 sSEfF
```

Figure 27 - File transfer 1

This is also confirmed when looking at the flow records from the Cisco ASA firewall shown in Figure 28, as the packets match up, with a time skew of approximately 8 seconds.

```
2011-04-27 20:06:55.631 TCP 172.30.1.77:20 172.30.1.77:20
-> 172.30.1.227:5003 DELETE Deleted 15872
```

Figure 28 - File transfer 2

TimeLine

2011-04-27

- **19:51:54** port sweep against the DMZ '172.30.1.231' is performed by an external system with address '172.30.1.77', suspected of being an attacker.
- **19:52:38.749332** Attacker attempts to connect to SSH (port 22) of DMZ, and performs a suspected brute-force attack.
- **20:01:08** Attacker successfully authenticates and starts a connection on the DMZ's SSH.
- **20:03:33.281463** DMZ performs a port sweep on ports '80' and '443', and discovers systems "192.168.30.30" and "192.168.30.90".
- **20:03:49.599723** DMZ performs Nmap port scan on the discovered systems.
- **20:04:09.818524** DMZ performs port sweep of the internal network on port '3389'. 3 systems with this port open were discovered.
- **20:04:32.330776** DMZ connects to '192.168.30.101' over RDP on port 3389.
- **20:05:33** the system '192.168.30.101' connects to '172.30.1.77' over port 21, commonly used for FTP.
- **20:07:03.952171** file transferred from '192.168.30.101' system to the attacker's address.

Preventing Further Compromise

The attacker makes use of a brute force attack. The best way to prevent a brute force attack is to lock out users who attempt to log in too many times. In the SSH protocol, the amount of authentication attempts a user is given is also configured in the 'sshd_config' file where a user can configure the maximum number of authentications allowed by a given system. This should be applied on all ports using SSH. Configuring the 'sshd_config' for better security can be seen in this blog (Boelen, 2018).

To prevent the hacker from being able to perform massive port scans on the network, the Cisco ASA switch can be configured to detect scans, although as seen some scanners can attempt to avoid some detection rules by altering the order of their addresses, or ports. The securest way would be to have no single port open on the DMZ listening to incoming connections from the internet. This blog (Harvey, 2018) describes other small practices that should be followed to maximise security include regularly reviewing firewall rules and keeping the software up-to-date.

From the evidence it is clear that the attacker gained access to a systems remote desktop straight away, meaning it is very insecure. There are several things that can be done to improve security of this protocol, such as using a strong password, as well using a firewall in the form of an RDP gateway for restricting access to only certain users. These basic security tips can be seen in this blog (Security.berkeley.edu, 2018).

WIRELESS NETWORKS - HACKME, INC.

An anonymous tip was received about a known hacker named 'InterOptic', who has been on the run and is suspected to be hiding in the area. Meanwhile a system admin named 'Joe', working at a local company named 'HackMe Inc', has reported some strange behaviour regarding his private WAP. He gets dropped and finds out that he can no longer access it. He hands all the traffic he captures to the Forensic Investigation team in order to figure out what has happened, and whether it may be relevant to the case. Joe states that he is the only one who should have access to the WAP, and that his device's MAC address is '00:11:22:33:44:55'.

It was found that the BSSID was '00:23:69:61:00:d0' from looking at the Beacon frames filtered on Wireshark. Through looking at the data frames it was also found that all traffic seemed to be encrypted, which is commonplace for wireless traffic. The investigation into traffic was started by looking into the addresses which received association frames from the BSSID. It was found that Joe successfully associated 4 times to the WAP. An unknown system with address '1c:4b:d6:69:cd:07' associated 68 times, and another with address 'de:ad:be:ef:13:37' just once. Looking at the sent and received data frames it was found that the unknown station with address '1c:4b:d6:69:cd:07' sent by far the most packets to the broadcast address 'ff:ff:ff:ff:ff:ff' – approximately three times more than Joe's device sent. It was found that the Broadcast frames coming from the address '1c:4b:d6:69:cd:07' lasted less than 69 seconds. Large segments of network activity, coming from an unknown address that was not permitted by Joe, could be a major concern. The other unknown station with address 'de:ad:be:ef:13:37' can also be seen sending data frames to the system with address '00:23:69:61:00:ce'. The address of this system is very similar to the WAP's BSSID, meaning that it is likely the WAP's MAC address that it uses to participate as a logically distinct station. When looking at outbound traffic from the WAP's STA interface, it can be seen that most frames were sent to Joe's station, however the station to receive the second most frames is the unknown station with address 'de:ad:be:ef:13:37', this suggests that the unknown system managed to breach the network if it was able to communicate with the WAP's STA interface without Joe's permission.

The investigation looked into management frames, and it was found that the WAP sent out just under 15000 management frames. This is abnormal considering that the second most management frames sent by a system was approximately a hundredth of this size. Looking into these management frames sent by the WAP, it was found that the vast majority of them are being sent to the unknown address '1c:4b:d6:69:cd:07'. The Management frames sent to this address consisted mostly of disassociation packets. The second most popular management frames sent out by the WAP, were to the broadcast address and were Deauthentication frames. Looking at the times that these Disassociation and Deauthentication frames were sent out, it was found that the WAP was broadcasting them at the same time. There are legitimate reasons for a device to send to the broadcast address, such as an ARP request, although in this case the volume looks suspiciously high to be just an ARP request.

Looking at the timeline of events, the behaviours exhibited by the address '1c:4b:d6:69:cd:07' seem to be typical of someone performing a WEP cracking attack. The suspected attacker begun sending both Authentication and Association Requests simultaneously, this is a necessary step in a WEP cracking attack because for an access point to accept a packet, the source MAC address must already be associated. Following this the suspected attacker '1c:4b:d6:69:cd:07' begins flooding the Broadcast address with larger volumes of data frames. This behaviour resembles a WEP cracking attack, as the attacker listens for ARP requests then reinjects them back into the network. Shortly after this, all traffic related to this stops, and then another unknown system 'de:ad:be:ef:13:37' sends its first data frame to the WAP's STA interface. This suggests that the systems are associated and the WEP cracking attack was successful, as the attacker figured out the WEP key.

This attack worked because wired Equivalent Privacy is an outdated encryption standard for wireless networks, and is therefore quite defective. The WEP cracking attack is possible for a number of reasons. The Initialisation Vector (IV) is only 24-bits long and is sent in cleartext. These security issues for WEP are illustrated in a conference by (Mekhaznia and Zidani, 2015). This Means that they can easily be gathered for decoding purposes. The attacker will gather a lot of the initialisation vectors through relaying broadcast requests, and the WAP will rebroadcast ARP packets and generate new IVs. This is known as injecting and it is done to speed up the process as it involves having the WAP resend selected packets over and over at a fast rate, which coincides with the spike in unique IVs. The attacker will then listen to the network traffic and save it, as the objective in a WEP cracking attack is to obtain a large number of IVs in a short period of time, which seems like what the suspected attacker is trying to do. The purpose of this step in a WEP cracking attack is to obtain the WEP key from the IVs gathered. This theory was tested, using tools like 'aircrack-ng' and following a tutorial by (darkAudax, 2010), against the packet capture provided by Joe, and the key was able to be found showing that the attacker would have been able to crack the key using the same frames. The tool airdecap-ng was then used to decode the encrypted frames using the discovered WEB key. The traffic between the unknown system 'de:ad:be:ef:13:37' and the WAP's STA was then inspected, and it was found that the system successfully authenticated using the password 'admin', proving that the unknown system gained unauthorised access.

To conclude, it is most likely that Joe's WAP was a victim to a WEP cracking attack performed by a malicious user with the mac address 'de:ad:be:ef:13:37'. This is apparent when looking at the timeline of events that took place in the packet capture, which coincides with the attributes of a WEP cracking attack. The station '1c:4b:d6:69:cd:07' is the station used by the attacker to crack the WEP key, as shortly after it stops sending network traffic, the 'de:ad:be:ef:13:37' system connects to the WAP and successfully authenticates, it is therefore very likely these machines are associated. The theory was further cemented by the fact that the investigation team were able to crack the key using the data frames from the packet capture, meaning the malicious user could have done the same. The decoded traffic showed that system 'de:ad:be:ef:13:37' managed to authenticate, proving that the attacker successfully breached the network.

NETWORK IDS AND ANALYSIS - INTEROPTIC SAVES THE PLANET

MacDaddy Payment Processor has deployed 'Snort', a network intrusion detection system that was used for detecting potentially malicious traffic. An alert was logged by snort, claiming that suspicious executable code was sent to the internal network from an external source. The security staff working at Macdaddy's gathered all of the snort alerts that are relevant to the case, as well as the corresponding log file of all TCP communications for an investigation into this alert. The rules that the NIDS follows is also provided for the investigator.

The investigation started by looking at the alert file for details regarding the suspicious alert. The log file that contained all of the TCP communications involved in the event were then searched for the same instance of the alert for further examination. The packet was sent from an external internet source with address '172.16.16.218' to an internal system on the network with address '192.168.1.169'. Consulting the rules that were used by the snort sensor, it can be seen that this packet triggered an alert through using excessive 'No Operation' (NOP) bytes. This behaviour was alerted as excessive NOPs are commonly used with buffer overflow attacks as illustrated in this blog (Hieu, 2018). The packet that triggered this alert was further investigated and was found to contain HTTP headers. through looking at the headers it was found to contain a JPEG file. Looking at this packet in Hexadecimal the suspected NOP sled appeared to be within this file. The file was extracted referencing the magic numbers associated with JPEG files found online (Asecuritysite.com, 2010) and the extracted file was passed onto a reverse-engineering malware specialist. The headers also show that the web page that the packet was sent from was using a web proxy. The proxy's cache could potentially be further analysed for information regarding how this JPEG was actually downloaded by the internal system '192.168.1.169'.

The external source '172.16.16.218' who delivered the possibly malicious JPEG was further investigated, and the alert file output by Snorts sensor was searched for any more suspicious activity. Nothing was found so the target system with address '192.168.1.169' was searched instead and it was found that over 100 alerts were set off relating to this address. These alerts were of 3 different categories, the first being an 'INFO Web Bug' alert, which was triggered by far the most. This alert, also known as invisible GIF, is a method used by web servers to track user behaviour and are not something that should really be worried about. These alerts were sent from 42 different web servers which were found to have no connection. The other alert was titled "Tcp Window Scale Option found with length > 14" which is an alert created by the snort pre-processor for parsing TCP options and watching for any suspicious behaviour. This alert was sent out by the address '192.168.1.169' to a variety of hosts on the internal network. Examining these packets further it was found that certain values stayed the same like ports and flags, as well as the sequence and acknowledgement numbers. These packets do not seem realistic and are most likely crafted.

Looking at the timeline of the events that transpired can give some more insight into the case. It is known that the '192.168.1.169' host was actively browsing the internet as there was an influx of alerts commonly used by web servers for tracking purposes. This went on for half an hour. During this time a JPEG file was downloaded by the host '192.168.1.169' from a web server. This was the file that contained the suspected NOP sled and set off the alarm that triggered this investigation. A few minutes later the system that downloaded the potentially malicious payload started sending out crafted packages to other systems on the internal network for a short duration. Interpreting these events can lead to theories of the investigation, such as it being a drive by exploit. Drive-by downloads can happen through accidentally downloading a file without knowing the contents, or through the exploitation of security vulnerabilities in systems or applications as illustrated in (Zaharia, 2016). The victim system '192.168.1.169' may have fallen victim to a drive-by exploit and downloaded a malicious file, and once the payload of the file was triggered, the victim system started sending out crafted

packets to other hosts on the system, possibly for some sort of network reconnaissance. This is just speculation at this point however, and to reach a solid conclusion for the investigation, more needs to be done. For example, analysis of the malware, or analysis of the web proxy cache that was the JPEG was downloaded from.

To conclude, it seems likely that a drive-by exploit did take place, although further analysis will need to take place in order to find out the specifics of the exploit. The investigation found out from looking at the NIDS sensor alerts, and the corresponding TCP dump file that an internal system with address '192.168.1.169' may have fallen victim to a drive-by download attack. The system consequently downloaded a JPEG file that may be malicious, as it contained what looked like a NOP sled, used for buffer overflow exploits. The payload for this malicious file is suspected to be responsible for the crafted packets that were sent to other systems around the internal network, suspected of being for gathering network information.

EVENT LOGS - LONE SH4RK'S REVENGE

Security staff for a company called Bob's dry cleaners noticed that their DMZ server with address '10.30.30.20' was suddenly hit with a wave of failed authentication attempts over secure shell. Since Bob's dry cleaners were on the alert from previous attacks, and they also store a lot of credit card numbers, the security team have been collecting logs from the servers, workstations and the firewall to be sent to a server for storage. The suspicious failed authentication behaviour was recorded in the authorisation logs, which records all authentications and privileged commands performed on the server. The security staff at Bob's dry cleaners are concerned that an attacker may have gained access and stolen their customers credit card information, so all of the log files coinciding with this suspicious behaviour will be investigated for any evidence of systems being compromised.

The investigation started with an examination of the authorisation log file for the failed log-in activity on the DMZ's SSH server called 'baboon.srv'. It is clear from looking at these logs that an external host with an address of '172.30.1.77' attempted to log-in to the administrative account 'root', which is the default administrator on most systems using Linux. there were two types of logs for these failed authentications, one that records a single log-in attempt, and a following alert that records two more log-in attempts. Counting all of the authentications logs that were made to root from the external address, it was found that 121 authentication attempts were made in total. Looking at the time stamps for each log-in attempt, it was found that they were spaced out by about 6 seconds for every 3 attempts. The volume of failed log-in attempts, along with the consistent delays between each attempt highly suggests that a brute force password-guessing attack took place against the 'baboon.srv' SSH server. After these failed authentications stopped, the same pattern of behaviour happened on the server for another SSH user 'bob'. Counting all of the failed authentication logs for this user is equal to 85. Looking at the timings of these logs, both users suspected of being brute-forced stopped trying to authenticate after the first failed log-in attempt which is not followed by a failed second two log-in attempts, suggesting that the attacker either exhausted the word list they were using, or that the attack was successful and the attacker '172.30.1.77' cracked the passwords for both users. Assuming the same word list was used for both attacks, the hacker would have cracked the password for the user bob as less failed authentication logs were recorded.

Filtering for successful SSH log-in attempts, it can be seen that the user bob successfully authenticated twice, the first attempt matched the pattern of the brute-force attack, the second was approximately half a minute later. This suggests that the second attempt was done manually, and further confirms that a successful brute force attack did occur against the server. Following the second log-in attempt there was suspicious activity coming from the server found in the authentication logs, which the attacker is suspected to be in control of. The attacker was able to execute privileged commands, suggesting he did find roots password. This can be seen as the sudo command is used to open the authorisation logs in a text editor, showing that the attacker may have been trying to cover up their footprints. This would have failed however, as the logs are sent to a remote log collection server. The attacker then tried to sniff traffic on the network using a tool called 'tcpdump', possibly for mapping purposes. In the authentication logs it can then be seen that the tool called Nmap was installed, which is a powerful network scanner. About 7 minutes following this the attacker logs out. It is assumed that within this 7-minute timeframe the attacker performed some malicious activity, however nothing more in the authentication logs can be seen, so the activity must not have required sudo permissions.

The logs recorded by the firewall were consulted for more information regarding the activity that occurred on the 'baboon-srv' server after Nmap was installed. As suspected, there was an increase in activity on the server following the time that Nmap was installed. The server made connections to about 212 IP addresses with over 200 ports, which highly suggests that Nmap was used to scan all open ports on the internal network. Shortly after this scan finished, a second spike in activity occurred

where more connections were made, although this time they were only made to port 3389 over a range of IP addresses. This port is commonly used as a Remote Desktop Protocol (RDP) for windows systems, which is what the attacker must have been searching for. After this scan, one of the workstations '192.168.30.101' with port 3389 open received more connections than any other system following this, so it has likely had its Remote Desktop targeted by the attacker. Looking at the workstation logs for this system, it can be seen that the connection was made with the user 'bob' - the account the attacker already knew the credentials for on the SSH server – suggesting that the same credentials were used allowing the attacker access to this workstation. It can also be seen that following this authentication, a program called 'FTP.exe' was executed. FTP is used for transferring files remotely, meaning that the attacker may have tried to leak files from this workstation. This is confirmed when looking back at the firewall logs for activity from this workstation after the FTP program was executed, as a connection to the external IP address '172.30.1.77' was made less than a minute later over port 21 – the port commonly reserved for FTP.

To conclude this investigation, it was found through the log files provided that an external attacker with address '172.30.1.77' was able to successfully gain access to the internal network and leaked important information, most likely related to credit card numbers. The attacker was able to do this through a brute-force password guessing attack against the 'baboon.srv' server's secure shell, which was found by investigators looking at the authorisation logs. After gaining the credentials of both 'root' and 'bob' the attacker then logged into the server as bob and used sudo commands to try and cover up their tracks and also install the network scanning tool Nmap. Using this tool, the attacker found that the workstation with address '192.168.30.101' had its Remote Desktop port open, and managed to connect to it using the same credentials found previously through the brute-force attack. This activity was found when looking at the logs from the Cisco ASA firewall. From this workstation, the attacker then connected to their external system '172.30.1.77' over port 21. It is evident that FTP was used when looking at the workstation's logs, as the program was executed shortly before the transfer.

CONCLUSION

To conclude this document, the activities were effective at giving insight into the range of topics involved in network forensics. It has provided the ability to conduct a forensic investigation regarding the featured topics. The investigations have allowed the opportunity to gather evidence from certain sources of a network, write reports about these investigations, create theories of the case, and also create timelines of the events that transpired. Each provided a good understanding of how the specific topic fits into network forensics and can be used within a certain scenario. For the first investigation into “Ann’s Bad Aim”, evidence of a recipe being leaked was found, and a timeline of the events leading up to this was successfully reconstructed. For the second part of this investigation into “Ann’s Rendezvous”, evidence of Ann’s location was extracted from a file sent over email and a timeline of events were successfully created for this investigation as well. These cases both demonstrated how effective packet analysis can be for gathering evidence in a forensic investigation. For the second topic, the investigation into the “The Curious Mr X.” case successfully found evidence of a brute force password guessing attack against the DMZ’s secure shell through studying the flow records, along with evidence of the attacker proceeding to scan the internal network and pivoting to other systems. It is also highly likely that the attacker exfiltrated a file from the compromised system to his own system. A timeline of these events was created, and mitigations for these attacks were also researched and provided for the security team to prevent future compromises to their network. Overall, the case gave a good understanding how analysing statistical flows can be useful in cases for identifying any compromised hosts, confirming any data leakages, or revealing the activity performed by given addresses over certain time periods.

For the other three topics, reports were written on the investigations. The third topic on wireless networks with the investigation called “HackMe, Inc.” gave good insight into how wireless traffic can be analysed for evidence. It is unclear whether the station with MAC address “1c:4b:d6:69:cd:07” is ‘interOptic’, however it is clearly a malicious hacker as it performed a WEP cracking attack against Joe’s WAP. The report discusses the evidence that the investigation found and discusses how a WEP cracking attack may have been performed. For the second topic, the investigation titled “InterOptic Saves the Planet”, gave a good understanding of a NIDS’s role in securing a network, and how its logged alerts can be useful for forensic investigations. From this investigation it can be seen that an internal host was browsing the internet, when they most likely fell victim to a drive-by download attack, which executed the shell code on their system, triggering the NIDS alert. Evidence was also gathered from the NIDS logs of the infected host sending crafted packets, suspected of being used for reconnaissance purposes, to other systems on the internal network. The report discusses this theory in detail, and also discusses how the NIDS alerts were used as evidence. For the final investigation titled “LOne Sh4rk’s Revenge”, the topic was regarding the use of event logs in forensic cases. This investigation gave good insight into how these logs can be used to provide better security, and how they can be analysed in order to reconstruct and visualise certain events in a network. From looking at the authentication logs, it was found that a malicious hacker did perform a brute force password guessing attack against the secure shell of the DMZ server, they then logged in, and performed some privileged commands that were seen in the log file. They were found to have scanned the network and pivoted to other internal system in the network. Looking at workstation logs, as well as firewall logs, revealed that the attacker most likely transferred a file from the compromised system back to the original attacking system. The report discusses the evidence that the investigation found using the event logs, and discusses the theory of the attack in detail.

REFERENCES

Boelen, M. (2018). OpenSSH security and hardening. [online] Linux Audit. Available at: <https://linux-audit.com/audit-and-harden-your-ssh-configuration/> [Accessed 28 Nov. 2018].

Harvey, C. (2018). Fine-tuning Firewall Rules: 10 Best Practices. [online] Esecurityplanet.com. Available at: <https://www.esecurityplanet.com/network-security/finetune-and-optimize-firewall-rules.html> [Accessed 28 Nov. 2018].

Security.berkeley.edu. (2018). Securing Remote Desktop (RDP) for System Administrators | Information Security and Policy. [online] Available at: <https://security.berkeley.edu/resources/best-practices-how-articles/system-application-security/securing-remote-desktop-rdp-system> [Accessed 28 Nov. 2018].

Mekhaznia, T. and Zidani, A. (2015). Wi-Fi security analysis. In: The International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015). [online] Procedia Computer Science 73, pp.172 – 178. Available at: https://ac.els-cdn.com/S1877050915034705/1-s2.0-S1877050915034705-main.pdf?_tid=badca5f3-30db-4c06-bdb6-a4bec49bf742&acdnat=1544544520_117faade497cce0d05be7e4a1180ad9a [Accessed 2 Dec. 2018].

darkAudax (2010). Tutorial: Simple WEP Crack. [Blog] Available at: https://www.aircrack-ng.org/doku.php?id=simple_wep_crack [Accessed 2 Dec. 2018].

Asecuritysite.com. (2010). Digital Forensics Magic Numbers. [online] Available at: <https://asecuritysite.com/forensics/magic> [Accessed 2 Dec. 2018].

Zaharia, A. (2016). How Drive-by Download Attacks Work – From Disbelief to Protection. [online] Heimdal Security Blog. Available at: <https://heimdalsecurity.com/blog/how-drive-by-download-attacks-work/> [Accessed 3 Dec. 2018].

Nmap.org. (2009). Port Specification and Scan Order | Nmap Network Scanning. [online] Available at: <https://nmap.org/book/man-port-specification.html> [Accessed 3 Dec. 2018].

Hieu, L. (2018). Exploit stack-based buffer overflow using NOP-sled technique. [online] Luong Trong Hieu. Available at: <https://lthieu.wordpress.com/2012/11/10/exploit-stack-based-buffer-overflow-using-nop-sled-technique/> [Accessed 5 Dec. 2018].