

Dale Stubbs - 14024149

Analysis of the Network Captures provided for this assignment task

Network Capture Analysis

# Abstract

For a forensic examiner the ability to comprehend the files associated with packet captures from a suspected network based attack is paramount. The ability to determine malicious traffic from regular traffic can be considered the most important ability. This paper looks into three different network packet capture files. The author looks into these files and attempts to determine if any malicious behaviour is being conducted within the file. Once the malicious behaviour has been detected the author attempts to provide a source of mitigation against these forms of malicious behaviours.

Keywords: PCAP, Network, Malicious Behaviour, Mitigation, Port Scan, Dictionary Password Attack, Brute Force Password Attack.

Contents

[List of Tables 3](#_Toc478123907)

[List of Images 3](#_Toc478123908)

[Introduction 4](#_Toc478123909)

[Pre-Investigation Hash Values 4](#_Toc478123910)

[Frameworks 4](#_Toc478123911)

[Tables of Terms 5](#_Toc478123912)

[Capture 1 6](#_Toc478123913)

[Capture Overview 6](#_Toc478123914)

[Table of Actors 6](#_Toc478123915)

[Malicious Behaviour 6](#_Toc478123916)

[Mitigation 7](#_Toc478123917)

[Capture 2 8](#_Toc478123918)

[Capture Overview 8](#_Toc478123919)

[Table of Actors 8](#_Toc478123920)

[Malicious Behaviour 8](#_Toc478123921)

[Mitigation 10](#_Toc478123922)

[Capture 3 11](#_Toc478123923)

[Capture Overview 11](#_Toc478123924)

[Table of Actors 11](#_Toc478123925)

[Malicious Behaviour 11](#_Toc478123926)

[Mitigation 13](#_Toc478123927)

[Post Investigation Hash Values 13](#_Toc478123928)

[Appendices 14](#_Toc478123929)

[Appendix 1 14](#_Toc478123930)

[Appendix 2 15](#_Toc478123931)

[Appendix 3 16](#_Toc478123932)

[Appendix 4 16](#_Toc478123933)

[Appendix 5 16](#_Toc478123934)

[Appendix 6 16](#_Toc478123935)

[Appendix 7 17](#_Toc478123936)

[Appendix 8 17](#_Toc478123937)

[Appendix 9 18](#_Toc478123938)

[Appendix 10 18](#_Toc478123939)

[Appendix 11 19](#_Toc478123940)

[Appendix 12 21](#_Toc478123941)

[References 24](#_Toc478123942)

# List of Tables

* Table 1 – Pre Investigation Hash Values
* Table 2 – Actors
* Table 3 – Acronyms
* Table 4 – Frequently Used Concepts
* Table 5 – Actors within NetworkCapture1.pcap
* Table 6 – Actors within NetworkCapture2.pcap
* Table 7 – Actors within NetworkCapture3.pcap
* Table 8 – Post Investigation Hash Values
* Table 9 – HTTP Requests Made in the Captures

# List of Images

* Image 1 – Friday joins using ARP
* Image 2 – Friday and Jarvis HTTP Requests and Responses
* Image 3 – Lucille MDNS Queries
* Image 4 – Multiple TCP SYN Requests Together
* Image 5 – Multiple TCP RST, ACK Responses Together
* Image 6 – Lucille leaves after completing the port scan
* Image 7 – Gideon connects to Jarvis
* Image 8 – Gideon stores a file
* Image 9 – Friday’s ARP Request
* Image 10 – Start of the password attack
* Image 11 – Port switch
* Image 12 – Gideon and Jarvis traffic
* Image 13 – Gideon, Jarvis and Friday traffic
* Image 14 – Friday is successful at cracking the password
* Image 15 – Friday transfers the confidential file
* Image 16 – Friday disconnects
* Image 17 – Gideon and Jarvis’ HTTP and TCP traffic
* Image 18 – Friday connects to Gideon
* Image 19 – Port scan on Gideon by Friday
* Image 20 – Friday finds open port
* Image 21 – Start of password attack
* Image 22 – Password not cracked
* Image 23 – TCP 3-way handshake
* Image 24 – HTTP Request and Response
* Image 25 – Graph of Port Scan in Network Capture 1
* Image 26 – Graph of Password Attack in Network Capture 2
* Image 27 – Graphs of Port Scan and Password Attack in Network Capture 3

# Introduction

Hash values are generated on each piece of evidence before (to give a ‘base’ value for comparison purposes) and after (to ensure the data has not been modified) any investigation on the material in question can be carried out. This is done to ensure the validity of the evidence gathered from the material being scrutinised.

This is accomplished by how hash values are generated. Hash values are *‘a numeric value of a fixed length that uniquely identifies data.’* *(Msdn.microsoft.com, 2016).* There are multiple ways to generate a hash value for any material which ensures that the data integrity has not been altered throughout the investigation.

The hash values are generated based on the hexadecimal values of the data provided. This means that a modification to a single byte of data will result in the hash value being generated after the investigation not matching the original hash value, rendering all of the evidence gathered as inadmissible in court.

|  |  |  |  |
| --- | --- | --- | --- |
| Evidence Name | MD5 Hash Value | SHA1 Hash Value | SHA512 Hash Value |
| Networkcapture1.pcap | b834dbf7fad0d1e07  4529b03ad141110 | 050bb43df2c7e8c03  ab920295370f79651  733288 | 6526bfd37e7ed6e9643c9a96e49fc  4e561371f292cafd8194ad7acc6ad  cabad6f1c9ecf3efc20871dc2e0c3e  8f3d3e5004007203ecb6f5f3b87a6  7384f7a4178 |
| Networkcapture2.pcap | 744d49d048735262  2d624837c2cf589c | b330894d9e5243cb8  06cef971e478233a9  d39b7a | ed66c7b49cde415ad0e4bc7abedf3b  a967cdfb39318374700cef73f22e5c  9364dc6e8f6b552d921fb18993a193  27d622abaed421576b263d9938a1a  fc36ee313 |
| Networkcapture3.pcap | e12b2d244d23b988  2301cd4ad52ed1de | b02de56eb4e6aac8a6  944775be95bae259c  f5483 | b5cb9b447c2713833b617abe07649  b895a027f46fd362e3e92fa3a5737e  e2c08802d86cd8e7d1d4431800271  b33cc179f22977144480917cb54f7506acd896b0 |

# Pre-Investigation Hash Values

Table 1: Pre Investigation Hash Values

Multiple hash values have been generated to ensure a multiple point validity check after the investigation of the material has been completed.

# Frameworks

ACPO Guidelines (Appendix 1). I will be using the ACPO guidelines for the ensuring that the evidence provided is admissible in court. More specifically, I will be using Principle 1, *No action taken by law enforcement agencies, persons employed within those agencies or their agents should change data which may subsequently be relied upon in court. (Williams, 2012)*

This refers to the hashing of the files before and after being worked on to ensure authenticity of the evidence gathered.

Wireshark - I will be using Wireshark to analyse and scrutinise the data provided to attempt to discover any and all malicious/inappropriate behaviour during the provided Network Packet Captures (pcap files).

# Tables of Terms

|  |  |
| --- | --- |
| Actor Name | Description |
| Target | The victim of the attack |
| Aggressor | The perpetrator of the attack |
| Intermediary | A middleman to the attack |
| Spectator | An innocent bystander that has nothing to do with the attack. |

Table 2: Actors

|  |  |  |
| --- | --- | --- |
| Acronyms | Full Term | Definition |
| TCP | Transmission Control Protocol | A standard the defines how a connection between two nodes |
| FTP | File Transfer Protocol | A standard internet protocol designed for the transfer of files between computers via the TCP/IP connection. |
| SYN | Synchronise Flag | A flag of the TCP connection |
| ACK | Acknowledge Flag | A flag of the TCP connection |
| RST | Reset Flag | A flag of the TCP connection |
| FIN | Finish Flag | A flag of the TCP connection |
| MAC Address | Media Access Control Address | A unique hardware number assigned to the computer |
| HTTP | Hypertext Transfer Protocol | A protocol for transferring files such as text, images, sound and video files on the World Wide Web. |
| TYPE I | Type Image | Sets the transfer data type to binary form |
| PASV | Passive FTP | Sets the FTP to a passive form (appendix 6) |

Table 3: Acronyms

|  |  |
| --- | --- |
| Concept Name | Description |
| Port Scan | A scan of all ports on a computer to determine any access points that may be exploited. |
| HTTP Request and Response | A transfer of HTTP packets between host and client |
| Password Attack | An attempt to guess someone’s password without authority to do so. |

Table 4: Frequently Used Concepts

# Capture 1

## Capture Overview

This is the first Network Capture provided named ‘Networkcapture1.pcap'. My name is Dale Stubbs and my student number is 14024149.

This network capture was completed on 11/02/2010 at 16:20, monitors network traffic for 5 minutes and ceases at 16:25. A total of 48,370 packets were captured in this time.

## Table of Actors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | IP Address | Description | Mac Address | Role |
| Jarvis | 192.168.56.101 | Server: apache/2.2.11 | CadmusCo\_00:48:4d | Target |
| Lucille | 192.168.56.104 | N/A | CadmusCo\_1d:1b:ab | Aggressor |
| Friday | 192.168.56.1 | N/A | CadmusCo\_8e:ca:c2 | Spectator |

Table 5: Actors within NetworkCatpure1.pcap

## Malicious Behaviour

* At approximately 16:20 on the 11/02/2010 the network capture began.
* An initialization of the TCP three-way handshake is started by Friday aimed at Jarvis using the SYN packet.
* Friday then sends out an ARP announcement to ensure that Jarvis is aware of their IP and MAC Address.

Image 1: Friday joins using ARP

* The connection between Friday and Jarvis is then completed (See appendix 1) via port 2812 of Friday and port 80 of Jarvis.
* Friday then begins to request several different items from Jarvis with little success (appendix 2).

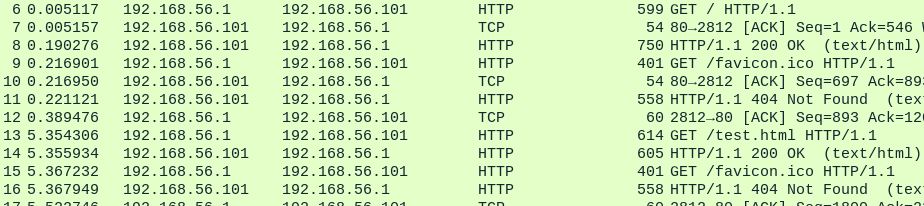


Image 2: Friday and Jarvis HTTP Requests and Responses

* Friday then switches to port 2813 using the three-way handshake (appendix 3) and proceeds to retrieve more items from Jarvis.
* Jarvis then terminates the connections to both ports 2812 and 2813 of Friday using the FIN packet (appendix 2).
* Lucille then joins the group via IGMPv3 (appendix 4) protocol and begins a series of MDNS queries (appendix 5).

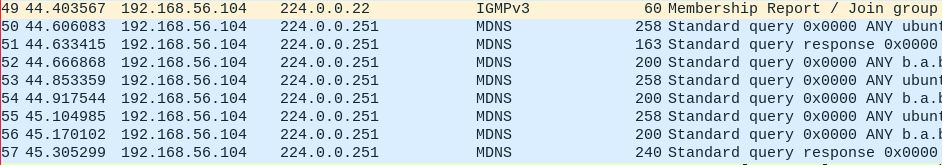


Image 3: Lucille MDNS Queries

* Lucille then sends out an ARP packet to request information about Jarvis’ IP Address.
* Lucille then pings Jarvis and then sends out and ARP packet to tell Jarvis their IP and MAC Addresses
* Friday reconnects to Jarvis and requests some more HTTP Get Requests (appendix 6).
* Lucille then resends the ARP packet for Jarvis’ information.
* Lucille then begins a Sequential Port Scan on Jarvis. Packet 94 to packet 48,261 is the complete run down of the port scan. This theory is backed up by the groups of SYN requests in between single RST, ACK replies at different stages of the scan.

Image 5: Multiple TCP RST, ACK Responses Together

Image 4: Multiple TCP SYN Requests Together

* At packet 48,321 Lucille leaves the group whilst Friday and Jarvis hand packets back and forth regarding HTTP requests.

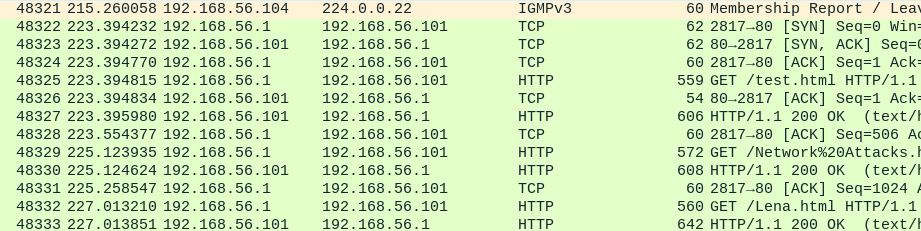


Image 6: Lucille leaves after completing the port scan

* Network capture concludes at 16:25 of the same date.

## Mitigation

Port scan on networks can easily be flagged up by using *‘Network security applications [can be] configured to alert administrators if they detect connection requests across a broad range of ports from a single host.’ (Lifewire, 2016)* This would enable anyone being the subject of a port scan to easily detect the port scan and can take the necessary steps to stop the attacker in their tracks.

Another way of mitigating an attack of this calibre would be to block the IP address that is guilty of port scanning you. This can be done via the router itself. You would need to have administrative privileges on the router in order to accomplish this. If you have administrator privileges then the use of the iptables rule is how to accomplish blocking ip addresses from the server. This program allows the user to manually add malicious IP addresses to a list of blocked IP addresses denying them contact in any future connection attempts. There is also a software available to complete this for you, Snort. Snort is an open-source network intrusion detection and prevention system. Using either of these aforementioned approaches would effectively mitigate against this form of attack.

# Capture 2

## Capture Overview

This is the second Network Capture provided named ‘Networkcapture2.pcap’. The examiners name is Dale Stubbs and my student number is 14024149.

This network capture was completed on 15/02/2010 at 15:38, monitors network traffic for one hour and 40 minutes and ceases at 17:18. A total of 14,240 packets were captured in this time.

## Table of Actors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | IP Address | Description | Mac Address | Role |
| Jarvis | 192.168.56.101 | Server: apache/2.2.11 | CadmusCo\_00:48:4d | Intermediary |
| Gideon | 192.168.56.102 |  | CadmusCo\_1d:1b:ab | Target |
| Friday | 192.168.56.1 |  | CadmusCo\_8e:ca:c2 | Aggressor |

Table 6: Actors within NetworkCapture2.pcap

## Malicious Behaviour

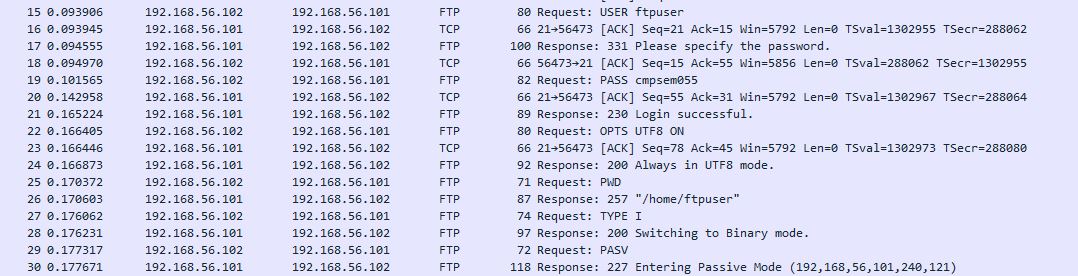
* At approximately 15:38 on 15/02/2010 the network capture begins and we are greeted with Gideon logging onto Jarvis using the user name ‘ftpuser’ and password ‘cmpsem055’ and begin to set a transfer mode using the commands ‘TYPE I’ and ‘PASV’ (see ‘Table 3: Acronyms’) and then prints the current working directory ‘PWD’ followed by listing all of the contents of the directory ‘LIST’. This exchange happens between packets 1 to 44.

Image 7: Gideon connects to Jarvis

* Following this exchange Gideon then logs onto Jarvis again and proceeds to store a document called Confidential Information.doc in the ‘/home/ftpuser’ directory. This occurs between packets 55 and 63

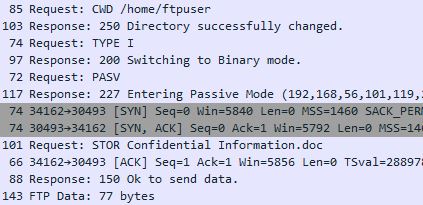


Image 8: Gideon stores a file

Changes current directory

Stores the document

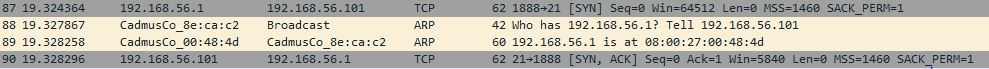
* Friday then connects to Jarvis at packet 87 on port 21 (appendix 7). Friday then issues an ARP command to request the MAC Address of Jarvis.

Image 9: Friday’s ARP Request

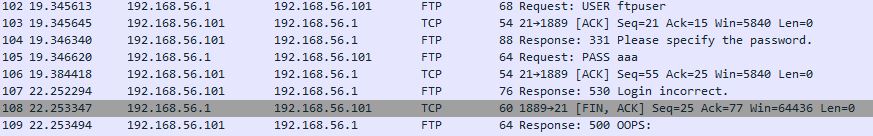
* Friday then begins a password attack in an attempt to gain access to the files stored within Jarvis. The password attack used in this scenario is called a ‘Dictionary Password Attack’ (appendix 8). The first attempt to access the files is at packet 105. Friday uses the user name ‘ftpuser’ and uses the password ‘aaa’.

Image 10: Start of the password attack

* After each unsuccessful password attempt, Friday switches the port that they are transmitting the requests from.

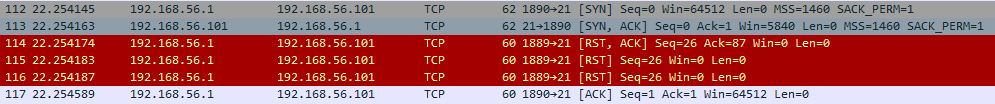


Image 11: Port switch

* During the attack Gideon and Jarvis continue to transfer packets for HTTP requests as well as FTP requests showing that Gideon has continue to go about their business unaware of the attack by Friday.

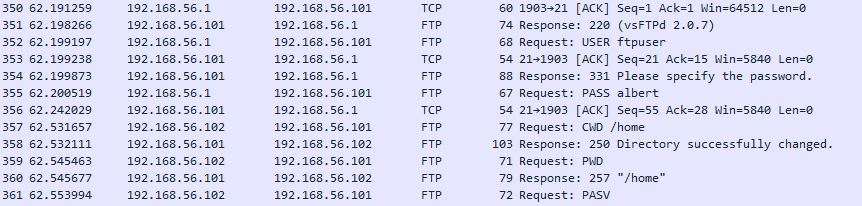


Image 12: Gideon and Jarvis traffic

Image 13: Gideon, Jarvis and Friday traffic

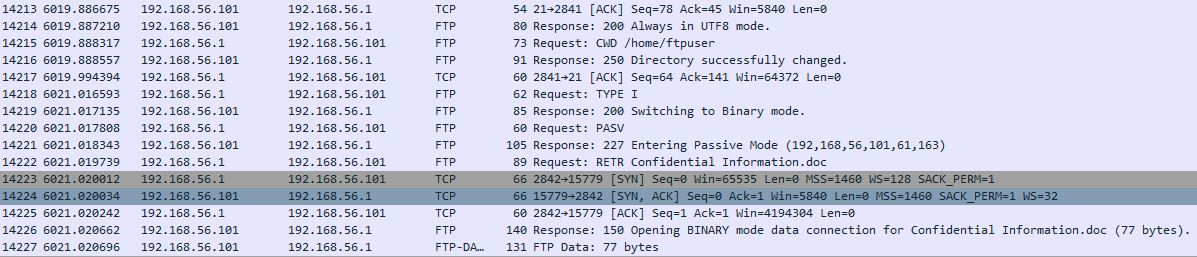
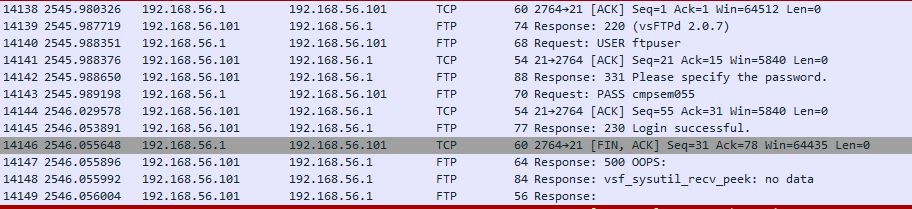
* At packet 14,143 Friday successfully logs on using the user name ‘ftpuser’ with the password ‘cmpsem055’ and successfully transfers the document ‘Confidential Information.doc’ to their own machine meaning the confidentiality of the contained information has now been destroyed at packet 14223.

Image 15: Friday transfers the confidential file

Image 14: Friday is successful at cracking the password

* Friday then terminates the connection to Jarvis at packets 14,238, 14,239 and 14,240.

Image 16: Friday disconnects

* Packet capture terminates at 17:18.

## Mitigation

There are several steps that can be taken to mitigate attempted password attacks on a network. They are all based on the network administrators’ perspective. Firstly, enforce a ‘lockout’ procedure when someone attempts to log in incorrectly after a set number of attempts, secondly, attempt to enforce the use much stronger passwords. For example, a phrase could be used as opposed to a word. Something along the lines of ‘mywifeisbeautiful’. *"It would take a computer about 898 THOUSAND YEARS to crack your password" (Collider, 2016).* Thirdly, not allow password reuse. Ensuring that the same password is never used more than once, fourthly, not allowing clear text storage and instead using a form of password salting and hashing (appendix 9), and finally never allow default passwords to be used after user creation. Some people never change from the default password, whether this is through laziness or simply not knowing how to change the password. This creates a very easy point of failure within the network as anyone can easily acquire a list of default passwords from the internet and use each of these very easily. The focus of mitigating password attacks is to make the cracking of the password more costly than what the attacker would gain from cracking the password.

# Capture 3

## Capture Overview

This is the third and final Network Capture provided named ‘Networkcapture3.pcap’. The examiners name is Dale Stubbs and my student number is 14024149.

This network capture was completed on 8/03/2010 at 12:21, monitors network traffic for 5 minutes and ceases at 12:46. A total of 81,189 packets were captured in this time.

## Table of Actors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | IP Address | Description | Mac Address | Role |
| Jarvis | 192.168.56.101 | Server: apache/2.2.11 | CadmusCo\_00:48:4d | Intermediary |
| Gideon | 192.168.56.102 |  | CadmusCo\_1d:1b:ab | Target |
| Friday | 192.168.56.1 | N/A | CadmusCo\_8e:ca:c2 | Aggressor |

Table 7: Actors within NetworkCapture3.pcap

## Malicious Behaviour

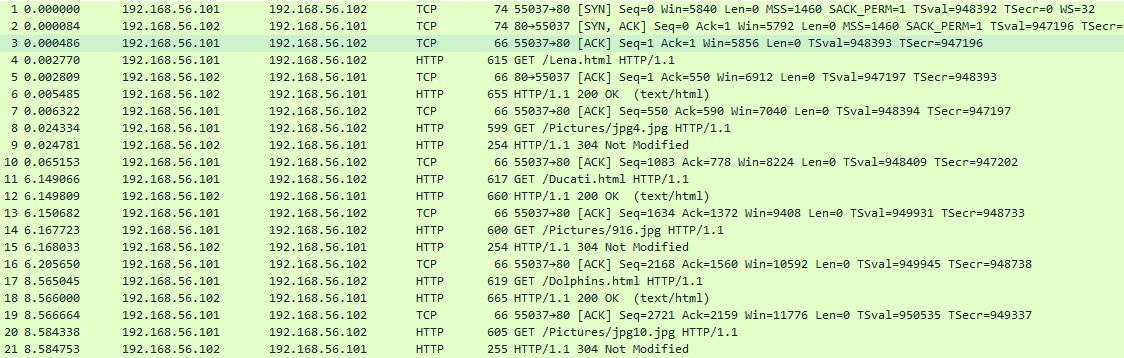
* At approximately 12:21 on 8/03/2010 Gideon and Jarvis were exchanging HTTP request and response packets from packets 1 to 29.

Image 17: Gideon and Jarvis’ HTTP and TCP traffic

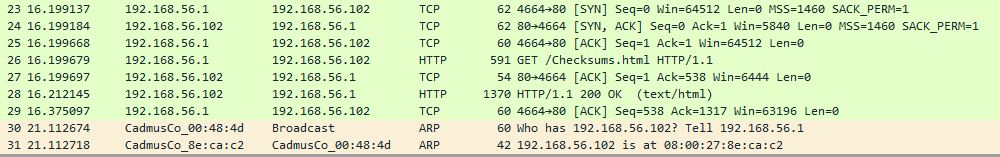
* Friday then connects directly to Gideon at packet 23. Friday sends a HTTP request for the Checksums.html page at packet 26.

Image 18: Friday connects to Gideon

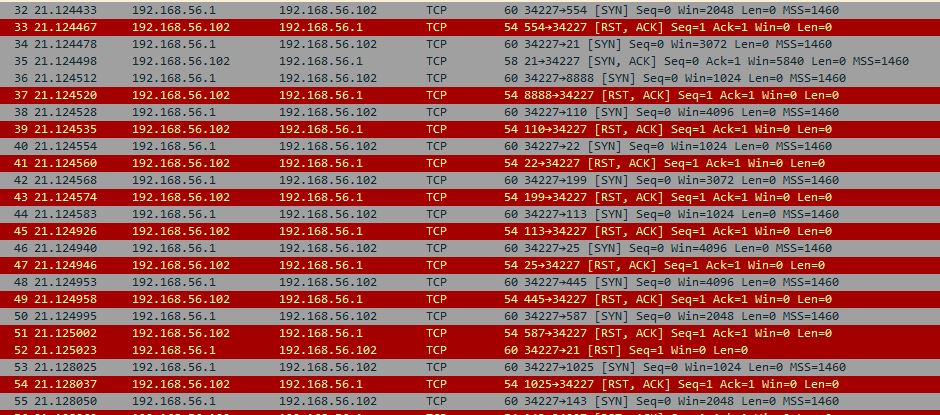
* Friday begins a port scan on Gideon at packets 32 to 88. A successful connection is found at packet 95 on port 21.

Image 19: Port scan on Gideon by Friday

Image 20: Friday finds open port



* Friday then begin a ‘Brute Force Password Attack’ (appendix 10) from packets 130 until the end of the capture.

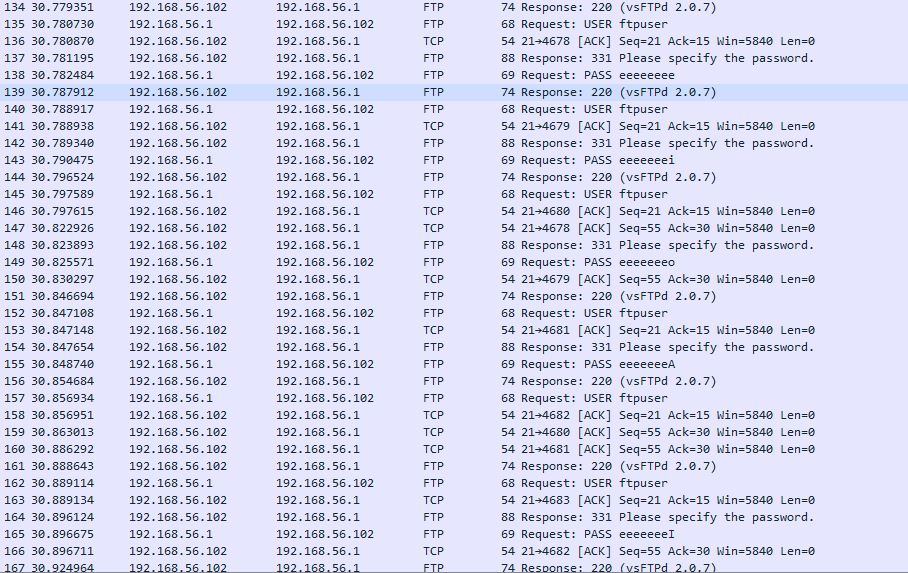


Image 21: Start of password attack

* Friday fails to crack the password during this attempted password attack before the capture is terminated. This is unusual as Friday is attempting to crack the password for the user name ‘ftpuser’ using a Brute Force attack after successfully cracking the password for this user using a Dictionary attack less than one month earlier.

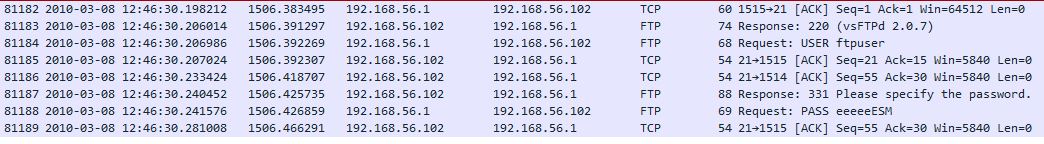


Image 22: Password not cracked

## Mitigation

In this network capture, there are two things to mitigate against. These are a port scan and a Brute Force password attack.

As covered in capture one of this assignment, port scans can be mitigated by using a network monitor set to alert the administrator for multiple connection requests from and to a specific machine.

Finally, any form of password attack can be mitigated by using the steps in the previous capture. The key to successfully mitigating a Brute Force type attack is to use longer passwords so the attacker must spend more time cycling through all of the possible combinations. As such, it would make their efforts much less valuable than the contents they are trying to access. A key point here is that all passwords can be cracked using a Brute Force attack so changing the password at regular intervals would greatly decrease the possibility of the attacker gaining the password.

# Post Investigation Hash Values

|  |  |  |  |
| --- | --- | --- | --- |
| Evidence Name | MD5 Hash Value | SHA1 Hash Value | SHA512 Hash Value |
| Networkcapture1.pcap | b834dbf7fad0d1e07  4529b03ad141110 | 050bb43df2c7e8c03  ab920295370f79651  733288 | 6526bfd37e7ed6e9643c9a96e49fc  4e561371f292cafd8194ad7acc6ad  cabad6f1c9ecf3efc20871dc2e0c3e  8f3d3e5004007203ecb6f5f3b87a6  7384f7a4178 |
| Networkcapture2.pcap | 744d49d048735262  2d624837c2cf589c | b330894d9e5243cb8  06cef971e478233a9  d39b7a | ed66c7b49cde415ad0e4bc7abedf3b  a967cdfb39318374700cef73f22e5c  9364dc6e8f6b552d921fb18993a193  27d622abaed421576b263d9938a1a  fc36ee313 |
| Networkcapture3.pcap | e12b2d244d23b988  2301cd4ad52ed1de | b02de56eb4e6aac8a6  944775be95bae259c  f5483 | b5cb9b447c2713833b617abe07649  b895a027f46fd362e3e92fa3a5737e  e2c08802d86cd8e7d1d4431800271  b33cc179f22977144480917cb54f7506acd896b0 |

Table 8: Post Investigation Hash Values

# Appendices

## Appendix 1

**ACPO Guidelines**

*“2. SECTION 2 – THE PRINCIPLES OF DIGITAL EVIDENCE*

*2.1 PRINCIPLES*

***2.1.1 Principle 1****: No action taken by law enforcement agencies, persons employed within those agencies or their agents should change data which may subsequently be relied upon in court.*

***2.1.2 Principle 2****: In circumstances where a person finds it necessary to access original data, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.*

***2.1.3 Principle 3****: An audit trail or other record of all processes applied to digital evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.*

***2.1.4 Principle 4****: The person in charge of the investigation has overall responsibility for ensuring that the law and these principles are adhered to.” (Williams, 2012)*

## Appendix 2

**HTTP Page Requests**

|  |  |
| --- | --- |
| HTML Address | Contents of HTML Page |
| 192.168.56.102/test.html |  |
| 192.168.56.102/favicon.ico | Unable to display contents |
| 192.168.56.102/Ducati.html | Unable to display contents |
| 192.168.56.102/Dolphins.html | Unable to display contents |
| 192.168.56.102/Checksums.html |  |
| 192.168.56.102/Network%20Attacks.html |  |
| 192.168.56.102/Lena.html | Unable to display contents |
| 192.168.56.102/HTML.html |  |

Table 9: HTTP Requests Made in the Captures

## Appendix 3

**TCP Three-Way Handshake**

*“The TCP three-way handshake in Transmission Control Protocol (also called the TCP-handshake; three message handshake and/or SYN-SYN-ACK) is the method used by TCP set up a TCP/IP connection over an Internet Protocol based network. TCP's three way handshaking technique is often referred to as "SYN-SYN-ACK" (or more accurately SYN, SYN-ACK, ACK) because there are three messages transmitted by TCP to negotiate and start a TCP session between two computers.” (Inetdaemon.com, 2016)*

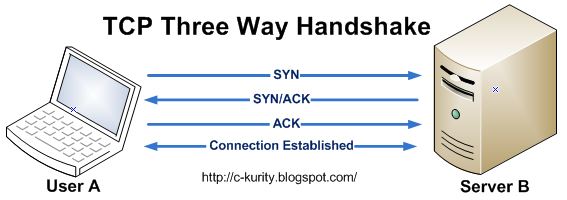


Image 23: TCP 3-way handshake

## Appendix 4

**IGMPv3**

*“Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) are the Multicast Group Membership Discovery (MGMD) protocols. They are essentially the same protocol, with IGMP used for IPv4 multicast groups and MLD used for IPv6 multicast groups. These protocols are used between end systems (often desktops) and the multicast router to request data for a given multicast group.”* *(Metaswitch.com, 2016)*

## Appendix 5

**MDNS**

*“Multicast DNS is a joint effort by participants of the IETF Zero Configuration Networking (zeroconf) and DNS Extensions (dnsext) working groups. The requirements are driven by the Zeroconf working group; the implementation details are a chartered work item for the DNSEXT group. Most of the people working on mDNS are active participants of both working groups. While the requirements for Zeroconf name resolution could be met by designing an entirely new protocol, it is better to provide this functionality by making minimal changes to the current standard DNS protocol. This saves application programmers from having to learn new APIs, and saves application programmers from having to write application code two different ways — one way for large configured networks and a different way for small Zeroconf networks. It means that most current applications need no changes at all to work correctly using mDNS in a Zeroconf network. It also means that engineers do not have to learn an entirely new protocol, and current network packet capture tools can already decode and display DNS packets, so they do not have to be updated to understand new packet formats.” (Cheshire, 2016)*

## Appendix 6

**HTTP**

*“HTTP allows for communication between a variety of hosts and clients, and supports a mixture of network configurations. To make this possible, it assumes very little about a particular system, and does not keep state between different message exchanges. This makes HTTP a stateless protocol. The communication usually takes place over TCP/IP, but any reliable transport can be used. The default port for TCP/IP is 80, but other ports can also be used. Custom headers can also be created and sent by the client. Communication between a host and a client occurs, via a request/response pair. The client initiates an HTTP request message, which is serviced through a HTTP response message in return…………*

*…………* *URLs reveal the identity of the particular host with which we want to communicate, but the action that should be performed on the host is specified via HTTP verbs. Of course, there are several actions that a client would like the host to perform. HTTP has formalized on a few that capture the essentials that are universally applicable for all kinds of applications.*

*These request verbs are:*

*GET: fetch an existing resource. The URL contains all the necessary information the server needs to locate and return the resource.*

*POST: create a new resource. POST requests usually carry a payload that specifies the data for the new resource.*

*PUT: update an existing resource. The payload may contain the updated data for the resource.*

*DELETE: delete an existing resource.” (Code Envato Tuts+, 2016)*

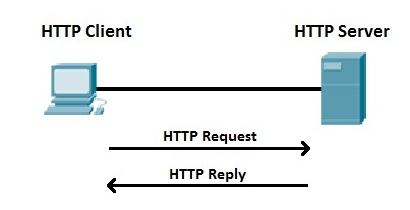


Image 24: HTTP Request and Response

## Appendix 7

**Port 21 & Passive FTP**

*The objectives of FTP are 1) to promote sharing of files (computer programs and/or data), 2) to encourage indirect or implicit (via programs) use of remote computers, 3) to shield a user from variations in file storage systems among hosts, and 4) to transfer data reliably and efficiently. FTP, though usable directly by a user at a terminal, is designed mainly for use by programs. (Postel and Reynolds, 1985)*

## Appendix 8

**Dictionary Password Attack**

*“A dictionary attack is a method of breaking into a password-protected computer or server by systematically entering every word in a dictionary as a password.”* *(SearchSecurity, 2016)*

The dictionary that is used can be obtained from anywhere and many of these dictionaries are readily available online for anyone to acquire.

## Appendix 9

**Password Hashing**

*“The general workflow for account registration and authentication in a hash-based account system is as follows: The user creates an account. Their password is hashed and stored in the database. At no point is the plain-text (unencrypted) password ever written to the hard drive. When the user attempts to login, the hash of the password they entered is checked against the hash of their real password (retrieved from the database). If the hashes match, the user is granted access. If not, the user is told they entered invalid login credentials. Steps 3 and 4 repeat every time someone tries to login to their account. In step 4, never tell the user if it was the username or password they got wrong. Always display a generic message like "Invalid username or password." This prevents attackers from enumerating valid usernames without knowing their passwords.”* *(Hornby, 2016)*

## Appendix 10

**Brute Force Password Attack**

*“Just as a criminal might break into, or "crack" a safe by trying many possible combinations, a brute force cracking application proceeds through all possible combinations of legal characters in sequence. Brute force is considered to be an infallible, although time-consuming, approach.”(SearchSecurity, 2016)*

## Appendix 11

**Python Script for Pcap Visualisation**

*#################################################*

*# Dale Stubbs - 14024149 #*

*# This script is designed to read in the #*

*# Network Captures from the coursework. #*

*# It gathers the information from the #*

*# capture and plots the information on #*

*# a graph using MatPlotLib. #*

*# Last Edited - 21/11/2016 #*

*#################################################*

*import dpkt*

*from dpkt.tcp import TCP*

*import sys*

*import matplotlib.pyplot as plt*

*import socket*

*# Plots a graph based on the port number and time stamp.*

*def plot\_graph1(ports, packets, in\_ip):*

*f = plt.figure(1)*

*plt.plot(packets, ports, 'b-')*

*plt.title("14024149 \n Port Scan")*

*plt.ylabel('Port Number')*

*plt.xlabel('Time Stamp of Packet (Aggressor IP: ' + in\_ip + ')')*

*f.show()*

*# Plots a graph based on the incorrect password attempts and time stamp.*

*def plot\_graph2(attempts, packets, in\_ip):*

*g = plt.figure(2)*

*plt.plot(packets, attempts, 'b-')*

*plt.title("14024149\n Password Attack")*

*plt.ylabel('Number of Incorrect Password Attempts')*

*plt.xlabel('Time Stamp of Each Attempt (Aggressor IP: ' + in\_ip + ')')*

*g.show()*

*def main():*

*x = 0*

*while x != 1:*

*if len(sys.argv) < 3:*

*print 'Usage: python NetCapAnalysis.py [Capture File] [IP of Interest/Aggressor IP]'*

*sys.exit()*

*else:*

*capture = sys.argv[1]*

*in\_ip = sys.argv[2]*

*x = 1*

*# Reads the PCAP file in using the 'read binary' command*

*f = open(capture, 'rb')*

*pcap = dpkt.pcap.Reader(f)*

*count = 1*

*ports = []*

*packets1 = []*

*packets2 = []*

*attempts = []*

*# Loops through each packet in the capture*

*for ts, buf in pcap:*

*eth = dpkt.ethernet.Ethernet(buf)*

*ip = eth.data*

*tcp = ip.data*

*# Filters the traffic based on the IP address*

*#print 'Works'*

*if type(ip.data) == TCP:*

*source\_ip = socket.inet\_ntoa(ip.src)*

*if source\_ip == in\_ip:*

*# Appends the Port Number and time stamp to the*

*# appropriate lists*

*packets1.append(ts)*

*ports.append(tcp.dport)*

*# Filters traffic again based on the '530' response code*

*# being present in the TCP Data.*

*# 530 is the code for 'Incorrect Login'*

*if 'PASS' in str(tcp.data):*

*packets2.append(ts)*

*attempts.append(count)*

*count += 1*

*# Check to see if the attack is a port scan or password attack.*

*with open('tcp.dport.txt', 'w') as q:*

*for line in ports:*

*q.write(str(line) + '\n')*

*q.close()*

*if '3' in capture:*

*plot\_graph1(ports, packets1, in\_ip)*

*plot\_graph2(attempts, packets2, in\_ip)*

*raw\_input()*

*else:*

*if len(attempts) < 10:*

*plot\_graph1(ports, packets1, in\_ip)*

*raw\_input()*

*else:*

*plot\_graph2(attempts, packets2, in\_ip)*

*raw\_input()*

*if \_\_name\_\_ == '\_\_main\_\_':*

*main()*

## Appendix 12

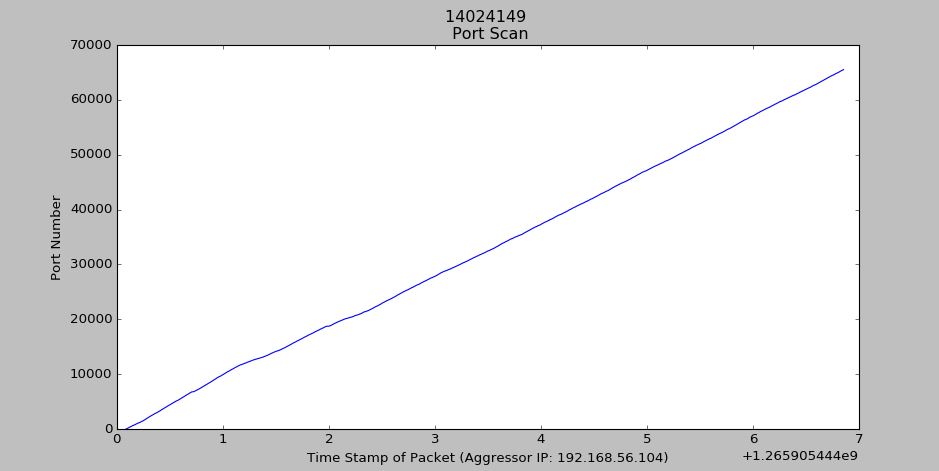
**Images Created Using the Above Script**

Image 25: Graph of Port Scan in Network Capture 1

The graph above show the port numbers and the timestamp of each packet within the NetworkCapture1 pcap file. The surge from port 1 to port 65,534 shows a sequential increase of the port numbers in a very short space of time leading to the conclusion that a sequential port scan being completed within the pcap file.

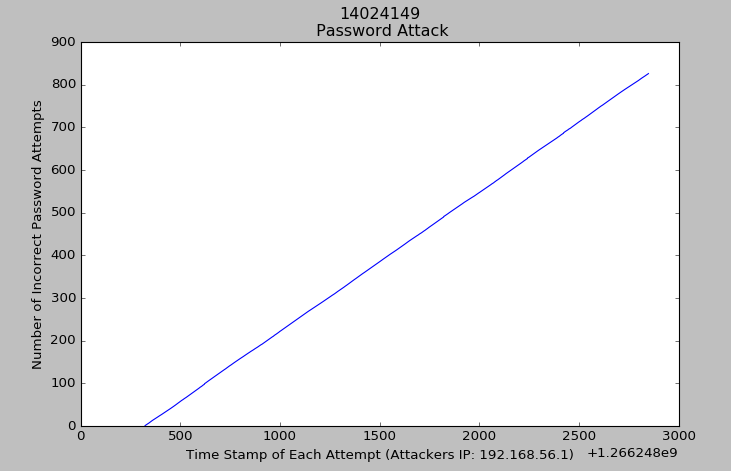
The graph above shows the number of attempts made to input a password showing a password attack within the NetworkCapture2 pcap file. Having almost 900 incorrect password entries within the file in such a small amount of time would suggest that a password attack is occurring.

Image 26: Graph of Password Attack in Network Capture 2

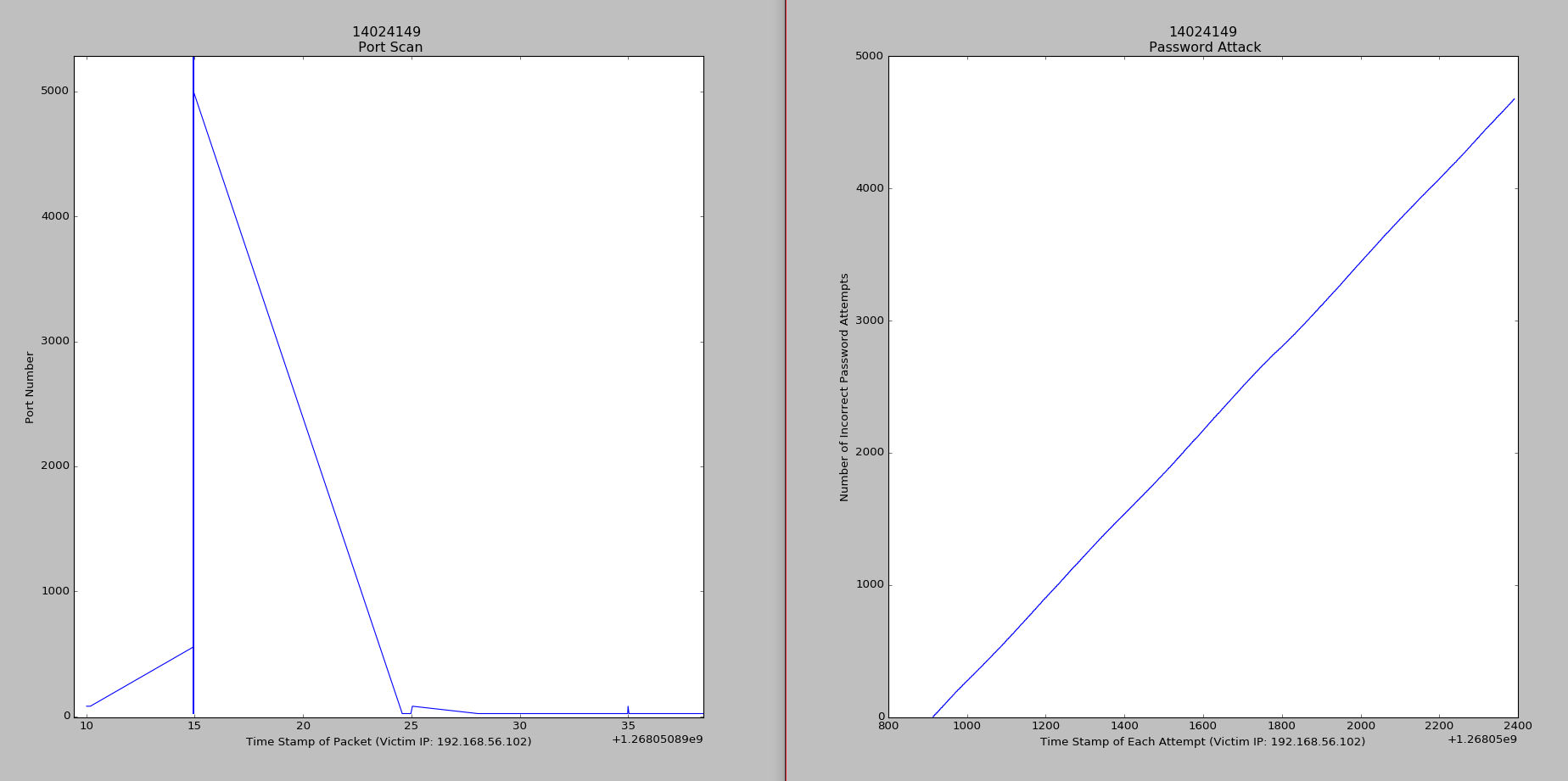
Although the graph on the left doesn’t look like a typical port scan, due the micro timings between the timestamps the significant changes are not visible however, this is confirmed as a port scan due to the port numbers being accessed in such a small time scale. The graph on the right show the number of attempts made to input a password showing a password attack within the NetworkCapture3 pcap file.

Image 27: Graph of Port Scan and Password Attack in Network Capture 3

# References

Msdn.microsoft.com. (2016). Ensuring Data Integrity with Hash Codes. [online] Available at: https://msdn.microsoft.com/en-us/library/f9ax34y5(v=vs.110).aspx [Accessed 5 Nov. 2016].

Williams, J. (2012). ACPO Good Practice Guide. 5th ed. [ebook] Metropolitan Police Service, p.6. Available at: http://www.digital-detective.net/digital-forensics-documents/ACPO\_Good\_Practice\_Guide\_for\_Digital\_Evidence\_v5.pdf [Accessed 5 Nov. 2016].

Lifewire. (2016). *Wondering How Port Scanning Works? Here's the Answer*. [online] Available at: https://www.lifewire.com/introduction-to-port-scanning-2486802 [Accessed 30 Oct. 2016].

Inetdaemon.com. (2016). TCP 3-Way Handshake (SYN,SYN-ACK,ACK) - InetDaemon's IT Tutorials. [online] Available at: http://www.inetdaemon.com/tutorials/internet/tcp/3-way\_handshake.shtml [Accessed 12 Dec. 2016].

Metaswitch.com. (2016). *What is Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)?*. [online] Available at: http://www.metaswitch.com/resources/what-is-internet-group-management-protocol-igmp-multicast-listener-discovery-mld [Accessed 30 Oct. 2016].

Cheshire, S. (2016). Multicast DNS. [online] Multicastdns.org. Available at: http://www.multicastdns.org/ [Accessed 30 Oct. 2016].

Code Envato Tuts+. (2016). HTTP: The Protocol Every Web Developer Must Know - Part 1. [online] Available at: https://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177 [Accessed 30 Oct. 2016].

Collider, S. (2016). How Secure Is My Password?. [online] Howsecureismypassword.net. Available at: https://howsecureismypassword.net/ [Accessed 5 Nov. 2016].

Postel, J. and Reynolds, J. (1985). FILE TRANSFER PROTOCOL (FTP). 1st ed. [ebook] Available at: https://tools.ietf.org/pdf/rfc959.pdf [Accessed 12 Dec. 2016].

SearchSecurity. (2016). What is dictionary attack? - Definition from WhatIs.com. [online] Available at: http://searchsecurity.techtarget.com/definition/dictionary-attack [Accessed 5 Nov. 2016].

Hornby, T. (2016). Salted Password Hashing - Doing it Right - CodeProject. [online] Codeproject.com. Available at: http://www.codeproject.com/Articles/704865/Salted-Password-Hashing-Doing-it-Right [Accessed 5 Nov. 2016].

SearchSecurity. (2016). What is brute force cracking? - Definition from WhatIs.com. [online] Available at: http://searchsecurity.techtarget.com/definition/brute-force-cracking [Accessed 7 Nov. 2016].