

# Forensic Investigations (FORIN)

## Coursework

### REPORT

Word Count: 3643

#### Disclaimer

By submitting this report for marking, I **fully understand** that and I **agree**:

- This is an independent piece of coursework, and it is expected that I have taken responsibility for all the design, implementation, analysis of results and writing of the report.
- For the Turnitin plagiarism software, the match score of the report must be below 20% overall and less than 5% to an individual source (excluding appendices). Marks will be deducted from the final mark according to the exact amount of similarity that exceeds the 20% margin.
- This marking scheme (rubric) sets out what would be expected for each marking band for this module's coursework.
- The University "general criteria applicable to essays, reports and aspects of projects and dissertations" also applies (on Moodle).
- It is blind marked.
- Any technical help, high-level advice and suggestions which may be provided in-person (e.g., labs) and electronically, has no contribution to or an indication of my final mark.
- Knowledge of a peer's work and their final mark is not justification for what my own mark should be.
- Any examples provided were used for ideas only, and "having followed an example" is not justification for what my mark should be.

# Word Count

Section	Word Count
Introduction	215
Design	313
Implementation	495
Results	633
Discussion / Analysis	1074
Conclusion	413
Total	3143

## Table of Contents

Introduction .....	4
What are honeypots? .....	4
How does a honeypot work? .....	4
Honeypot Design .....	5
Honeypot Implementation .....	6
Server Creation .....	6
Firewall Configuration .....	6
Port Configuration .....	7
Username and Password Configuration .....	7
Working Honeypot .....	8
Results .....	9
Server Interaction .....	9
Extracted Usernames and Passwords .....	10
Extracted Adversary Information .....	11
Evidence of Bruteforcing .....	12
Attempts of Changing Password .....	12
Downloaded Files .....	13
Discussion/Analysis .....	14
IP Address Analysis .....	14
Downloads Analysis .....	16
Username and Password Analysis .....	17
Command Analysis .....	19
Common patterns and methods .....	20
Conclusion .....	21
Appendices .....	22
Extra Code .....	22
Figures .....	22
Tools Used .....	23
References .....	23

## Introduction

For this investigation, I will aim to acquire attack data from my own honeypot and discuss key points such as common attack methods, how the attackers gained access to the network and how we can overcome these weak points. To do this, I will use common data carving tools such as regex to reduce the data collected and focus on highlighting important points and areas, this will give an overall view on the attacks and risks the client may be most susceptible to.

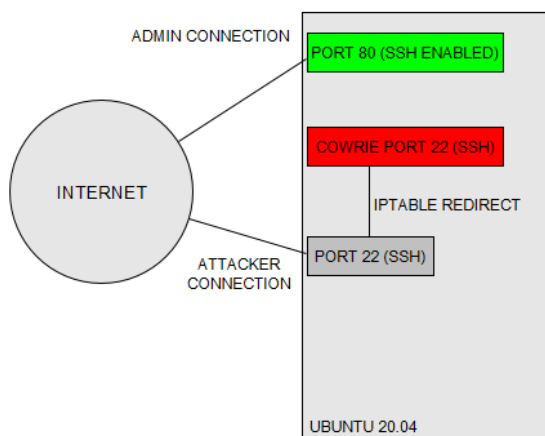
### What are honeypots?

Honeypots are systems purposefully created with the intent to trick attackers by resembling a genuine, vulnerable system over the network. These systems are designed to make attackers want to exploit them by being setup with files which have enticing names, users who seem real and exploitable ports open such as port 22.

### How does a honeypot work?

Honeypots provide a point within the network which can be used as bait to initially isolate an attacker from the main network. This design is used to prevent harm coming to the network and devices connected to it but is not only limited to this as it is also used to analyse the attacker's actions once they are caught, providing threat intelligence (*What Is a Honeypot? How Honeypots Help Security*, n.d.).

## Honeypot Design



*Figure 1 – Network Design*

To gather as accurate as possible threat data, I will host the honeypot on a digital ocean server, there for it will be on a separate network and will not run the risk of attackers gaining access to the clients' network. I will also be utilising and configuring the Cowrie honeypot, a honeypot specifically made for the observation of attacker's behaviour(Oosterhof, 2022).

Two approaches will be used to gather data, first I will disable the cowrie password, this is so attackers are able to instantly gain access and secondly, I will enable the password so attackers will need to exploit to gain access. By using this approach, I hope to gain valuable data on the types of attacks used once the attacker has access to the system and how attackers attempt to gain access.

To ensure the investigation is as accurate to a real network as possible, I will alter the ports within the digital ocean server. This will allow me to allocate port 22 (SSH) to the honeypot which will create a more realistic environment; alongside a name change of the honeypot to make it less suspicious to the attacker. During this investigation, stealth is of the utmost importance as it is required to ensure the attackers stay in the honeypot and attempt to access the system's files or escalate privileges; if the attackers leave too quickly, there will be no to very minimal attack data for analysis.

Cowrie in and of itself is a medium to high interaction honeypot made for the analysis of brute force and shell interaction (Oosterhof, 2022). As standard cowrie is setup as medium interaction which is how it will be used in this investigation. This will present attackers with full file system to navigate and allow for the ability to log user actions and securely capture downloaded files by attackers.

# Honeypot Implementation

## Server Creation

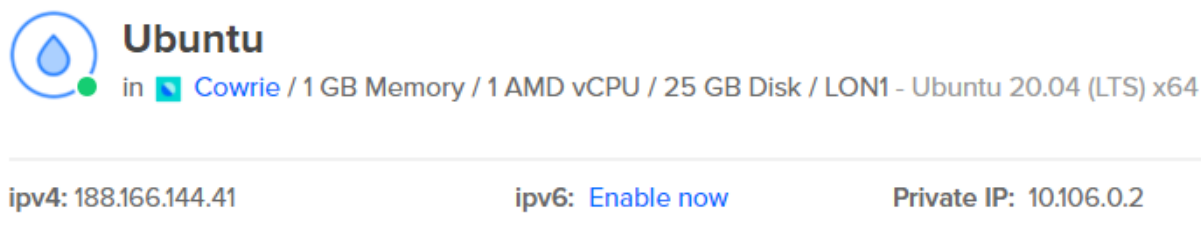


Figure 2 – Ubuntu Virtual Machine | Digital Ocean

Shown in Figure 2, the digital ocean server contains a simple, up to date Ubuntu virtual machine (version 20.04), hosted on a server with 1GB of Memory, an AMD CPU and 25GB storage, enough to store the logs and downloads on the server and run multiple SSH connections at one time. This server is hosted on the main Digital Ocean London server with ports 80 and 22 open, 22 which will be accessible to attackers, 80 which is used to access root on the Linux server with the password oN1on1963#a as per the network design, Figure 1.

## Firewall Configuration

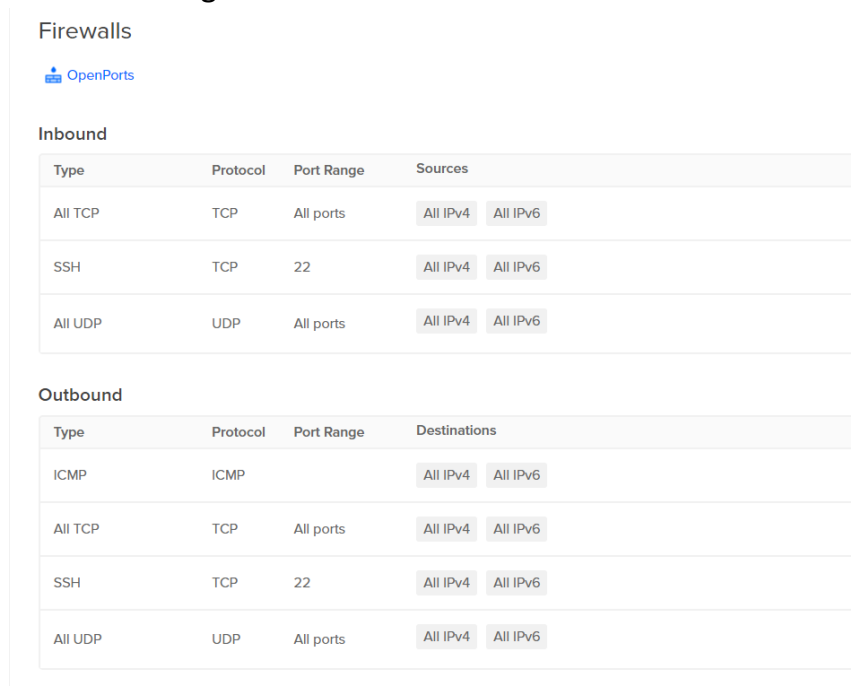


Figure 3 – Digital Ocean Firewall Configuration

Within the digital ocean interface, I setup a new firewall configuration, Figure 3, opening all TCP and UDP ports and port 22 for inbound and outbound connections, then assigned this configuration the Ubuntu server. This will allow attackers to connect to connect to the server. Once the server was setup, I created my new user (steve) then installed the cowrie dependencies from GitHub: 'git clone <https://github.com/jpm-code/forin>'. I then changed into my honeypot user (steve) and cloned the cowrie honeypot: 'git clone <https://github.com/cowrie/cowrie>'.

## Port Configuration

By default cowrie runs on port 2222 due to it not having permission to listen on any port under 1024, but to ensure the investigation results are as realistic as possible once cowrie was installed, I altered this using authbind to port 22 by editing the cowrie.cfg file following section 5.7.2 of the cowrie manual (Oosterhof, 2022)

## Username and Password Configuration

```
# Example userdb.txt
# This file may be copied to etc/userdb.txt.
# If etc/userdb.txt is not present, built-in defaults will be used.
#
# ':' separated fields, file is processed line for line
# processing will stop on first match
#
# Field #1 contains the username
# Field #2 is currently unused
# Field #3 contains the password
# '*' for any username or password
# '!' at the start of a password will not grant this password access
# '/' can be used to write a regular expression
#
root:x:!root
root:x:!123456
tomcat:x:!
oracle:x:!
```

Figure 4 – Standard userdb.txt file

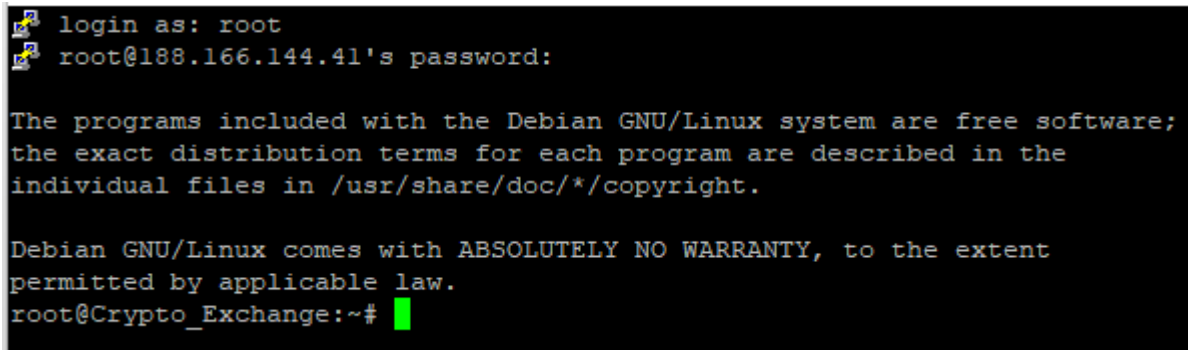
The userdb file contains the username and password combinations for the honeypot. Figure 4 shows the combinations for the cowrie server without a password. This will allow anyone to access the honeypot with the username 'root' and any password. By implementing this I hope to gain a vast amount of data on how attackers act once they gain access to a system, this cuts down the time of attacker's brute forcing and attempting to attack through different methods.

```
# Example userdb.txt
# This file may be copied to etc/userdb.txt.
# If etc/userdb.txt is not present, built-in defaults will be used.
#
# ':' separated fields, file is processed line for line
# processing will stop on first match
#
# Field #1 contains the username
# Field #2 is currently unused
# Field #3 contains the password
# '*' for any username or password
# '!' at the start of a password will not grant this password access
# '/' can be used to write a regular expression
#
root:x:!root
root:x:!123456
root:x:iloveyou
root:x:localhost1
```

Figure 5 - userdb.txt file containing username and password combination for honeypot

Figure 5 then shows the configured password 'iloveyou' which I chose from the rockyou.txt file, a popular file used for brute forcing and 'localhost1' a standard password used for systems. These passwords will be implemented after sufficient data has been gathered commands attackers use.

## Working Honeypot

A terminal window showing a login sequence. The prompt is 'login as: root'. The user enters 'root@188.166.144.41's password:'. The terminal displays the MOTD (Message of the Day) for Debian GNU/Linux, stating that the programs are free software and that the distribution terms are described in individual files in /usr/share/doc/\*/copyright. It also states that Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY. The prompt then changes to 'root@Crypto\_Exchange:~#', indicating the hostname has been changed to 'Crypto\_Exchange'.

```
login as: root
root@188.166.144.41's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@Crypto_Exchange:~#
```

*Figure 6 – Working Honeypot with Server name change*

Figure 6 shows the honeypot working as expected, displaying the hostname (Crypto\_Exchange) and MOTD that will be shown to the attacker after they log into the honeypot. The MOTD is the default that is setup with cowrie. The hostname is chosen during the setup process of the honeypot, in this case I chose Crypto\_Exchange as crypto currency, i.e., bitcoin and Ethereum, are currently very popular and this may trick the attacker into spending more time trying to gain access to information on the server.



## Results

In this section I will extract the results which are collected from a 10-day period between the 14<sup>th</sup> of March 2022 and the 24<sup>th</sup> of March 2022 (Prior to honeypot password change). These results include the number of attacks, username and password attempts, IP addresses and countries regions of the attackers. Logs from 25<sup>th</sup> of March 2022 to 3<sup>rd</sup> of April 2022 will be used for attempts to login to the honeypot, this can include shell access, brute forcing, SSH keys and attempts to change the password.

### Server Interaction

```
steve@Ubuntu:~/cowrie$ grep attempt var/log/cowrie/cowrie.log* | egrep -o '2022-[0-9]+-[0-9]+' | uniq -c
  394 2022-03-25
  518 2022-03-14
 1007 2022-03-15
   857 2022-03-16
   813 2022-03-17
  1090 2022-03-18
  1074 2022-03-19
  1422 2022-03-20
  2426 2022-03-21
  1342 2022-03-22
  1428 2022-03-23
  2976 2022-03-24
steve@Ubuntu:~/cowrie$
```

*Figure 7 - Interactions/attempts to access server 2022-03-14 – 2022-03-24*

As can be seen in figure 7, I used 'grep' to find "attempt" within all logs, 'egrep' to find the date of each file and 'uniq -c' to count how many times attempt was used per date. By using this command, I found between 2022-03-14 and 2022-03-24 (10-day period) there were 14,953 attempts to access the server.

```
  6612 2022-03-25
 14940 2022-03-26
 13732 2022-03-27
 20054 2022-03-28
 18056 2022-03-29
 19980 2022-03-30
   7653 2022-03-31
 18860 2022-04-01
   5985 2022-04-02
 14296 2022-04-03
```

*Figure 8 – Interactions/attempts to access server 2022-03-25 – 2022-04-03*

After changing the password, I used "grep" and "egrep" again to count how many attempts were made to access the server within the second 10-day period between 2022-03-25 – 2022-04-03. Figure 8 shows there were 140,168 interactions with the server where adversaries attempted to gain access.

## Extracted Usernames and Passwords

```
root
root
arnabd
root
root
root
root
xjm
debashree
botsinus
```

*Figure 9 – Extracted Usernames from log files*

```
abcoeur
Ty2t1hv3oC
123456
SYLVAIN
cnlink-sh
STELLA
Qwerty123
123456
123456
botsinus
PUPUCE
NOISETTE
```

*Figure 10 – Extracted Passwords from log files*

With the use of both the `grep` and `awk` commands, Figure 22, I extracted all usernames and passwords from the first 10-day period, Figures 9-10. Each login attempt has the username and password stored within apostrophes, using `awk` I set the delimiter to this and extracted these lists separately.

```
steve@Ubuntu:~/cowrie/var/log/cowrie$ grep attempt cowrie.log* | awk -F "'" '{print $2, $4}'
root abcoeur
root Ty2t1hv3oC
arnabd 123456
root SYLVAIN
root cnlink-sh
```

*Figure 11 – Extracting Username and Password Combinations*

Extracting the combined username and passwords, Figure 11, to find the most used combination and pairing this with statistics from Figures 9-10 should provide me with a solid base of critical passwords to avoid.

## Extracted Adversary Information

161.35.38.36	United States	New Jersey	North Bergen	DIGITALOCEAN-ASN
66.25.175.139	United Kingdom	Portsmouth	Southsea	Virgin Media Limited
128.139.74.173	Singapore		Singapore	DIGITALOCEAN-ASN
144.22.251.63	Brazil	Sao Paulo	Sao Paulo	ORACLE-BMC-31838
222.100.69.36	South Korea	Incheon	Incheon	Korea Telecom
112.65.42.128	China	Shanghai	Shanghai	CHINA UNICOM China163 Backbone
112.65.42.87	China	Shanghai	Shanghai	CHINA UNICOM China163 Backbone
146.195.137.24	Netherlands	North Holland	Amsterdam	DIGITALOCEAN-ASN
128.139.145.20	Singapore		Singapore	DIGITALOCEAN-ASN
165.154.46.18	Hong Kong	Central and Western District	Central	UCloud INFORMATION TECHNOLOGY HK LIMITED
142.33.161.168	Germany	Hesse	Frankfurt am Main	DIGITALOCEAN-ASN
194.226.43.21	Russia	Samara Oblast	Tolyatti	Zty Corp LLC
76.108.103.69	United States	Florida	Port Saint Lucie	COMCAST-1982
122.134.229.54	China	Shanghai	Chaichang	CHINA UNICOM China163 Backbone
46.13.133.18	Switzerland	Zurich	Zurich	Private Layer INC
46.146.218.79	Russia	Perm Krai	Perm	JSC ER-Telecom Holding
52.205.24.173	France	Bas-Rhin	Strasbourg	East Europe GmbH
106.12.152.36	China			Beijing Baidu Netcom Science and Technology Co., Ltd.
112.65.42.83	China	Shanghai	Shanghai	CHINA UNICOM China163 Backbone
175.126.176.21	South Korea	Seoul	Songpa-gu	SK Broadband Co Ltd
80.31.91.122	Spain	Valencia	Canals	COGENT-174
31.27.105.101	Italy	Provincia di Rieti	Cittaducale	Vodafone Italia S.p.A.
201.87.151.166	Brazil	Sao Paulo	Campanas	Megatelecom Telecomunicacoes Ltda
162.241.115.39	United States			UNIFIEDLAYER-AS-1
190.187.218.41	Peru			AMERICATEL PERU S.A.
212.127.95.129	Poland	Lower Silesia	Wroclaw	Korbank S. A.
123.212.190.62	South Korea	Chungcheongnam-do	Gongju	SK Broadband Co Ltd
133.136.116.64	China			Yunify Technologies Inc.
103.163.187.24	Netherlands	Groningen	Groningen	RoyalHostings B.V.
165.22.55.35	Singapore		Singapore	DIGITALOCEAN-ASN
128.139.225.7	Singapore		Singapore	DIGITALOCEAN-ASN
182.253.73.196	Indonesia	East Java	Surabaya	BIZNET NETWORKS
43.154.75.200	Hong Kong	Central and Western District	Central	Tencent Building, Kejizhongyi Avenue
192.116.113.241	Israel	Tel Aviv	Tel Aviv	Partner Communications Ltd.
200.207.95.20	Brazil	Sao Paulo	Ribeirao Pires	TELEFONICA BRASIL S.A
61.95.174.240	India			BHARTI Airtel Ltd.
122.134.229.31	China	Shanghai	Chaichang	CHINA UNICOM China163 Backbone
61.177.172.92	China	Shanghai	Baochuan	Chinanet
21.76.86.223	Vietnam	Ho Chi Minh	Ho Chi Minh City	Viettel Group
21.76.86.108	Vietnam	Ho Chi Minh	Ho Chi Minh City	Viettel Group
61.177.137.133	China	Jiangsu	Wuxi	Chinanet
127.0.0.1	Reserved space			
40.68.203.216	Netherlands	North Holland	Amsterdam	MICROSOFT-CORP-MSN-AS-BLOCK
167.71.234.157	India	Karnataka	Bangalore	DIGITALOCEAN-ASN
187.105.40.231	Brazil	Ceara	Fortaleza	CLARO S.A.
132.210.206.18	United States			AS-COLOCROSSING
129.159.138.48	Israel	Jerusalem	Jerusalem	ORACLE-BMC-31838
3.129.150.66	United States	Ohio	Columbus	AMAZON-02
40.77.40.108	United States	Iowa	Des Moines	MICROSOFT-CORP-MSN-AS-BLOCK
167.71.220.220	Singapore		Singapore	DIGITALOCEAN-ASN
37.139.5.94	Netherlands	North Holland	Amsterdam	DIGITALOCEAN-ASN
159.65.155.55	India	Karnataka	Bangalore	DIGITALOCEAN-ASN
112.65.42.229	China	Shanghai	Shanghai	CHINA UNICOM China163 Backbone
101.32.36.205	Hong Kong	Central and Western District	Central	Tencent Building, Kejizhongyi Avenue
112.65.42.15	China	Shanghai	Shanghai	CHINA UNICOM China163 Backbone
45.83.66.244	Germany			Alpha Strike Labs GmbH
116.21.204.72	China	Guangdong	Guangzhou	Chinanet
122.134.229.41	China	Shanghai	Chaichang	CHINA UNICOM China163 Backbone
164.30.217.153	Germany	Hesse	Frankfurt am Main	DIGITALOCEAN-ASN
176.128.73.254	United States	California	Santa Clara	DIGITALOCEAN-ASN
121.4.187.6	China			Shenzhen Tencent Computer Systems Company Limited
139.59.25.164	India	Karnataka	Bangalore	DIGITALOCEAN-ASN
103.133.42.55	Vietnam	Hanoi	Hanoi	Nguyen Ngoc Thanh Trading Limited Company
139.59.88.197	India	Karnataka	Bangalore	DIGITALOCEAN-ASN
154.17.16.136	Belarus	Minsk City	Minsk	Mobile TeleSystems JLLC
20.106.169.99	United Kingdom	England	London	MICROSOFT-CORP-MSN-AS-BLOCK
163.197.40.112	United States	Illinois	Chicago	Gigabitbank Global
61.148.56.158	China	Beijing	Banqiao	China Unicom Beijing Province Network
165.232.116.79	Germany	Hesse	Frankfurt am Main	DIGITALOCEAN-ASN
114.67.68.255	China			China Telecom Group
43.132.156.75	Hong Kong	Central and Western District	Central	Tencent Building, Kejizhongyi Avenue
202.170.57.253	Malaysia	Penang	George Town	Universiti Sains Malaysia USM
103.235.192.21	Azerbaijan			Azertelecom LLC
169.254.147.83	Mexico	San Luis Potosí	San Luis Potosí, CI	Uninet S.A. de C.V.
101.43.31.99	China			Shenzhen Tencent Computer Systems Company Limited
23.100.2.154	Netherlands	North Holland	Amsterdam	MICROSOFT-CORP-MSN-AS-BLOCK
103.221.252.46	Bangladesh			university of dhaka
87.37.76.214	Hungary	Budapest	Budapest	Invitech ICT Services Kft.
45.67.34.253	Russia			Chernyshev Aleksandr Aleksandrovich
5.188.61.118	Russia			OOO Network of data-centers Selectel
115.134.130.53	Malaysia	Kedah	Changloun	TM Net, Internet Service Provider
128.139.123.0	Singapore		Singapore	DIGITALOCEAN-ASN
139.59.116.3	Singapore		Singapore	DIGITALOCEAN-ASN
187.161.222.175	Mexico	Nuevo León	Monterrey	Televisión Internacional, S.A. de C.V.
119.28.29.14	Hong Kong	Central and Western District	Central	Tencent Building, Kejizhongyi Avenue
103.142.14.205	Indonesia	East Java	Jombang	Pemerintah Kabupaten Jombang
118.63.55.101	Vietnam	Ho Chi Minh	Ho Chi Minh City	FPT Telecom Company
112.196.62.36	India	Telangana	Hyderabad	Guadrant Televentures Limited
31.1.27.184	Germany	Bavaria	Adelschlag	Deutsche Telekom AG
59.63.205.47	China	Jiangxi	Nanchang	CHINANET Jiangxi province IDC network
182.253.28.125	Indonesia	Jakarta	Jakarta	Biznet ISP
79.31.30.190	France	Upper Savoy	Annecy	SFR SA
34.110.210.233	Belgium	Antwerp Province	Aartselaar	Orange Belgium SA
122.134.229.41	China	Shanghai	Chaichang	CHINA UNICOM China163 Backbone
62.201.207.53	Iraq			IQ Networks

Figure 12 - Attacker IP's, Countries, Regions, Cities, and ISP's (duplicates removed)

Figure 12, shows the cumulative data collected from the honeypot logs of attacker's information with duplicate IP addresses removed to provide a broader view of the unique IP's which attacked the server and their location. These will be further processed to show the frequency of unique attacks and frequency of attacks from the same address/country.

## Evidence of Bruteforcing

admin 000000	root 000000	user 000000	user2 000000
admin 1	root 1	user 1	user2 1
admin 1111	root 1111	user 1111	user2 1111
admin 111111	root 111111	user 111111	user2 111111
admin 12	root 12	user 12	user2 12
admin 121212	root 121212	user 121212	user2 121212
admin 123	root 123	user 123	user2 123
admin 123!@#qwe	root 123!@#qwe	user 123!@#qwe	user2 123!@#qwe
admin 123123	root 123123	user 123123	user2 123123
admin 123321	root 123321	user 123321	user2 123321
admin 1234	root 1234	user 1234	user2 1234
admin 12345	root 12345	user 12345	user2 12345
admin 123456	root 123456	user 123456	user2 123456
admin 1234567	root 1234567	user 1234567	user2 1234567

*Figure 13 - Evidence of brute forcing using wordlists*

Shown in figure 24, I used a combination of grep and awk to grab contents relating to a specific attacker from the cowrie.log file. Figure 13 displays the results of this and suggests that the attacker was using two wordlists, a username wordlist containing the usernames 'admin, root, user and user2' and then a password wordlist containing commonly used passwords including 'iloveyou' the password I extracted from rockyou.txt explicitly to capture brute forcing methods.

The reason this suggests brute forcing, using software such as hydra, is when attempting to access the server, the attacker enters each username systematically along with the same password before moving onto the next password.

## Attempts of Changing Password

```
CMD: echo "root:6H2KmKLXziNi"|chpasswd|bash
Command found: bash

] CMD: echo "root:3shQE0a8cK03"|chpasswd|bash
] Command found: bash

CMD: echo "root:tuNLNk1TSo0j"|chpasswd|bash
Command found: bash

CMD: echo "root:KnYQqIOfBJT7"|chpasswd|bash
Command found: bash
```

*Figure 14 - Attempts to change password of root user*

Post enumeration, one common technique adversaries would use to attempt to gain access to the server is changing the password which, shown in figure 14, multiple attempts have been made to change the password of the root user using bash.

## Downloaded Files

```
root@Ubuntu:/home/steve/cowrie/var/lib/cowrie/downloads# ls
01a37e80a982b9a82e9d8dd19fec64841ef54dd22ec599e7f304170a54fb067  a33784dbee5d59cc3a407af31910fb538d34e51253699bfc2cb5a278f26511bd
0243be7c00e557ef65394015c15dac5023d4e37d7ec0d10ccf333f1be7c57fe5  a41c3ce67c7e011e7295e357048181472ca8d6863845d1280c512486037eafe9
06e4ae105237c80ec9402bcee919912e17006c532c7e88fcbc76264b433e16f4  a8460f446be540410004b1a8db4083773fa46f7fe76fa84219c93daa1669f8f2
0b577c65228a3fc01362085e8ac6915a99ce5316b6dc2b013141c70626391d76  a877649f7d498125c8c9646c376d3c176444798c9b9a0e3d1f625aefc7ad2617
0e9e4e3eebbe87edaa27370f2c33e6bfc7ced9b3df876edbdc0022e045ea0cd  aaaa02832476ac1459e2fe00e29c56bf9e147e9fa4578a0dadedf75ce156faeec
1553a94ec17c969624b6ddb600ed97503116bcdcc8e99be4cb8082b79450c734  ac846e1b13e65a1f04135c0a865bf7140d8952796ded7981963f66428166f77a
15e9aa8ed2b551569eae5b46c5907dd8ea4758c047e3f15cd143945d5072499a  ac9674feb8f2fad20c1e046de67f899419276ae79a60e8cc021a4bf472ae044f
1a526fe7b74ec36ef2facd3588e12b6acbd9c205bd224f7a1d7c54153c2afec  adc675f971f8225092ea92d4f3237e9e0ef31c5d953d5e15e06572984431b26f
24e348e7416c6a38805dd5f148a962ca5a1c1786347fd27005fc20e50cab9cb2  af9e68205295634dd5f5cfe8f8cfa62f478ee4b804ae3f521c8f5a68b0c82b9
28a0b2e9ce7bfa8d8c7be0ac8a9593898c6fda1a319c873b08220a43e68d0ea4  b184a5afa10a11431431ba676c3f5626152ffc0fb150309aad58e8a2a2b3ba3
2acd93cf8068a2d6a71866af6588e821985d4fc86a5c1196eb09ba3e373d203  b64ae7917dd5bec132faebe1936cb6d4b3e0049d2d1648dacf92a2eb3c1c2513
2b61b9f4549b84a370579d0fc67f91fe2826968eb286dbe31457c1b6c6843b45  be54efa3b2fa2d1ec9215c9f0c02f4626f727c16e2ed04bdced186e251324c68
2c50564f7710d3c1963da5a93ec09c77fb7e49b59df7df29cd558b5e989c3f2e  c479ec459b7f93da7c9b0e4c0fe2913eb98f3e2c109bcbdb598a8e7a16633c6a4
2df79021148e598a41d7928f72dc0239684aa65a5b974ef3f7f81874a88c2fff  c72b9d9036131144be4e46813c5c497305dab9e11678500bf56cb03558cb0e9f
362871c9c74c82add6ecf0f3a6341f749c0f30d3d02cbf5d53b75a95548915bf  d2a704bff4eba80918f304d701ee6fa9dd1252886e7b75c540f003fba53afde8
36688cc052469c03b9af86432b28f9b769f6dc8cf5744d5b986900816ed8edbb  de1e1ad41b8639490e9789eb530f88c146f283129711a02eeeb76fb3d521a14
379cbfffd4e13a02a4662cbaf4731feeee2a0f7d9e51db054f8233a85222ba1f  e1789a5cd44d53a3f86eb05e1446dec7770ae175e9bfc86d4cfffbb6d07443989
3b0285d601232ddee79a28f3923462f3e4f8c6cedb809377ed0da07ba06b651e  e3b0c44298f1c149afb4c8996fb92427ae41e4649b934ca95991b7852b855
4355a46b19d348dc2f57c046f8ef63d4538ebb936000f3c9ee954a27460dd865  e6acac4178b06f9ef5439d4f65e1ec8c289de48660809f60f2683f8f42d210
45c6de4d5b1397876197c510a78b1966fde5a2916d96ce7129e730705c423079  e76e66d5ede838d312d155e9c6ad91adfe630f28f4fd3f0af6eeec14ea6d8d76
462fc0c39bf1c30bf063b5c3e06fda69cfbf612dc123301d2edd31ad9b3f41b1  fb771b83734e19786a6ae0df67bfc0bf5d6c8b61e28023d586b16d451f7fe38b
469aa49f4f628498111af193d9220fcc41825d94525246656e40b0560d4cd267  ff6f81930943c96a37d7741cd547ad90295a9bd63b6194b2a834a1d32bc8f85d
```

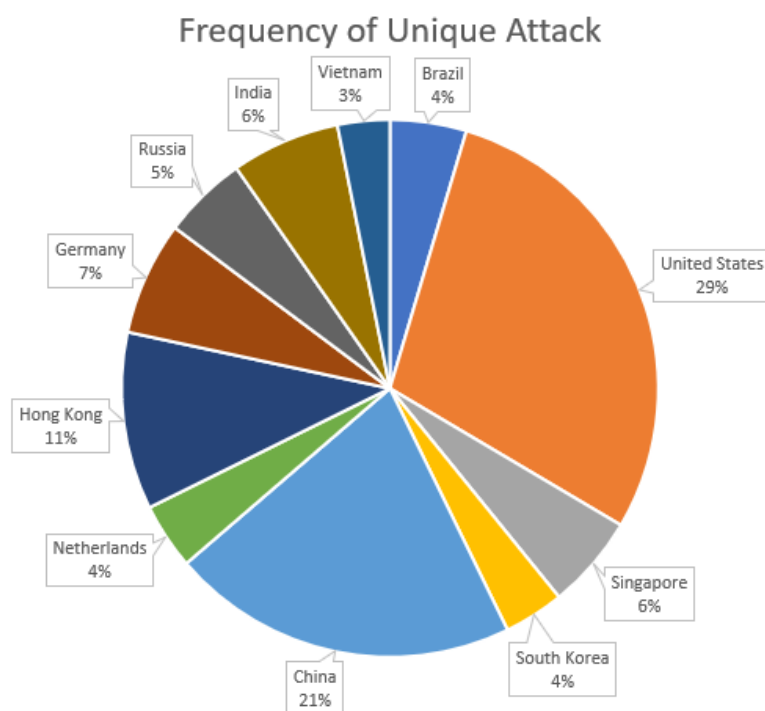
Figure 15 - Downloaded files within the honeypot downloads folder

Changing directory to /lib/cowrie/downloads and using the ls command to display the contents, figure 15 covers the honeypot downloads folder displaying a list of files which have been downloaded within the honeypot. These contents can be analysed further by software to detect what types of files they are as these files could be malicious, i.e., trojans, rootkits or keyloggers.

## Discussion/Analysis

In this section I will take a closer look at the results focusing more on the analytical side of the data that has been collected. I will also discuss how attackers gained access to the honey pot, i.e., what ports and attacks were used. I will also focus on common patterns and methods.

### IP Address Analysis



*Figure 16 - Frequency of Unique attack (Qualitative Data)*

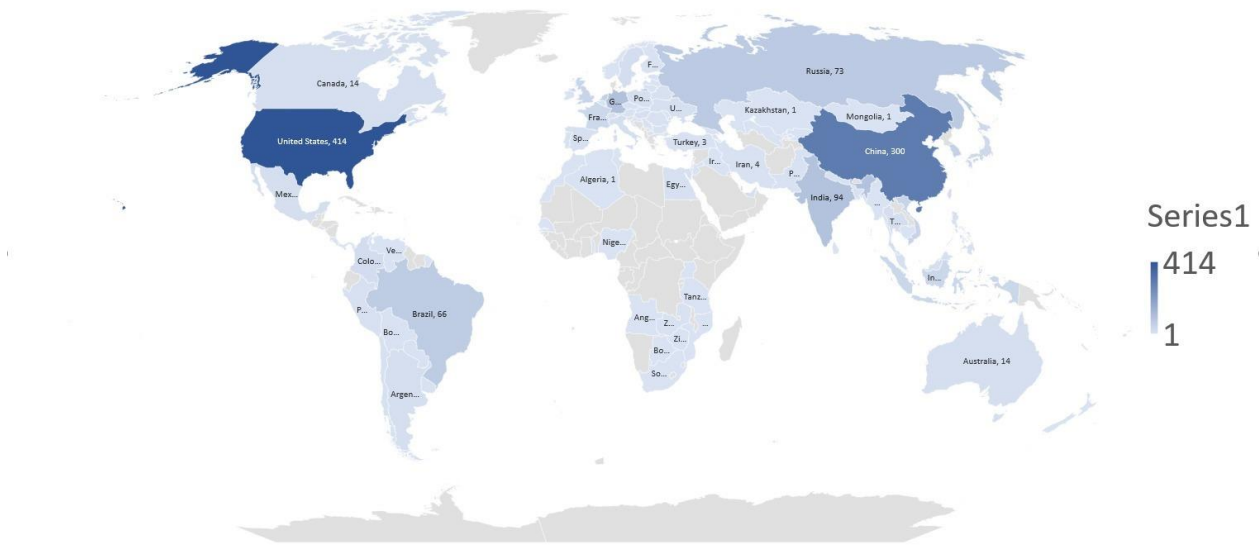
In relation to Figure 12, Figure 16 displays the frequency of a unique attack by country based on the attackers IP addresses. Once the results were gathered from the honeypot, I exported these using FileZilla and imported them into ip2geo to automate the geolocation of the IP addresses, shown in figure x.

This data was then imported into excel where I:

- Copied and removed duplicate data.
- Implemented COUNTIF() function to count how many times a country appears in the original list.
- Filtered results by top 10
- Produced qualitative data pie chart

This finalised data, figure 16, displays that: United states, China and Hong Kong are the top 3 attackers according to the dataset.

## Frequency of Unique Attack



*Figure 17 - Frequency of Unique Attack (Quantitative Data)*

Similarly, to producing the results in figure 16, I used the same process results in figure 17 but did not remove duplicate data. This attack map shows the number of standalone attacks from each country there was during the 10-day period. Although these attacks are not unique and some from the same IP, each attack is a new connection with a “new” connection.

## Downloads Analysis

AhnLab-V3	① Shell/ElfDownloader.S1	Antiy-AVL	① Trojan.Generic.ASMalwS.352E78F
Arcabit	① Application.Linux.Generic.D1712	Avast	① BV:Downloader-AAN [Drp]
Avast-Mobile	① ELF:Mirai-UM [Trj]	AVG	① BV:Downloader-AAN [Drp]
Avira (no cloud)	① LINUX/Agent.SH.B	Baidu	① Multi.Threats.InArchive
BitDefender	① Generic.Bash.MiraiA.9D6F9F94	BitDefenderTheta	① AtPacker.FC7AD9D11F
CAT-QuickHeal	① Trojan.Shell.Downloader.39008	ClamAV	① Win.Trojan.IRCBot-785
Comodo	① Malware@#m213rp1z6qqw	Cylance	① Unsafe
Cynet	① Malicious (score: 99)	Cyren	① SH/Mirai.A.gen!Camelot
DrWeb	① Linux.DownLoader.700	Elastic	① Malicious (moderate Confidence)
Emsisoft	① Generic.Bash.MiraiA.9D6F9F94 (B)	eScan	① Generic.Bash.MiraiA.9D6F9F94
ESET-NOD32	① Multiple Detections	F-Secure	① Malware.LINUX/Agent.SH.B
Fortinet	① Linux/Agent.SHS!tr.dldr	GData	① Linux.Trojan.Mirai.J
Gridinsoft	① Ransom.Win32.Sabsik.sa	Ikarus	① Trojan-Downloader.Linux.Morla
Jiangmin	① RiskTool.Linux.bkh	K7AntiVirus	① Trojan ( 005715eb1 )
K7GW	① Trojan ( 005715eb1 )	Kaspersky	① Backdoor.Perl.IRCBot.ml
Lionic	① Trojan.Linux.Agent.mtc	Malwarebytes	① Spyware.PasswordStealer
MAX	① Malware (ai Score=80)	MaxSecure	① Trojan.Malware.140219931.susgen
McAfee	① Linux/Downloader.w	McAfee-GW-Edition	① Linux/Agent.a
Microsoft	① TrojanDownloader.Linux/Morla!MTB	NANO-Antivirus	① Trojan.Script.Downloader.fjajs
Panda	① Trj/GdSda.A	Rising	① Backdoor.IRCBot/BASH!t.CC7E (CLASSIC...
Sangfor Engine Zero	① Virus.Generic-Script.Save.ba1	SentinelOne (Static ML)	① Static AI - Malicious Archive
Sophos	① Mal/PerlBot-A	Trellix (FireEye)	① Generic.Bash.MiraiA.9D6F9F94
TrendMicro	① ELF_MIRAILOD.SM	TrendMicro-HouseCall	① ELF_MIRAILOD.SM
VBA32	① BScope.Trojan.Eb	VirIT	① Linux.DownLoader.ZO
Zillya	① Trojan.Agent.Win32.2688266	ZoneAlarm by Check Point	① HEUR:Backdoor.Perl.IRCBot.mo

**Figure 18 - Malware detected within download file**

With the main purpose of this investigation being capturing attacks it's important that the downloads folder is analysed to discover what files/applications attackers are downloading onto the system. Shown in figure 15 is the downloads folder listed in an SSH window of the system. This view does not give the name of any of files but rather some jumbled letters.

Process to analyse downloads folder:

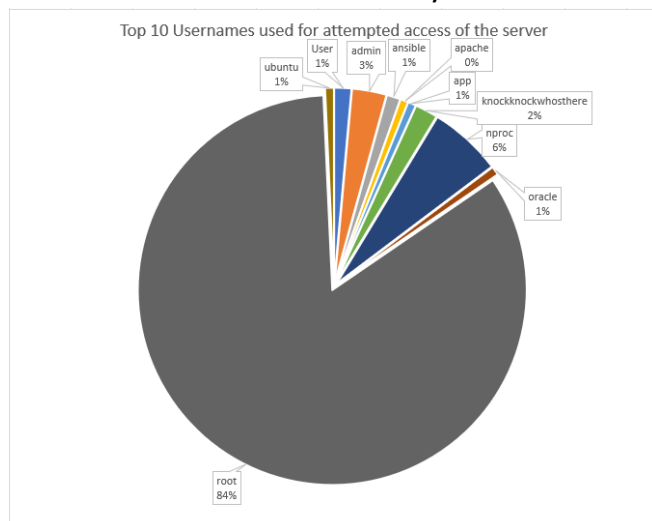
- Connect to honeypot via FileZilla
- Download the /downloads directory in secure environment (i.e., virtual machine)
- Upload folder to (Geolocation Lookup Tool, n.d.) for analysis of files.

Following the results of the download folder analysis, figure 18 confirms multiple downloads containing content such as trojans, malware and bash files. Trojans are executables disguised as commonly used programs or installers which when run, install a process or application to exploit the system. These types of files can be used to gather user data or hide executable code, i.e., bash executables, to gain root access to a system.

The fact that many of these types of files have been downloaded to the system show that users could be installing malware to attempt to gain root access on the server or gather user data which would also be reasonable as, due to the server's name 'Crypto\_Exchange' the attacker may think there is valuable data on the server.



## Username and Password Analysis



*Figure 19 - Top 10 usernames used for attempted access of a server*

Using the data gathered and shown in figure 9, I used Microsoft Excel to narrow down username attempts when attacking the server which resulted in the following top 10 usernames also shown in figure 19:

- root
- nproc
- admin
- knockknockwhosthere
- user
- ansible
- ubuntu
- oracle
- app
- apache

This data is gathered from counting each time the result is within the list of usernames gathered from the logs on the server. From this data, the most common username to use and attempt to gain access to a server is root, which is the standard username given to Linux systems.

Top 10 Passwords used for attempted access of the server

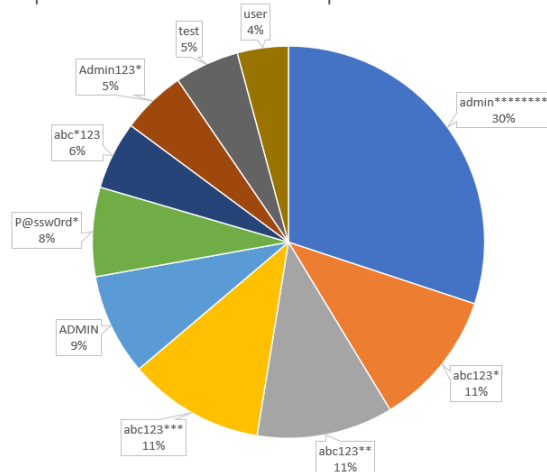


Figure 20 - Top 10 passwords used for attempted access of a server

Using a similar process for the usernames in figure 19, I gathered the password data, Figure 10, and processed them to display the top 10 passwords used for attempted access of the server as shown in figure 20:

- admin\*\*\*\*\*
- abc123\*
- abc123\*\*
- abc123\*\*\*
- ADMIN
- P@ssw0rd\*
- Abc\*123
- Admin123\*
- Test
- User

This data shows the most common password being 'admin\*\*\*\*\*'.

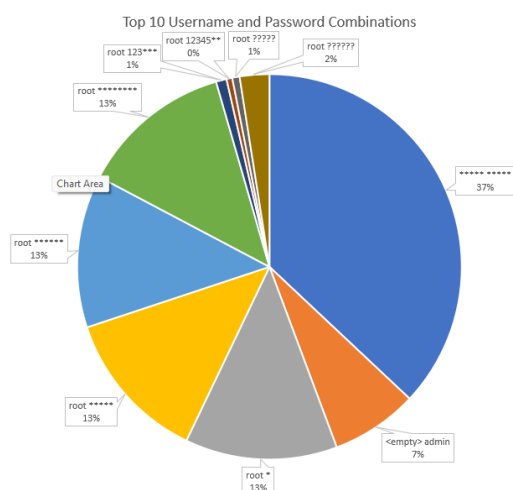


Figure 21 – Top 10 Username and Password Combinations

Further looking at the combined usernames and passwords adversaries used when attempting to access the server, it is clear from within the top 10, Figure 21, root is a very popular username also shown in figure 19. Similarly, to the data gathered in Figure 20, this displays some similarities, with the most popular entries being combinations of numbers, asterisks and admin.

## Command Analysis

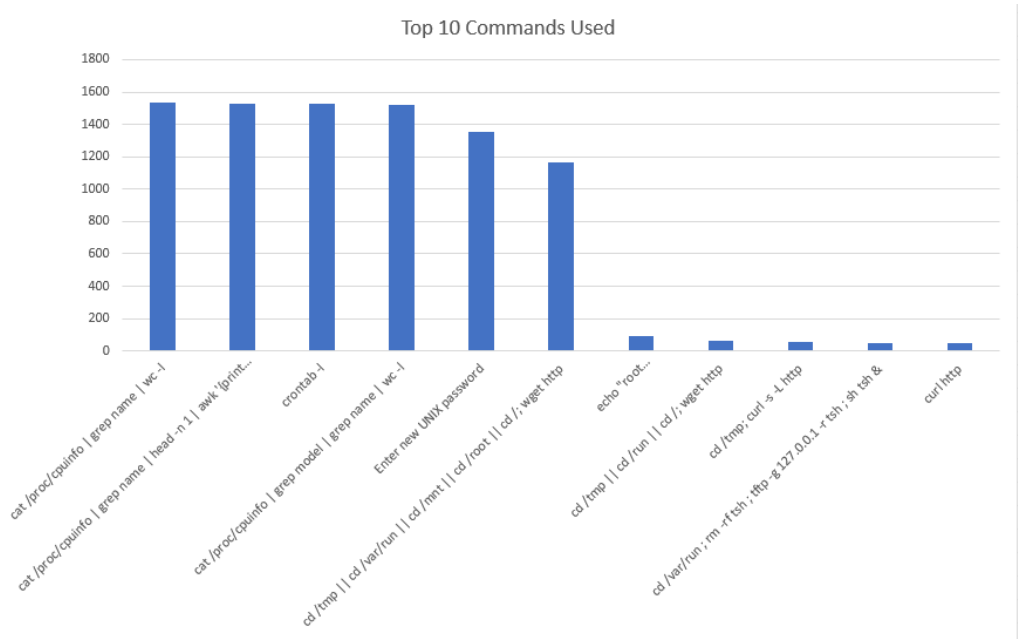


Figure 22 - Top 10 Commands Used

Linked to Figure 18, some of the top 10 commands, shown in Figure 22, used in the system are data gathering commands. Commands such as “cat /proc/cpuinfo” are aimed at enumerating the system to find vulnerabilities adversaries can exploit. Figure 22 also displays “wget” and “curl” commands, these are used to download files to the system, which also links to the downloaded malware in Figure 18.

## Common patterns and methods

Common patterns can be seen in the honeypot log files from the attackers attempting to access the server and from when they are inside the server.

Beginning with then the attackers are attempting to access the server, it's common for them to use two types of methods: brute forcing and changing the password. When attempting to access the server, there are multiple instances within the log files where attackers use brute forcing techniques, this is shown by multiple connections being processed from the same IP address with the same usernames, i.e., "admin, root, admin, root" with a different password each time the username is retried. As shown in figure 13, wordlists are also being used with the potentially identified rockyou.txt wordlist due to the user gaining access via the honeypot password extracted from this wordlist.

Attackers also commonly attempt to change the password of the root user using the 'chpasswd' command and passing it to bash as shown in figure 14. Linking with figure 22, these results show the command "Enter new UNIX password" where adversaries have attempted to trick the server into allowing them to change the password of the user. From these results this command can be seen to have been used close to 1400 times on the server.

Once the attackers gain access to the server, it's common to see attackers begin their enumeration process to try and gather information about the system itself for example username using 'uname -a' and listing cronjobs on the system using 'crontab -l' a popular command used close to 1600 times, shown in figure 22.

## Conclusion

Concluding the findings from this investigation, the honeypot log files from the digital ocean server indicated adversaries are currently using a multitude of approaches to attack the system.

Highlighting their main objective once they are in the system, attackers begin to enumerate the system, finding weaknesses and vulnerabilities which they can exploit to gain further access. This is particularly shown by the most common commands executed. Exploiting these vulnerabilities will give them access to information on the system or achieve their end goal of wanting to become root user.

From the variety of attacks on the system, it shows that attackers are not only interested in gaining root but malware which attackers downloaded to the system display that some attackers also have the intention of gaining data from the server. Many files downloaded to the server, list shown in figure 18, are trojans with antivirus Malwarebytes flagging a file as a “password stealer”, the chosen attack method by attackers as trojans replicate an application for the purpose of tricking the user of the system(Johansen, 2020).

This investigation points to multiple recommendations that can be made to ensure the security of networks and systems regarding username and password combinations, account security and active ports.

Beginning with the username and password combinations that adversaries tried, the results and analysed data, figures 19-20 show attackers beginning attacks with simple, common usernames and passwords such as: root, admin, \*, and 12345. It's advised that the client ensures passwords are strong, i.e., using mixed characters, symbols, and numbers with a minimum length of 8 and usernames are not common or default such as those shown in Figures 19-21.

Unnecessary ports on the server should also be closed. As shown from the investigation, a port such as 22 is easily accessible and is susceptible to different kinds of attacks such as brute forcing if not secured correctly. Closing the ports which are least needed and/or limiting access to certain users, user groups and programs, will greatly reduce the risk of adversaries gaining access to the client server.

File permissions should also be closely considered on the client's server. With the honeypot having an open file structure, attackers could read and write information, however, note that certain directories had file permissions which could not be changed, i.e., the downloads folder which attackers could not execute within. Implementing closely monitored file permissions will help to increase the security of the server in the event an attacker does access the system.

## Appendices

### Extra Code

```
steve@Ubuntu:~/cowrie/var/log/cowrie$ grep attempt cowrie.log* | awk -F '"' '{print $4}' | sort > password.txt
steve@Ubuntu:~/cowrie/var/log/cowrie$ grep attempt cowrie.log* | awk -F '"' '{print $2}' | sort > sername.txt
```

Figure 23 - Exporting individual passwords and usernames from log files using grep and awk

```
steve@Ubuntu:~/cowrie$ bin/cowrie restart
Stopping cowrie...
Using default Python virtual environment "/home/steve/cowrie/cowrie-env"
Starting cowrie: [twistd --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
```

Figure 24 - Restarting Honeypot so change of uname/pass takes effect

```
root@Ubuntu:/home/steve/cowrie/var/log/cowrie# grep 188.166.149.63 cowrie.log.2022-03-28 | grep attempt | awk -F '"' '{print $2,$4}' | sort
```

Figure 25 - Getting login attempt username and passwords from specific IP

### Figures

Figure 1 – Network Design .....	5
Figure 2 – Ubuntu Virtual Machine   Digital Ocean .....	6
Figure 3 – Digital Ocean Firewall Configuration .....	6
Figure 4 – Standard userdb.txt file .....	7
Figure 5 - userdb.txt file containing username and password combination for honeypot.....	7
Figure 6 – Working Honeypot with Server name change.....	8
Figure 7 - Interactions/attempts to access server 2022-03-14 – 2022-03-24.....	9
Figure 8 – Interactions/attempts to access server 2022-03-25 – 2022-04-03 .....	9
Figure 9 – Extracted Usernames from log files.....	10
Figure 10 – Extracted Passwords from log files.....	10
Figure 11 – Extracting Username and Password Combinations.....	10
Figure 12 - Attacker IP's, Countries, Regions, Cities, and ISP's (duplicates removed) .....	11
Figure 13 - Evidence of brute forcing using wordlists .....	12
Figure 14 - Attempts to change password of root user .....	12
Figure 15 - Downloaded files within the honeypot downloads folder.....	13
Figure 16 - Frequency of Unique attack (Qualitative Data).....	14
Figure 17 - Frequency of Unique Attack (Quantitative Data).....	15
Figure 18 - Malware detected within download file .....	16
Figure 19 - Top 10 usernames used for attempted access of a server .....	17
Figure 20 - Top 10 passwords used for attempted access of a server .....	18
Figure 21 – Top 10 Username and Password Combinations.....	18
Figure 22 - Top 10 Commands Used.....	19
Figure 23 - Exporting individual passwords and usernames from log files using grep and awk.....	22
Figure 24 - Restarting Honeypot so change of uname/pass takes effect.....	22
Figure 25 - Getting login attempt username and passwords from specific IP .....	22

## Tools Used

- Ip2geo – Used to automate IP Geolocation.
- Digital Ocean - Used for server hosting
- Putty – Used for connection to server
- Python – Environment used for cowrie and regular expressions
- Cowrie – Honeypot software
- Microsoft excel – Used for processing results from honeypot
- Filezilla – Downloading files from honeypot

## References

- Bulk IP Location Finder - Geolocation Lookup Tool*. (n.d.). Retrieved April 18, 2022, from <https://ip2geo.org/>
- Johansen, A. G. (2020). *What is a Trojan? Is It Virus or Malware? How It Works | Norton*. Norton - Security Centre. <https://us.norton.com/internetsecurity-malware-what-is-a-trojan.html>
- Oosterhof, M. (2022). *Cowrie Documentation*. 1. <https://readthedocs.org/projects/cowrie/downloads/pdf/latest/>
- What is a honeypot? How honeypots help security*. (n.d.). Retrieved April 11, 2022, from <https://www.kaspersky.co.uk/resource-center/threats/what-is-a-honeypot>