Botnet Detection: Honeypots and the Internet of Things

By

Ryan Chinn

MS MIS Candidate (2015)

Eller College of Management

Advisor: Dr. Hsinchun Chen

ACKNOWLEDGEMENTS

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the University of Arizona, or the National Science Foundation (NSF). This material is based upon work supported by the NSF under Grants No. DUE-1303362 and No. SES-131463. I also acknowledge the funding and program support provided by the NSF Scholarship-for-Service (SFS) Program.

TABLE OF CONTENTS

LIST OF FIGURES 6

LIST OF TABLES 7

ABSTRACT 8

1 INTRODUCTION 8

2 LITERATURE REVIEW 10

2.1 Botnets 10

2.1.1 Active Detection Techniques 12

2.1.2 Passive Detection Techniques 13

2.2 Honeypots 14

2.2.1 Malware Honeypots 15

2.2.2 Cloud Honeypots 16

2.3 Internet of Things (IoT) 16

2.3.1 Shodan Search Engine 17

2.3.2 Related Scanning Tools 18

3 RESEARCH GAPS AND QUESTIONS 19

4 RESEARCH TESTBED 19

5 RESEARCH DESIGN 19

5.1 Honeypot Deployment 20

5.1.1 Honeypot Requirements and Selection 20

5.1.2 Network Location 21
5.1.3 Configuration 21
5.1.4 Deployment Duration 21
5.2 Malware Analysis 23
5.3 Device Identification 23
6 RESULTS AND DISCUSSIONS 23
6.1 VirusTotal Results 24
6.1.1 Top Malware 24
6.2 Shodan Results 25
6.2.1 Top Device Types 25
6.2.2 Top 10 Products 26
6.2.3 Top 10 Countries 27
6.2.4 Top 10 Organizations 29
6.2.5 Top 10 Open Ports 30
6.3 Discussion 31
6.3.1 Conficker Worm 31
6.3.2 Malware Bots versus Port Scanning Bots 31
6.3.3 Router Vulnerability Case Study 32
7 CONCLUSION 34
8 REFERENCES 35

9 APPENDIX A: Ports Scanned by Shodan 38

10 APPENDIX B: Dionaea Honeypot Data Dictionary 39

11 APPENDIX C: Shodan Data Dictionary 40

LIST OF FIGURES

Figure 1: Example of Zeus Botnet Source Code Sold on Hacker Forum 9

Figure 2: Example of DDoS Service for Sale that Runs on Botnet of Hacked Home Routers (Krebs, 2015) 11

Figure 3: Illustration of Top Spam-Sending Botnets (Symantec, 2015) 11

Figure 4: Research Design 20

Figure 5: Illustration of Dionaea Network Status after Deployment 22

Figure 6: Illustration of DionaeaFR Dashboard 22

Figure 7: Map of Malware Bot Countries 27

Figure 8: Map of Port Scanning Bot Countries 28

Figure 9: Map of Vulnerable Routers 33

LIST OF TABLES

Table 1: Summary of Prior Botnet Studies 12

Table 2: Examples of Honeypot Software 14

Table 3: Summary of Prior Honeypot Studies 15

Table 4: Summary of Prior IoT Studies 17

Table 5: Research Testbed 19

Table 6: Amazon EC2 Instance Configuration 21

Table 7: VirusTotal Malware Classifications 24

Table 8: Malware Bot Device Types 25

Table 9: Port Scanning Bot Device Types 25

Table 10: Malware Bot Products 26

Table 11: Port Scanning Bot Products 26

Table 12: Malware Bot Countries 27

Table 13: Port Scanning Bot Countries 28

Table 14: Malware Bot Organizations 29

Table 15: Port Scanning Bot Organizations 29

Table 16: Malware Bot Open Ports 30

Table 17: Port Scanning Bot Open Ports 30

Table 18: Vulnerable Routers by Country 33

ABSTRACT

With the growing trend of Internet-enabled devices and the emergence of the Internet of Things (IoT), cybercrimes such as those carried out by botnets becomes a major issue. Previous research has attempted to estimate botnet population size, locate command and control servers, and utilize network security scanners. However, little work has been done that studies the characteristics of compromised devices belonging to botnets. In this research, we use data from several passive detection techniques including honeypots, VirusTotal, and Shodan to gain insights into these devices.

1 INTRODUCTION

As society becomes increasingly interconnected and more devices are becoming Internet-enabled, cybersecurity is a growing concern. With this increased degree of connectivity comes increased risk of cybercrime (Verizon, 2015). The prevalence of cybercrime is further exacerbated by easy access to hacking tools and tutorials within hacker communities and black markets (Holt, 2013).

One particularly dangerous aspect of cybercrime is the threat imposed by botnets. Botnets are collections of infected computers, often referred to as bots, drones or zombies, which are issued commands to carry out malicious activities. Example botnet activities include distributed denial of service (DDoS) attacks, spam distribution, and the spreading of malware.

In the fight against botnets, previous studies have relied on a mix of active and passive detection techniques, including the use of honeypots. While these studies have been useful in estimating botnet population size and locating the command and control (C&C) servers sending out

malicious commands, there has been a lack of work studying devices belonging to the botnets. Researching these compromised devices can:

- Provide insight into botnet composition
- Aid in identifying prevalent and emerging malware
- Assist device owners in improving their security posture

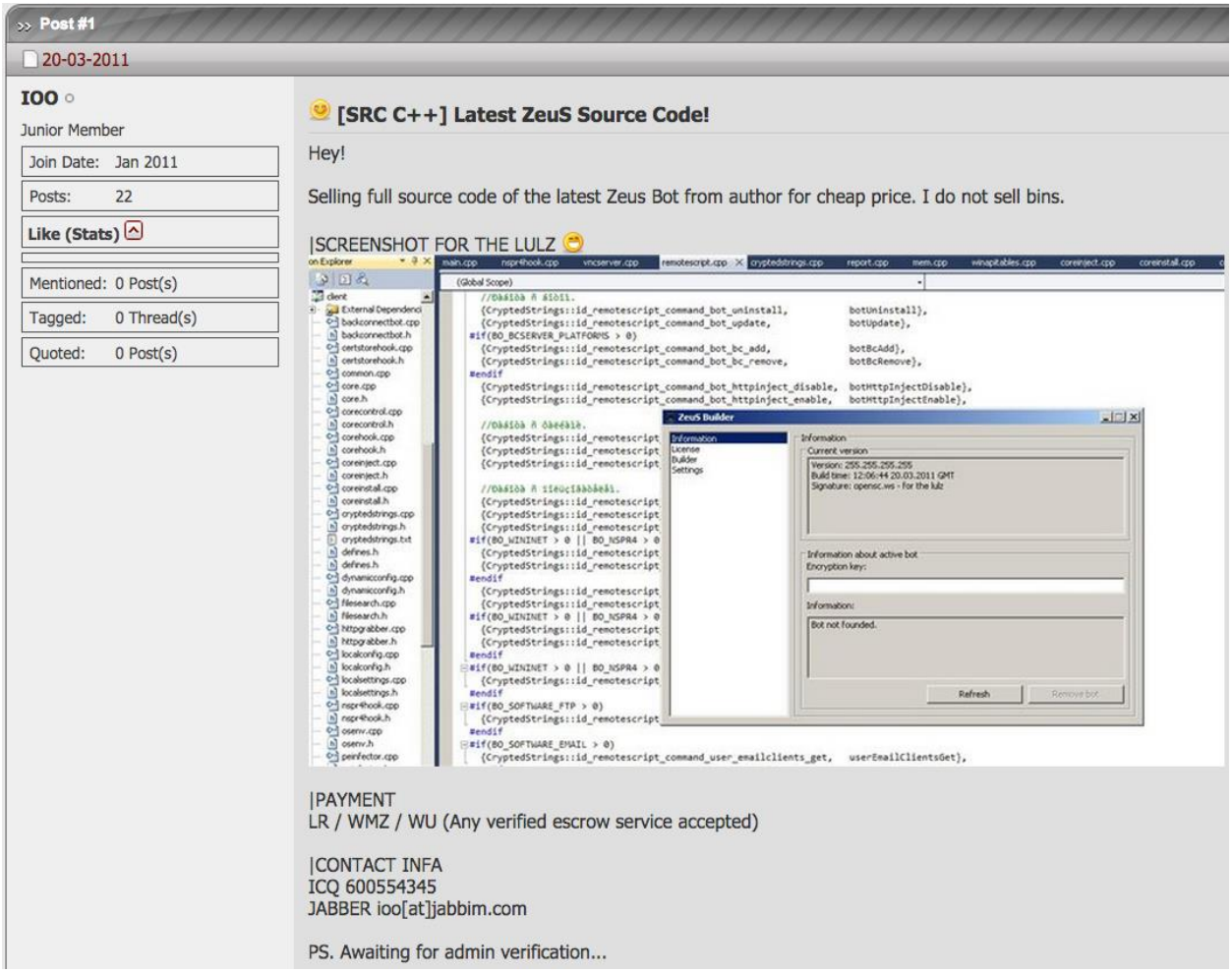

Figure 1: Example of Zeus Botnet Source Code Sold on Hacker Forum

Therefore, in this piece we are motivated to develop an automated framework for detecting and identifying compromised devices belonging to botnets and analyzing malicious software.

2 LITERATURE REVIEW

To form the basis for this research, we review literature from three key areas:

- Botnets
- Honeypots
- Internet of Things (IoT)

2.1 Botnets

A botnet is a network of compromised devices that have been assimilated to carry out malicious activities (Mielke & Chen, 2008). The attacker controlling these devices, also called a bot master or bot herder, utilizes a command & control (C&C) infrastructure to issue the malicious commands.

C&C architectures exists in one of two forms: centralized or decentralized. In a centralized architecture, the bots communicate with either one or a small number of command and control servers which in turn are being controlled by the bot master. In a decentralized architecture, commands are received by at least one device and spread to other devices on a peer-to-peer basis.

Botnets can be utilized for numerous nefarious purposes including: distributed denial-of-service (DDoS) attacks, spamming, and identity theft.

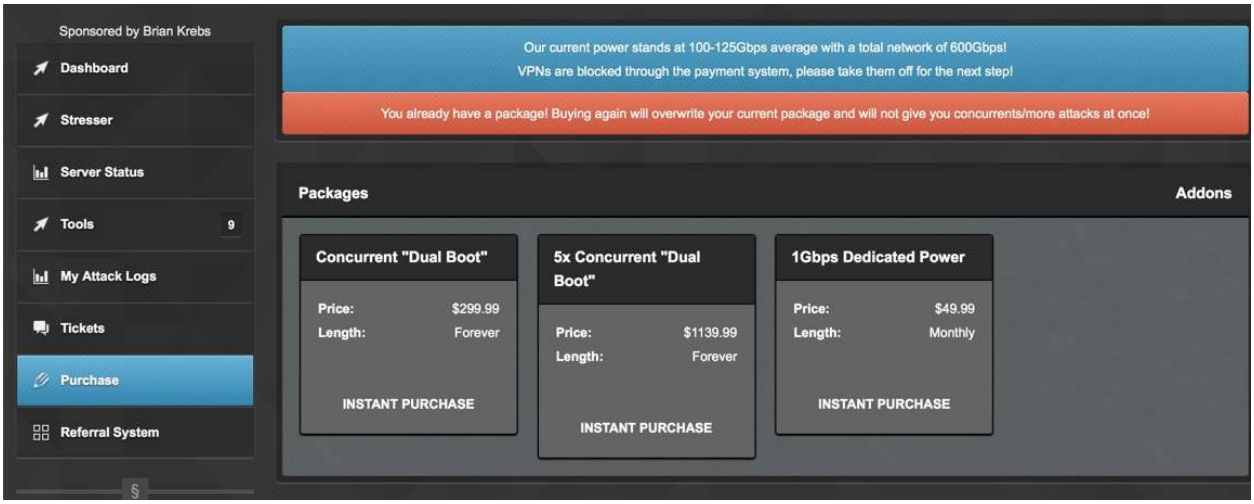

Figure 2: Example of DDoS Service for Sale that Runs on Botnet of Hacked Home Routers (Krebs, 2015)

Spam Botnet Name	Percentage of Botnet Spam	Estimated Spam per Day	Top Sources of Spam From Botnet					
			Rank #1		Rank #2		Rank #3	
KELIHOS	51.6%	884,044	Spain	10.5%	United States	7.6%	Argentina	7.3%
UNKNOWN/ OTHER	25.3%	432,594	United States	13.5%	Brazil	7.8%	Spain	6.4%
GAMUT	7.8%	133,573	Russia	30.1%	Vietnam	10.1%	Ukraine	8.8%
CUTWAIL	3.7%	63,015	Russia	18.0%	India	8.0%	Vietnam	6.2%
DARKMAILER5	1.7%	28,705	Russia	25.0%	Ukraine	10.3%	Kazakhstan	5.0%
DARKMAILER	0.6%	9,596	Russia	17.6%	Ukraine	15.0%	China	8.7%
SNOWSHOE	0.6%	9,432	Canada	99.9%	United States	0.02%	Japan	0.01%
ASPROX	0.2%	3,581	United States	76.0%	Canada	3.4%	United Kingdom	3.3%
DARKMAILER3	0.1%	1,349	United States	12.7%	Poland	9.6%	South Korea	9.1%
GRUM	0.03%	464	Canada	45.7%	Turkey	11.5%	Germany	8.5%

Top 10 Spam-Sending Botnets, 2014

Source: Symantec

Figure 3: Illustration of Top Spam-Sending Botnets (Symantec, 2015)

Study	Focus	Testbed	Methods	Findings
Dagon et al. (2006)	Analyzing botnet propagation dynamics	Captured malware samples over a 6-month period	DNS sinkholing	Developed botnet propagation model and identified botnet populations
Gu et al. (2007)	Detecting bot infections	2,019 malware infections	IDS packet inspection	Developed bot infection profile analysis tool
Livadas et al. (2006)	Identifying botnet traffic	Network traffic traces from a campus network and traces from a botnet	Machine learning for IRC traffic flow analysis	Distinguished between non-malicious and malicious IRC traffic
Mielke & Chen (2008)	Tracking botnets	20 GB of IRC log files collected by the ShadowServer Foundation	IRC-based measurement, social network analysis, clustering	Identified key botnet herders and number of bots

Table 1: Summary of Prior Botnet Studies

2.1.1 Active Detection Techniques

In detecting and measuring botnets, certain methods are considered active techniques when they involve participation in the botnet operation or interaction with the information sources being monitored (Khattak et al., 2014; Plohmann et al. 2011). While active techniques can capture a deep level of data, their use may be detected by bot masters.

Past studies have used a technique called DNS sinkholing as a means to estimate the size of botnet populations (Dagon et al., 2006). Sinkholing cuts off the compromised host from the C&C server by redirecting traffic to a server controlled by researchers or law enforcement. For the DNS request redirects to take place, this approach requires knowledge of the botnet C&C server as well as the cooperation of the DNS server owner used by the botnet.

Other studies have monitored Internet Relay Chat (IRC) channels, which are commonly used for C&C botnet administration (Mielke & Chen, 2008). By joining known IRC botnet C&C channels, channel participants can be counted and identified as either botnet herders or drones. This is achieved by parsing and developing signatures based on IRC protocol events such as JOIN, QUIT, PRIVMSG, and NICK.

2.1.2 Passive Detection Techniques

Passive detection approaches gather botnet data through observation and unobtrusively analyzing their activities. Therefore, passive detection techniques are often transparent and hidden to bot masters. This may come at the expense of not being able to collect as detailed data compared to active techniques.

A common passive detection approach is through the use of packet inspection and intrusion detection systems (IDSs). IDSs can be either signature based or anomaly based and traditionally focus on the inspection of inbound packets. One piece of work used the open source Snort IDS in conjunction with customized rules and malware analysis plugins to detect bot infections (Gu et al., 2007). By inspecting both the inbound and outbound packets throughout the infection lifecycle, the infected host and the C&C server could be identified.

An alternative to inspecting individual packets is an aggregate form of analysis examining flow records. Traffic flow characteristics include attributes such as source and destination address, total packets exchanged in the flow, and flow duration. Machine learning techniques are often applied to traffic flows to identify botnet traffic. One example classified IRC traffic flows as either benign or botnet related by using J48 (an open source Java implementation of the C4.5 decision tree algorithm), Naïve Bayes, and Bayesian network classifiers (Livadas et al., 2006). Another popular technique for the passive detection of botnets is the deployment of honeypots.

2.2 Honeypots

A honeypot is a network resource whose value is derived from being attacked and exploited (Spitzner, 2002). Honeypots can be divided into two broad categories: production honeypots and research honeypots. Production honeypots are deployed within organizations in order to detect attacks, mitigate risks, and improve the security of the network. They tend to be easier to deploy, but they provide less information on the attacker. Research honeypots on the other hand are more complex but have the ability to capture more detailed information. They are deployed with the goal to better understand the tactics, techniques, and procedures used by hackers and to research threats.

Honeypots can further be classified based on the level of interaction between the attacker and the honeypot (Spitzner, 2003). Low-interaction honeypots emulate vulnerable services and are therefore easier to manage and contain less risk. High-interaction honeypots are actual vulnerable systems that can be compromised.

Honeypot	**Classification**	**Description**
Dionaea	Low-interaction	Captures attack payloads and malware
Glastopf	Low-interaction	Emulates web server and collects web application-based attacks
HIHAT	High-interaction	Transforms PHP applications into web-based honeypots
Honeywall CDROM	High-interaction	Bootable CD for data capture, control, and analysis
Kippo	Low-interaction	Logs SSH brute force attacks and commands

Table 2: Examples of Honeypot Software

Study	Focus	Testbed	Methods	Findings
Al Awadhi et al. (2013)	Assessing cloud security	Three Dionaea honeypots on Amazon EC2	Splunk, VirusTotal	Identified top attackers, malware, and ports
Baecher et al. (2006)	Automatic, large-scale malware collection	Collected over 14,000 unique malware binaries	Vulnerable service emulation, antivirus scanning	Developed the Nepenthes platform
Brown et al. (2012)	Characterizing attack traffic across cloud providers	Dionaea and Kippo honeypots on Amazon EC2 and Windows Azure	OS fingerprinting, geolocation, VirusTotal	Identified top attacks and similarity between cloud environments
Goebel et al. (2007)	Analyzing autonomous spreading malware	13.4 million attacks within a university environment	Nepenthes honeypots, sandboxing, antivirus scanning	Introduced a malware measurement method and presented malware statistics

Table 3: Summary of Prior Honeypot Studies

2.2.1 Malware Honeypots

An important feature of honeypots used in the detection of botnets is their ability to capture malware. Collecting malware supports the saying "know your enemy" by aiding in the creation of malicious signatures used in IDSs and antivirus systems and by learning about attack patterns. However, collecting malware in the wild was previously a non-trivial task. It often required a detailed forensic examination of an infected machine to collect malware. To overcome the difficulties of this manual approach, the Nepenthes platform was developed (Baecher et al., 2006). Nepenthes offered an automatic approach to collecting malware in the form of a low-interaction honeypot. It was highly scalable since it emulated vulnerable services as opposed to high-interaction honeypots which are actually vulnerable.

The collection of malware through the use of Nepenthes honeypots is often the first step of analysis, out of a series of steps, when attempting to detect botnets. For example, a prior study collected over 13 million malware binaries before passing them into a sandbox tool for behavioral analysis (Goebel et al., 2007). The malware binaries were classified using four antivirus scanners and an additional tool was used to monitor for communications back to C&C servers.

2.2.2 Cloud Honeypots

With the growing popularity of cloud computing in the forms of infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), cloud security has been a growing concern as well. In order to assess the state of cloud security, work has been conducted deploying honeypots in the cloud (Al Awadhi et al., 2013; Brown et al., 2012). The studies deployed honeypots in data centers around the world and were able to identify the most prevalent attackers and strains of malware. They also identified that honeypots residing on different cloud providers are equally susceptible to attack. One of the drivers of cloud adoption has been the need to store and process the massive amount of data being generated by the Internet of Things.

2.3 Internet of Things (IoT)

The Internet of Things is the concept of physical objects connecting to the Internet. These smart objects, also referred to embedded systems or cyber-physical systems, contain embedded technology allowing them to sense or interact with the environment (Kopetz, 2011). A range of devices make up the IoT including:

1. Enterprise IT devices such as PCs, routers, servers, and switches
2. Operational technology such as medical machinery, process control, and supervisory control and data acquisition (SCADA) devices

3. Consumer devices such as smartphones, tablets, and wearables
4. Other single-purpose devices such as embedded communication systems in automobiles (Pescatore, 2014).

Study	Focus	Testbed	Methods	Findings
Bodenheim et al. (2014)	Evaluation of the Shodan search engine's ability to index and identify industrial control systems	Four Allen-Bradley ControlLogix programmable logic controllers	Device banner manipulation	Shodan indexed the devices within 19 days
Durumeric et al. (2013)	Security applications of Internet-wide scanning	IPv4 address space	Optimized probing mechanism	Developed ZMap, IPv4 address space can be scanned in under 45 minutes
Leverett (2011)	Locating and visualizing Internet-connected industrial control systems	Shodan search engine	29 manual key word searches	Identified and categorized over 7,500 devices
Radvanovsky (2014)	Quantifying the number of SCADA devices on the Internet	Shodan search engine	927 manual key word searches	Identified approximately 600,000 devices before project was discontinued

Table 4: Summary of Prior IoT Studies

2.3.1 Shodan Search Engine

As with cloud security, security of the IoT is becoming increasingly relevant. With 50 billion devices expected to be connected to the Internet by the year 2020, securing these devices will be a major issue (Evans, 2011). A tool that has been garnering attention for its ability to assess the security of the IoT is called Shodan. Originally developed as a market intelligence tool to allow

businesses to view what kind of networking devices their competitors were using, Shodan was created by John Matherly in 2009. Shodan is a search engine for the Internet of Things and has been called the scariest search engine on the Internet and the Google for hackers (Goldman, 2013; Hill, 2013). Rather than crawl the Internet for websites like Google, Shodan crawls the web and collects metadata for over one billion Internet-connected devices every month. Shodan currently scans over 180 different ports and operates by indexing the metadata it receives back from devices in the form of banners (refer to Appendix A). Information stored includes IP address, operating system, product, version, latitude, longitude, timestamp, and other device data.

Prior literature has primarily used Shodan in the context of locating and quantifying industrial control systems and SCADA devices that are exposed on the Internet (Leverett, 2011; Radvanovsky, 2014). The approach to locating these devices has relied on manual key word searches related to product or vendor names as well as subject matter expertise related to data contained in the device banners. A study also focused on Shodan's indexing capabilities and discovered that Shodan was able to index programmable logic controllers within 19 days of being connected to the Internet (Bodenheim et al., 2014).

2.3.2 Related Scanning Tools

Other network discovery and security scanners include Nmap and ZMap. Nmap is a popular open source tool used in security auditing that operates by actively probing and attempting to contact each service on the device being scanned. More recently, ZMap was developed as an efficient Internet-wide scanner that could scan the IPv4 address space in under 45 minutes, which is over 1,300 times faster than Nmap's capabilities (Durumeric et al., 2013). Thus, ZMap has security applications in performing Internet-wide scans to enumerate vulnerable hosts.

3 RESEARCH GAPS AND QUESTIONS

Previous studies have relied on a mix of active and passive detection techniques, including honeypots, for botnet detection. While these studies have been useful in estimating botnet population size, locating the command and control (C&C) servers sending out malicious commands, and performing network security scanning, there has been a lack of work studying the actual compromised devices in both an automated and passive fashion. Based on these research gaps, we propose the following research questions:

1. How can passive techniques be automated for botnet detection?
2. What are the characteristics of malware being propagated by compromised devices?
3. What are the characteristics of compromised devices belonging to botnets?

4 RESEARCH TESTBED

Eight Dionaea low-interaction honeypots (one in each region of Amazon EC2) are used in this research to log attack information and to automatically collect malware samples.

Honeypot Region	**# of Attacks**	**# of Unique Malware Samples**
Asia Pacific (Singapore)	25,041	11
Asia Pacific (Sydney)	25,209	22
Asia Pacific (Tokyo)	486,568	296
EU (Ireland)	1,236,498	700
South America (Sao Paulo)	268,233	283
US East (N. Virginia)	16,791	18
US West (Oregon)	16,806	14
US West (N. California)	17,446	15
Total	2,092,592	1,359

Table 5: Research Testbed

5 RESEARCH DESIGN

The research design consists of three primary phases:

- Honeypot deployment and attack collection on Amazon EC2
- Malware analysis using VirusTotal
- Device identification using Shodan

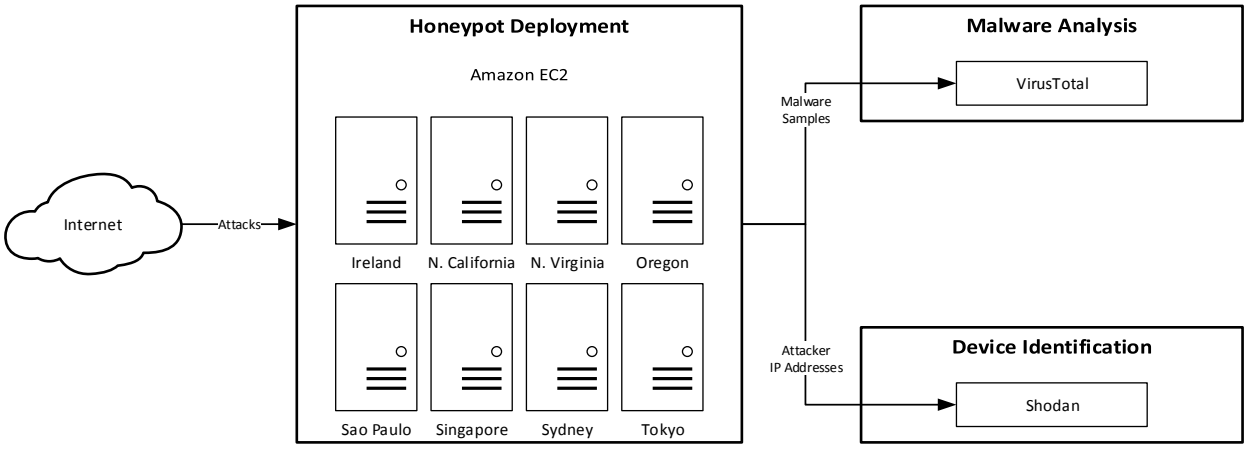

Figure 4: Research Design

5.1 Honeypot Deployment

In deploying honeypots for gathering attack data, several factors are taken into consideration including honeypot requirements, network location, configuration settings, and deployment duration.

5.1.1 Honeypot Requirements and Selection

For this research, a key requirement is the ability to capture malware in the wild. Thus, the low-interaction honeypot Dionaea was selected because of its widespread use throughout literature as an automated approach to capturing malware (Baecher et al., 2006; Goebel et al., 2007). Dionaea emulates a vulnerable Windows system and is the successor to the Nepenthes honeypot. Dionaea supports the following protocols: SMB (port 445), HTTP (port 80), FTP (port 21), TFTP (port 69), MSSQL (port 1433), MySQL (port 3306), and SIP VoIP (port 5060). SMB port 445, used

for resource sharing, is a primary port of interest because malware and bots commonly propagate by exploiting this port (Baecher et al., 2008).

5.1.2 Network Location

To eliminate the risk associated with deploying honeypots on our own local network, eight Dionaea honeypots were deployed on the Amazon Elastic Compute Cloud (EC2). In addition to reducing risk, deploying on Amazon EC2 provides a larger network attack surface, and thus the potential to collect more attacks and malware. The honeypots were deployed in data centers located in the following regions: Ireland, N. California, N. Virginia, Oregon, Sao Paulo, Singapore, Sydney, and Tokyo. Amazon recently introduced a data center in Frankfurt, however a honeypot was not deployed in that region since its introduction postdated the launching of the other honeypots.

5.1.3 Configuration

To setup the instances on which the honeypots are hosted, we created Amazon Web Services free tier accounts and configured the instances using the details below.

Amazon Machine Image	Type	Memory (GB)	Storage (GB)
Ubuntu Server 12.04 LTS	t1.micro	0.613	30

Table 6: Amazon EC2 Instance Configuration

After the required Ubuntu packages and dependencies were installed, Dionaea was compiled and set to log incoming attack information to an SQLite database residing on the honeypot (refer to Appendix B).

5.1.4 Deployment Duration

After the instances were properly configured, Dionaea was launched simultaneously across the honeypots at midnight on October 30, 2014. The honeypots collected attack data for 125 days,

ending on March 4, 2015. Throughout the data collection phase, a web front-end called DionaeaFR was used to monitor the status of the honeypots.

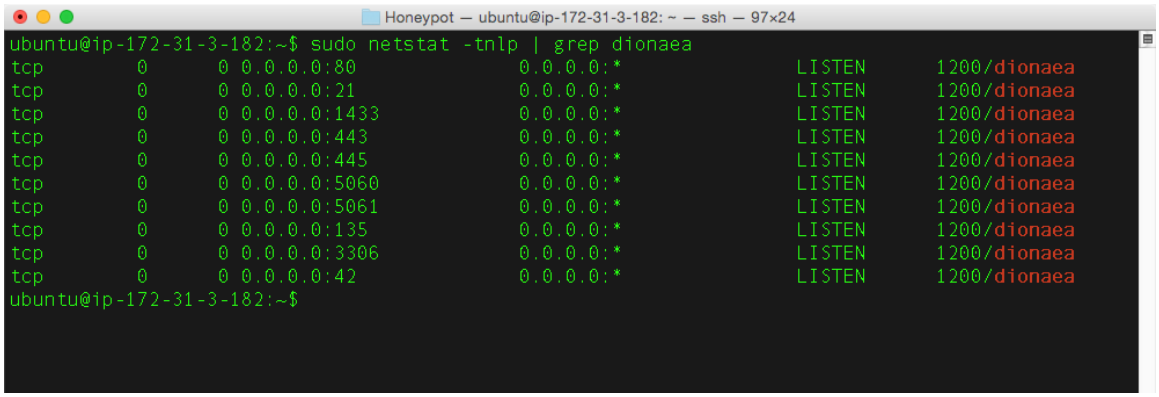

Figure 5: Illustration of Dionaea Network Status after Deployment

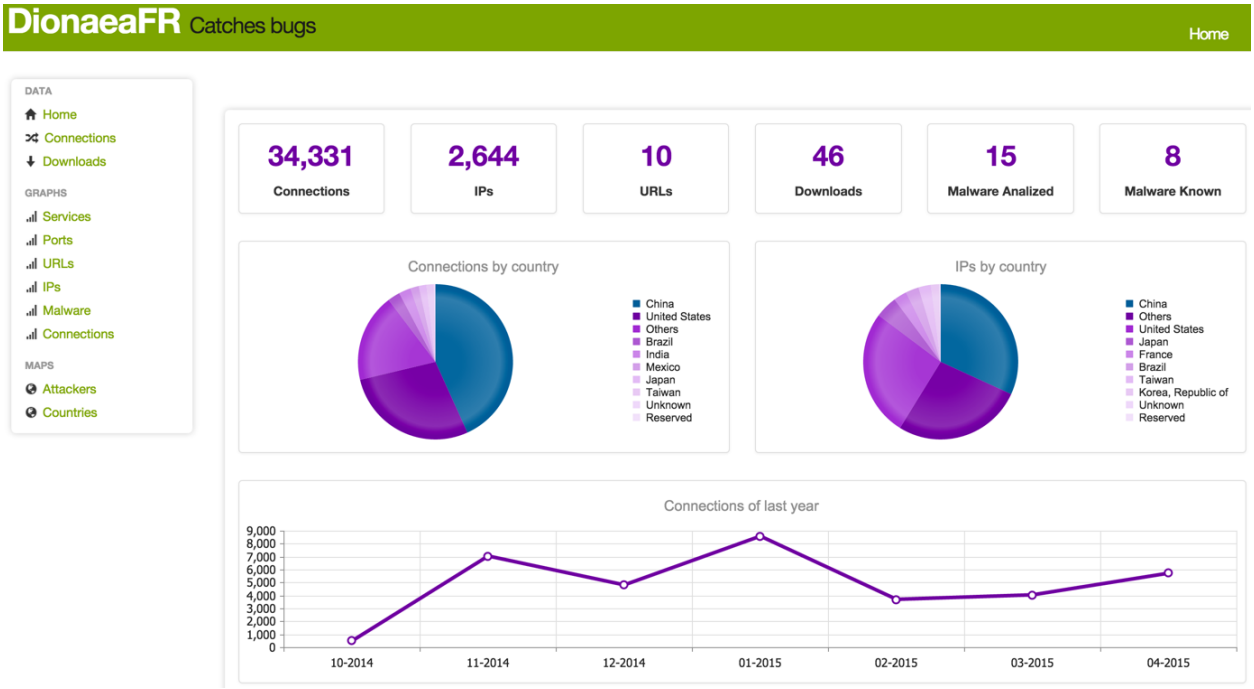

Figure 6: Illustration of DionaeaFR Dashboard

5.2 Malware Analysis

VirusTotal, a freely available online service used to identify malicious files and URLs, was used to analyze malware captured by the honeypots. After Dionaea downloaded a copy of the malware, it was automatically submitted for analysis using the VirusTotal API. The malware was analyzed using 54 different antivirus engines and the resulting malware classification was stored in the honeypot database. The Sophos antivirus engine was selected as the classifier of choice because of its high detection rate and low false positive rate (AV-Comparatives, 2015).

5.3 Device Identification

After collecting the attack information and submitting the malware for analysis, a Python script was developed to retrieve the IP addresses of the attackers and to search for those IPs on Shodan. The script calls the Shodan REST API, parses the JSON response, and stores the device data in a MySQL database for further analysis (refer to Appendix C).

6 RESULTS AND DISCUSSIONS

The following results are derived from 13,700 unique IP addresses attacking the eight honeypots over the 125-day timespan. The 13,700 IP addresses were passed into Shodan resulting in the discovery of 7,107 devices. The devices not discoverable via Shodan are either hidden behind a firewall or do not have any open ports to be indexed by Shodan. The 7,107 devices are divided into two subsets for examination: 2,442 devices that were attempting to infect the honeypots with malware and 4,665 devices that were only probing or port scanning the honeypots.

6.1 VirusTotal Results

The VirusTotal scan result is the classification of the malware according to the Sophos antivirus engine and the count is the total number of times that variant of malware attempted to infect the honeypot.

6.1.1 Top Malware

The Conficker worm accounts for 99 percent of the attempted infections while the remaining infections consist of generic forms of malware, Trojans, and spyware.

VirusTotal Scan Result	**Count**
Mal/Conficker-A	366,607
W32/Confick-O	204,003
Troj/Agent-UOB	191,786
W32/Confick-C	882
W32/Confick-D	58
Troj/DLoad-IK	53
Mal/Spy-Y	51
Mal/Dropper-O	47
W32/Confick-A	24
W32/Confick-F	23
Mal/PWS-JJ	18
Mal/Generic-L	17
Troj/Brambul-A	17
Troj/Agent-ZIU	15
Troj/Agent-ABCG	6
Mal/TDSSPack-T	4
W32/Sality-I	4
W32/Virut-Gen	2
Mal/Generic-S	1
Total	763,618

Table 7: VirusTotal Malware Classifications

6.2 Shodan Results

The count column represents the distinct number of devices found on Shodan. Unfortunately, the results returned by Shodan's device type field are quite sparse and contain a large amount of null values. The count column in the top ports section refers to the number of devices with that particular port open.

6.2.1 Top Device Types

Of the device types that Shodan was able to classify, the top devices included wireless access points (WAPs), firewalls, webcams, routers, and private branch exchanges (PBXs) used in business telephone systems.

Device Type	**Count**
Null	2,411
WAP	38
webcam	27
router	4
firewall	2
security-misc	2
broadband router	1
PBX	1
Total	2,442

Table 8: Malware Bot Device Types

Device Type	**Count**
Null	4,491
WAP	208
firewall	40
webcam	18
router	14
PBX	9
broadband router	4
security-misc	3
media device	2
storage-misc	2
specialized	1
Total	4,665

Table 9: Port Scanning Bot Device Types

6.2.2 Top 10 Products

Each device discovered via Shodan may be running multiple products including web servers (Microsoft IIS, Apache, Allegro RomPager, nginx), databases (MySQL, Microsoft SQL Server), and remote access and control software (OpenSSH, VNC).

Product	Count
Microsoft IIS httpd	171
MySQL	117
Apache httpd	112
VNC	83
Microsoft ftpd	80
Allegro RomPager	61
Dropbear sshd	61
Microsoft ESMTP	53
Microsoft SQL Server	50
OpenSSH	40

Table 10: Malware Bot Products

Product	Count
OpenSSH	624
Apache httpd	396
MySQL	371
Microsoft IIS httpd	312
Microsoft ftpd	199
Apache Tomcat/Coyote JSP engine	177
nginx	107
VNC	83
Linksys wireless-G WAP http config	81
Allegro RomPager	77

Table 11: Port Scanning Bot Products

6.2.3 Top 10 Countries

Using the geolocation data provided by Shodan, the countries with the highest number of devices are shown in Tables 12 and 13. In Figures 7 and 8, darker shades of red indicate countries with a higher number of devices.

Country	Count
Russia	286
Taiwan	229
United States	213
India	136
Romania	116
Ukraine	97
Venezuela	91
Bulgaria	77
Argentina	75
Vietnam	74

Table 12: Malware Bot Countries

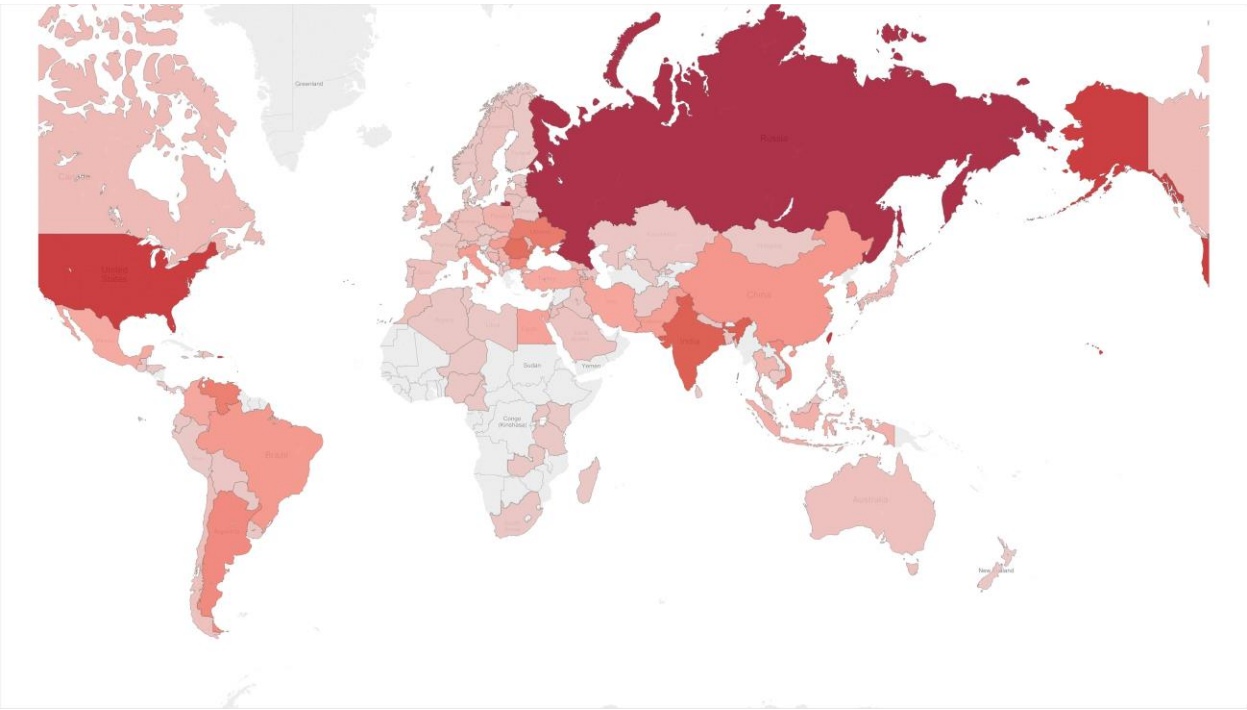

Figure 7: Map of Malware Bot Countries

Country	Count
China	1,334
United States	855
India	195
Taiwan	176
Brazil	150
Russia	125
Mexico	82
Vietnam	78
Germany	69
Netherlands	69

Table 13: Port Scanning Bot Countries

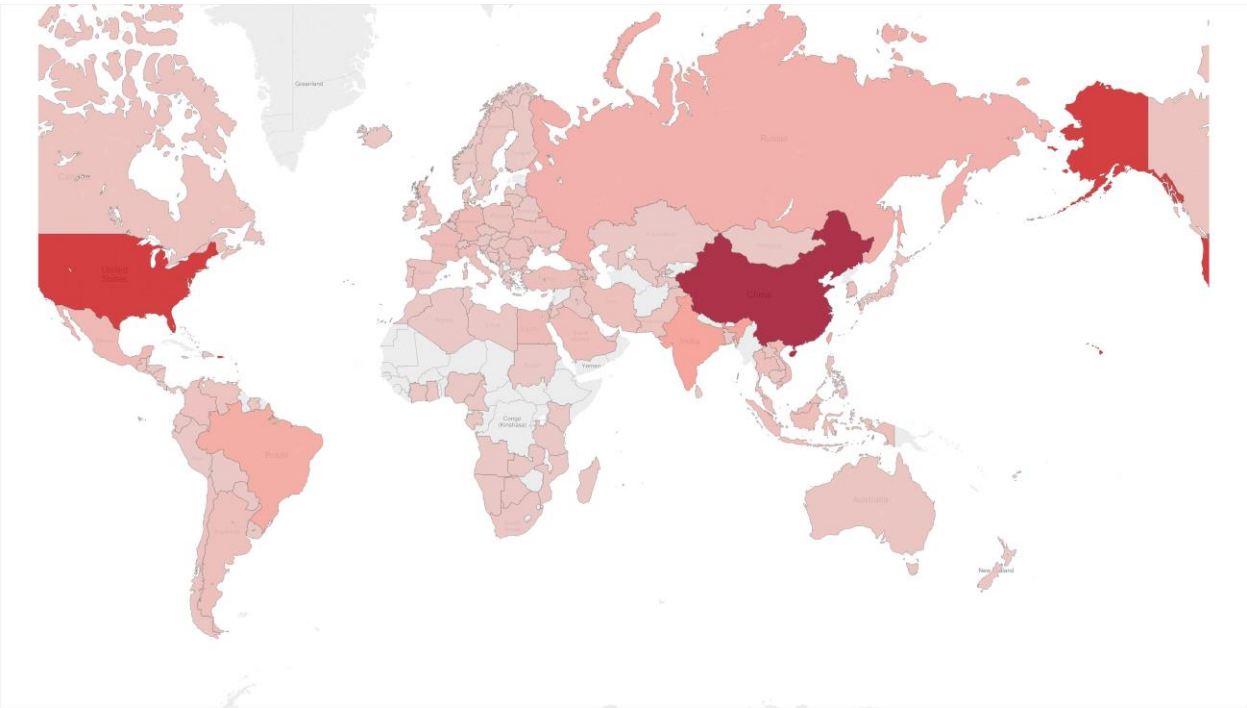

Figure 8: Map of Port Scanning Bot Countries

6.2.4 Top 10 Organizations

With the exception of Amazon.com, Shodan illustrates that the devices primarily belong to telecommunications companies in China, Russia, and Taiwan.

Organization	Count
CHTD, Chunghwa Telecom Co., Ltd.	139
Beeline	93
CANTV Servicios, Venezuela	80
OJSC Rostelecom	70
Chunghwa Telecom Data Communication Business Group	67
BSNL	55
Telefonica de Argentina	50
Telecom Italia	41
McLaut ISP	40
TE Data	37

Table 14: Malware Bot Organizations

Organization	Count
China Telecom jiangsu province backbone	158
Amazon.com	148
CHTD, Chunghwa Telecom Co., Ltd.	118
Psychz Networks	103
BSNL	91
China Telecom Guangdong	70
China Telecom Jiangxi	69
China Telecom Shanghai	68
Shenzhen Tencent Computer Systems Company Limited	64
Uninet S.A. de C.V.	58

Table 15: Port Scanning Bot Organizations

6.2.5 Top 10 Open Ports

For a device to be indexed by Shodan, it must have one or more open ports accepting connections, as opposed to closed ports that reject connections. The number of devices with the specified open port are shown below, with port 80 being the top port.

Port Number	Service Name	Count
80	HTTP	1,016
137	NetBIOS	1,004
445	SMB	626
3389	RDP	577
7547	Modem Web Interface	241
21	FTP	183
1900	UPnP	143
5357	Microsoft-HTTPAPI/2.0	128
8080	HTTP	128
22	SSH	127

Table 16: Malware Bot Open Ports

Port Number	Service Name	Count
80	HTTP	1,779
3389	RDP	1,282
22	SSH	910
137	NetBIOS	843
21	FTP	581
8080	HTTP	580
7547	Modem Web Interface	377
3306	MySQL	374
1723	PPTP	334
500	IKE	239

Table 17: Port Scanning Bot Open Ports

6.3 Discussion

6.3.1 Conficker Worm

Of all the malware attempting to infect the honeypots, the results were almost entirely Conficker worm variants (Troj/Agent-UOB is a variant of Conficker). Conficker is a self-propagating Internet worm that seeks out vulnerable machines and recruits them as part of a botnet. Despite the worm being discovered in November 2008 and the formation of the Conficker Working Group to combat the infections, Conficker is still a prevalent threat (Rendon Group, 2011).

6.3.2 Malware Bots versus Port Scanning Bots

When observing the differences between the devices attempting to propagate malware versus those performing port scans, there are some interesting findings. The top countries for malware spreading devices are Russia and Taiwan whereas the top countries for devices conducting port scans are China and the United States. Viewing the top organizations for port scanning devices indicates that Amazon.com is the second highest source of devices. Examining the honeypot logs and Shodan data reveals that these devices are Amazon EC2 instances that may have been compromised, once again reiterating the importance of cloud security. An interesting port that is open on a number of devices carrying out port scans is port 7547. This port is used by the protocol called TR-069 or CWMP (customer-premises equipment wide area network management protocol). CWMP is used to remotely troubleshoot and configure routers and is one of the ports associated with a major router vulnerability discussed in the next section.

6.3.3 Router Vulnerability Case Study

When observing the products associated with both the malware propagating and port scanning bots, an interesting result that stands out among the traditional web servers and database products is the Allegro RomPager. RomPager is an embedded web server, most commonly found in small office/home office (SOHO) routers, that is made by Allegro Software Development Corporation, a provider of embedded Internet software components.

Researchers discovered a critical vulnerability, called Misfortune Cookie, in over 200 brands of routers using the embedded web server and CVE-2014-9222 was released on December 24, 2014 (Check Point Software Technologies, 2015). Routers with versions of RomPager version 4.34 and below (most commonly version 4.07) are vulnerable. The vulnerability allows intruders to remotely take over routers and gain administrative privileges by sending specially crafted HTTP cookies to their public IP address. This vulnerability is significant because once a router has been compromised, any devices connected to the network are also at risk of being compromised. Hackers can exploit this to monitor Internet traffic, steal personal information, and infect other computers and IoT devices such as printers, security cameras, and more (Grau, 2015).

When the vulnerability was first discovered, researchers conducted an Internet-wide scan using ZMap and detected over 12 million exploitable devices. As of May 2015, 7,459,383 exploitable devices could still be detected by a Shodan search using the query below.

product:"Allegro RomPager" version:"4.07 UPnP/1.0"

The top countries, as indicated by Shodan, containing routers affected by the Misfortune Cookie vulnerability are shown below. Countries filled with a darker shade of red contain higher quantities of vulnerable routers.

Country	**Count**
Mexico	1,612,331
India	733,766
Italy	576,340
Egypt	564,720
Colombia	444,911
Iran	434,299
Indonesia	373,965
Thailand	323,802
Malaysia	246,246
Vietnam	184,634

Table 18: Vulnerable Routers by Country

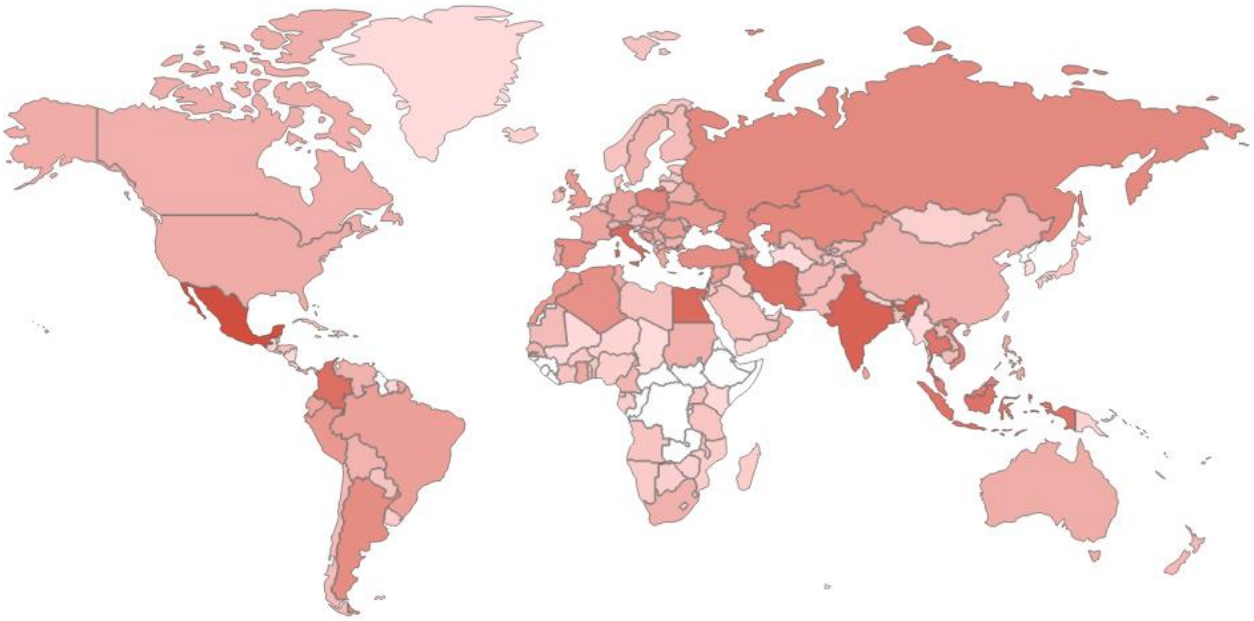

Figure 9: Map of Vulnerable Routers

7 CONCLUSION

As the number of Internet-connected devices continues to increase and the Internet of Things begins to take shape, cybersecurity is especially relevant. With higher levels of connectivity comes an increased risk of cybercrime and the potential for remote exploitation. The collective threat represented by botnets is one particular area of cybercrime that needs addressing.

For this research, botnets, specifically the compromised devices belonging to the botnets, are detected and identified by leveraging multiple passive detection techniques. Dionaea low-interaction honeypots were deployed on Amazon EC2 to collect attack information and malware binaries. VirusTotal and Shodan were used for malware classification and device identification respectively. The top malware variants, device types, products, countries, organizations, and open ports were examined. Compromised devices along with the continued existence of a critical router vulnerability were identified.

This research was unique in that it used honeypots in conjunction with the Shodan search engine to study compromised devices. Device owners can use these findings to increase their awareness of device vulnerabilities and to increase their overall security posture. To expand upon this research, the deployment of honeypots can be automated to achieve greater scalability and to greatly increase the amount of data collected. High-interaction honeypots can be explored in order to collect more granular data and to supplement the low-interaction honeypots. Further work is also needed to develop additional device signatures for device identification via Shodan or related scanning technologies.

8 REFERENCES

Al Awadhi, E., Salah, K., & Martin, T. (2013). Assessing the security of the cloud environment. *GCC Conference and Exhibition (GCC), 2013 7th IEEE*. doi:10.1109/IEEEGCC.2013.6705785

AV-Comparatives. (2015). File Detection Test - March 2015. Retrieved from http://www.av-comparatives.org/wp-content/uploads/2015/04/avc_fdt_201503_en.pdf

Baecher, P., Holz, T., Koetter, M., & Wicherski, G. (2008). Know your Enemy: Tracking Botnets. Retrieved from https://www.honeynet.org/papers/bots

Baecher, P., Koetter, M., Holz, T., Dornseif, M., & Freiling, F. (2006). The Nepenthes Platform: An Efficient Approach to Collect Malware. In D. Zamboni & C. Kruegel (Eds.), *Recent Advances in Intrusion Detection SE - 9* (Vol. 4219, pp. 165–184). Springer Berlin Heidelberg. doi:10.1007/11856214_9

Bodenheim, R., Butts, J., Dunlap, S., & Mullins, B. (2014). Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. *International Journal of Critical Infrastructure Protection*, *7*(2), 114–123. doi:http://dx.doi.org/10.1016/j.ijcip.2014.03.001

Brown, S., Lam, R., Prasad, S., Ramasubramanian, S., & Slauson, J. (2012). *Honeypots in the Cloud*. University of Wisconsin-Madison. Retrieved from http://pages.cs.wisc.edu/~sbrown/downloads/honeypots-in-the-cloud.pdf

Check Point Software Technologies, (2015). Misfortune Cookie. Retrieved from http://mis.fortunecook.ie/

Dagon, D., Zou, C., & Lee, W. (2006). Modeling Botnet Propagation Using Time Zones. In *In Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*.

Durumeric, Z., Wustrow, E., & Halderman, J. A. (2013). ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)* (pp. 605–620). Washington, D.C.: USENIX. Retrieved from https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

Evans, D. (2011). *The Internet of Things How the Next Evolution of the Internet Is Changing Everything*. Retrieved from https://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

Goebel, J., Holz, T., & Willems, C. (2007). Measurement and Analysis of Autonomous Spreading Malware in a University Environment. In B. M. Hämmerli & R. Sommer (Eds.), *Detection of Intrusions and Malware, and Vulnerability Assessment SE - 7* (Vol. 4579, pp. 109–128). Springer Berlin Heidelberg. doi:10.1007/978-3-540-73614-1_7

Goldman, D. (2013). Shodan: The scariest search engine on the Internet. Retrieved from http://money.cnn.com/2013/04/08/technology/security/shodan/index.html

Grau, A. (2015). Can you trust your fridge? *Spectrum, IEEE*. doi:10.1109/MSPEC.2015.7049440

Gu, G., Porras, P., Yegneswaran, V., Fong, M., & Lee, W. (2007). BotHunter: Detecting Malware Infection Through IDS-driven Dialog Correlation. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium* (pp. 12:1–12:16). Berkeley, CA, USA: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm?id=1362903.1362915

Hill, K. (2013, September). The Terrifying Search Engine That Finds Internet-Connected Cameras, Traffic Lights , Medical Devices , Baby Monitors And Power Plants. Retrieved from http://www.forbes.com/sites/kashmirhill/2013/09/04/shodan-terrifying-search-engine/

Holt, T. J. (2013). Examining the Forces Shaping Cybercrime Markets Online. *Social Science Computer Review*, *31*(2), 165–177. doi:10.1177/0894439312452998

Khattak, S., Ramay, N. R., Khan, K. R., Syed, A. A., & Khayam, S. A. (2014). A Taxonomy of Botnet Behavior, Detection, and Defense. *Communications Surveys & Tutorials, IEEE*. doi:10.1109/SURV.2013.091213.00134

Kopetz, H. (2011). Internet of Things. In *Real-Time Systems SE - 13* (pp. 307–323). Springer US. doi:10.1007/978-1-4419-8237-7_13

Krebs, B. (2015). Lizard Stresser Runs on Hacked Home Routers. Retrieved from http://krebsonsecurity.com/2015/01/lizard-stresser-runs-on-hacked-home-routers/

Leverett, E. P. (2011). Quantitatively assessing and visualising industrial system attack surfaces. *University of Cambridge, Darwin College*. Retrieved from https://www.cl.cam.ac.uk/~fms27/papers/2011-Leverett-industrial.pdf

Livadas, C., Walsh, R., Lapsley, D., & Strayer, W. T. (2006). Usilng Machine Learning Technliques to Identify Botnet Traffic. *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. doi:10.1109/LCN.2006.322210

Mielke, C. J., & Chen, H. (2008). Botnets, and the cybercriminal underground. *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*. doi:10.1109/ISI.2008.4565058

Pescatore, J. (2014). *Securing the "Internet of Things" Survey*. Retrieved from https://www.sans.org/reading-room/whitepapers/analyst/securing-internet-things-survey-34785

Plohmann, D., Gerhards-Padilla, E., & Leder, F. (2011). *Botnets: Measurement, Detection, Disinfection and Defence*. Retrieved from https://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/botnets/botnets-measurement-detection-disinfection-and-defence

Radvanovsky, B. (2014). *Project SHINE Findings Report*. Retrieved from http://www.slideshare.net/BobRadvanovsky/project-shine-findings-report-dated-1oct2014

Rendon Group. (2011). *Conficker Working Group: Lessons Learned*. Retrieved from http://www.confickerworkinggroup.org/wiki/uploads/Conficker_Working_Group_Lessons_Learned_17_June_2010_final.pdf

Spitzner, L. (2002). *Honeypots: Tracking Hackers*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Spitzner, L. (2003). The Honeynet Project: trapping the hackers. *Security & Privacy, IEEE*. doi:10.1109/MSECP.2003.1193207

Symantec. (2015). *2015 Internet Security Threat Report, Volume 20*. Retrieved from http://www.symantec.com/security_response/publications/threatreport.jsp

Verizon. (2015). *2015 Data Breach Investigations Report*. Retrieved from http://www.verizonenterprise.com/DBIR/

9 APPENDIX A: Ports Scanned by Shodan

Port	Service	Port	Service	Port	Service
7	Echo	1900	UPnP	7071	Zimbra HTTP
11	Systat	1911	Tridium Fox	7547	Modem Web Interface
13	Daytime	1962	PCWorx	7657	HTTP (7657)
15	Netstat	2067	DLSW	7777	Oracle
17	Quote of the day	2082	cPanel	8000	Qconn
19	Character Generator	2083	cPanel + SSL	8069	OpenERP
21	FTP	2086	WHM	8080	HTTP (8080)
22	SSH	2087	WHM + SSL	8087	Riak Protobuf
23	Telnet	2123	GPRS Tunneling Protocol	8089	Splunk
25	SMTP	2152	GPRS Tunneling Protocol	8090	Insteon Hub
37	rdate	2323	Telnet (2323)	8098	Riak Web Interface
53	DNS	2375	Docker	8129	Snapstream
67	DHCP	2376	Docker + SSL	8139	Puppet Agent
79	Finger	2404	IEC-104	8140	Puppet Master
80	HTTP	2455	Codesys	8181	GlassFish Server
81	HTTP (81)	2628	Dictionary	8333	Bitcoin
82	HTTP (82)	3000	ntop	8443	HTTPS (8443)
83	HTTP (83)	3128	Squid Proxy	8834	Nessus
84	HTTP (84)	3306	MySQL	8888	AndroMouse
88	Kerberos	3386	GPRS Tunneling Protocol	9000	NAS Web Interfaces
102	Siemens S7	3388	RDP (3388)	9051	Tor control port
110	POP3	3389	RDP	9100	Printer Job Language
111	Portmap	3479	2-Wire RPC	9151	Tor control port
119	NNTP	3780	Nexpose	9160	Cassandra
123	NTP	3790	Metasploit	9200	ElasticSearch
129	Password generator	4022	Udpxy	9600	OMRON FINS
137	NetBIOS	4040	Chef	9943	Pipeline Pilot + SSL
143	IMAP	4369	Erlang Port Mapper Daemon	9944	Pipeline Pilot
161	SNMP	4443	Symantec Data Center	9981	HTS/ tvheadend
389	LDAP	4500	IKE-NAT-T	9999	Telnet (Lantronix)
443	HTTPS	4911	Tridium Fox + SSL	10000	Webmin
445	SMB	4949	Munin	10001	Automated Tank Gauge
465	SMTP + SSL	5000	Synology	10243	Microsoft-HTTPAPI/2.0
500	IKE	5001	Synology	11211	MemCache
502	Modbus	5006	Mitsubishi MELSEC-Q	16010	Hbase
515	Line Printer Daemon	5007	Mitsubishi MELSEC-Q	18245	General Electric SRTP
523	IBM DB2	5008	NetMobility	18246	General Electric SRTP
623	IPMI	5060	SIP	20000	DNP3
626	serialnumbered	5094	HART-IP	20547	ProConOS
631	CUPS	5222	XMPP	25565	Minecraft
771	RealPort	5353	mDNS	27017	MongoDB
789	Red Lion	5357	Microsoft-HTTPAPI/2.0	28017	MongoDB Web
992	Telnet + SSL	5432	PostgreSQL	32764	Router backdoor
993	IMAP + SSL	5560	Oracle HTTP	44818	EtherNetIP
995	POP3 + SSL	5632	PC Anywhere	47808	BACnet
1023	Telnet (1023)	5900	VNC	49152	Supermicro Web
1200	Codesys	5901	VNC (5901)	50100	Telnet
1234	Udpxy	5985	WinRM 2.0	55553	Metasploit (55553)
1434	MS-SQL Monitor	5986	WinRM 2.0 + SSL	55554	Metasploit (55554)
1471	Hak5 Pineapple	6000	X Windows	62078	iPhone
1604	Citrix	6379	Redis	64738	Mumble server
1723	PPTP	6666	Voldemort		

10 APPENDIX B: Dionaea Honeypot Data Dictionary

Attribute	**Description**
Connection Timestamp	When the attack occurred
Local Port	Honeypot port being attacked
Remote Host	IP address of the attacker
Remote Port	Originating port of the attack
Download URL	Originating location of the malware
Download MD5 Hash	Hash value of captured malware
VirusTotal MD5 Hash	Hash value of malware submitted to VirusTotal
VirusTotal Timestamp	When the malware was first submitted to VirusTotal
VirusTotal Permalink	Link to report generated by VirusTotal
VirusTotal Scanner	Name of antivirus engine used in the scan
VirusTotal Result	Classification of the malware

11 APPENDIX C: Shodan Data Dictionary

Attribute	Description
Device Information	
Port	Port number the host is operating on
Banner Data	Contains the banner information for the service
Timestamp	Timestamp of the scan
HTML	HTML source of the site
Product	Product name which generated banner
CPE	Common Platform Enumeration for device
Info	Miscellaneous data about device
Version	Version of the product that generated banner
Opts	Other raw info (e.g. SSL certs, robots.txt, etc.)
Device Type	Type of device (webcam, router)
OS	Operating system of the device
Uptime	Number of minutes device has been online
Network Information	
IP	IP address of the host as an integer
IP_str	IP address of the host as a string
Org	Name of the organization assigned this IP
ISP	Internet Service Provider of this organization
ASN	Autonomous System Number of device
Hostnames	All the hostnames this device has had
Domains	Top level domain of the device
Link	Network link type (e.g., Ethernet or modem)
Location Information	
Country Code	2-letter country code for the device location
Country Code 3	3-letter country code for the device location
Country Name	Country the device is located in
Latitude	Latitude of device
Longitude	Longitude of device
City	City the device is located in
Postal Code	Postal code for the device's location
Area Code	Area code for the device's location. US only
DMA Code	Designated Market Area code for the area. US only
Region Code	Name of the region where the device is located