Capability Analysis of Internet of Things (IoT) Devices in Botnets and Implications for Cyber Security Risk Assessment Processes

Prof. Andrew R. Schmitt, Metropolitan State University

Andrew Schmitt is an information security professional with a passion for networking. Starting his career in application and end-user support, his passion for network and security technologies was quickly realized. Currently, Schmitt is a cybersecurity professional with a focus on network security and network threat prevention. Additionally, he is a community faculty member at Metropolitan State University where he teaches cybersecurity courses. His favorite part of being an information security professional is the rapidly changing environment and the challenge of keeping enterprises protected.

Theresa Chasar, Newell Brands

Theresa Chasar is an Information Security Operations Director who works with Newell Brands to protect the company's critical assets and continuously monitor and improve its security defenses. Theresa applies her depth of experience in security engineering and technical architecture to business initiatives, ensuring the alignment of innovation and security. She believes that a balance between business operations and security operations is achievable when teams collaborate on a shared vision.

Theresa holds a Master's degree in Security Technologies from the University of Minnesota.

Miss Mangaya Sivagnanam, Ingersoll Rand

Mangaya Sivagnanam is a principal cybersecurity systems architect with 17 years of experience in software applications design, analysis, development, testing and deployment of web/enterprise based on client/server applications and commercial industrial control systems. She is responsible for the framework and application design and development of web-based and embedded software for control systems. Mangaya has expertise and experience in innovation, security architecture for the web application, industrial control systems, internet of things, mobile, cloud computing, big data security, smart connected buildings and smart cities. She has extensive experience with heterogeneous system's software design (Secure SDLC), threat modeling, security and risk analysis, penetration testing. She is also responsible for coordinating and managing the incident response process for the advanced building automation systems and solutions. She received an MS degree in Security Technologies | Cybersecurity in Technology Leadership Institute University of Minnesota.

Dr. Faisal Kaleem, Metropolitan State University

Dr. Faisal Kaleem received his Ph.D. in Electrical Engineering from Florida International University (FIU), Miami, FL. Since 1998 he has served as an educator in different academic institutions. Currently, he is serving as an Associate Professor in the Department of Information and Computer Sciences at Metropolitan State University, and a Senior Fellow at the Leadership Institute (TLI) at the University of Minnesota. Dr. Kaleem is an experienced lifelong cybersecurity practitioner. His research interests include multiple aspects of cybersecurity including Smart Grid Security, Computer, and Network Security but more specifically in the area of mobile device security, mobile malware analysis, and attribution, and mobile forensics. In the past few years, Dr. Kaleem has developed and taught several courses (Ethical Hacking, Digital Forensics Engineering, Mobile Device Forensics, Malware Reverse Engineering, etc.) in the area of cybersecurity.

Dr. Kaleem has extensive experience managing federally funded cybersecurity programs, including programs funded by the National Science Foundation (NSF) to conduct research using Smart Glasses applications to identify the unique skillsets of cybersecurity analysts, learning gaps, and augmented reality learning solutions. In addition, he recently received NSA grants to provide cybersecurity training to veterans and other underrepresented communities. He also developed modules on Mobile Forensics with grant support from Intel Corporation. With the help of Minnesota IT Center of excellence, Dr. Kaleem has established MnCyber—a statewide institute for Cybersecurity and forensics research and education. He

is currently serving as the executive director of MnCyber. He is also the co-founder and executive member of Minnesota Cyber Career Consortium (MNC3) whose mission is to address Minnesota's cybersecurity workforce needs and to help assist Minnesota businesses in handling cyber risks.

Dr. Kaleem has established a solid track record in teaching and has received numerous awards including the best professor and the best course awards (cybersecurity) from various graduating cohorts. Dr. Kaleem is an advisory board member to various organizations as well as the faculty advisor for the Cybersecurity and Forensics Students Organization. He also leads the Collegiate Cyber Defense Competition (CCDC) at Metropolitan State University. During his free time, he also provides various Internet Safety workshop for parents and children. Dr. Kaleem served as the program committee member for NICE 2016 conference as well as the Academic co-chair of the NICE 2017 conference that was held in Dayton, OH. He continually appears on various local news channels discussing various issues in the area of cybersecurity and currently holds various industry certifications such as CISSP, CEH, Security+, MCT, CCLO, and CCPA.

Capability Analysis of Internet of Things (IoT) Devices in Botnets and Implications for Cyber Security Risk Assessment Processes

Metropolitan State University

Abstract: *Internet of Things based botnet attacks are increasing. In September 2016, "Krebs on Security," which focuses on security news and investigation, was taken offline after a sustained Distributed Denial of Service (DDoS) attack. The botnet, comprised of IoT devices, generated over 636 Gb of traffic per second. DDoS attacks following the attack on Krebs have generated over 1.2 Tb per second. To date, there is limited research to quantify the attack capabilities of IoT devices. Our research analyzes the maximum attack capability that can be generated and harnessed in a single target IoT botnet attack. This analysis will help defenders predict how DDoS attacks will affect their systems and consequently architect resilient infrastructure solutions. As the IoT landscape continues to grow, this paper provides timely research for security professionals who need to understand attack capabilities in an IoT based botnet and the risk associated with potential botnet DDoS attacks.*

Keywords: *Internet of Things, Denial of Service Attacks, Information Security, Network Security*

1. INTRODUCTION

The Internet of Things (IoT) is made up of connected devices from smart household appliances to industrial sensors. Innovation across all industries is leading to explosive growth in the number of IoT devices connected to the internet. In 2017, an estimated 28.4 billion devices are internet connected [1]; each with an IP address and the ability to transmit data. These devices have little or no built-in security, and users often fail to change default passwords before connecting them to their network; making them easy targets for hackers. By comprising these connected devices and harnessing them into a botnet, IoT has intensified the threat of conventional distributed denial of service (DDoS) attacks.

In September 2016, large IoT botnet attacks started drawing attention. What started as a rarely seen or largely theoretical attack become front page news. Three major sites were hit in a month; disrupting services and affecting users worldwide. Those sites include the French Internet service provider OVH (1.1 Tbps attack), DNS service provider DynDNS (1.2 Tbps attack), and journalist Brian Krebs' website (636 Gbps attack) [2].

Our research focuses on the maximum network bandwidth attack capabilities that can be generated from IoT bots and botnets. By emulating IoT devices in a lab environment and testing the bandwidth capabilities of compromised devices, we can simulate situations that real world defenders find themselves facing. The results of this research can assist defenders in architecting solutions with the resiliency to defend against high volume bandwidth attacks.

1.1 Hypotheses

There are two hypotheses that we used as the basis of this research. The first hypothesis is that the CPU and RAM of an IoT device will have an effect on the ability of a device to generate bandwidth in a socket connection. We estimated that a device with higher CPU and RAM would generate more bandwidth than a device with lower CPU and RAM resources. The second hypothesis was that network capability would not be the limiting factor for any given IoT device because these devices will not approach the limitation of 802.3 (Ethernet) or 802.11 (wireless) standards.

2. RELATED WORK

Significant research has been done on IoT risk assessments and the DDoS attack capabilities of botnets. Our research builds upon those risk assessments and the vulnerabilities that make IoT devices susceptible to compromise, resulting in participation in DDoS attacks.

Perakovic et al. [3] found that the emergence of IoT increases the number of internet connected devices. Their paper synthesized and analyzed statistical data on protocols used in the generation of infected bots; concluding that the growth of IoT device numbers will affect their use in creating botnets used in DDoS attacks. Likewise, in a general study of IoT vulnerabilities [4], Alsaadi and Tubaishat found that, in the distributed form of architecture in IoT, attackers could hijack unsecured network devices converting them into bots to attack third parties.

Ronen and Shamir [5] identify four categories of attacks used in conjunction with IoT devices; ignoring, reducing, misusing, and extending the functionality of the devices. Botnets are created by ignoring the functionality of devices and comprising them like any standard computing device. The vulnerabilities that allow IoT devices to be comprised are outlined in the OWASP Internet of Things Top 10 [6] list. The vulnerabilities that constitute the highest risk are insecure interfaces, insufficient authentication, and lack of encryption. Through exploitation of those vulnerabilities, IoT devices can be infected and participate as bots in DDoS attacks.

In a related work, “Analysis of a Botnet Takeover” [7], Stone-Gross et al. describe the experiences of actively seizing control of the Torpig botnet and studying its operations for 10 days. In an attempt to quantify the botnet footprint, the authors calculated the total number of infected bots by counting the unique IP addresses connected to the Command and Control (C&C) server. Acknowledging that this method was problematic, due to the network effects of DHCP churn and NAT, they concluded that another method needs to be formulated.

Considering established research on IoT risks and the call of published researchers to further quantify the effects of compromised devices, we will focus our research on the maximum bandwidth capability that can be generated per IoT device in a single-target botnet attack.

3. METHODS

3.1 Definitions

3.1.1 Denial of Service Attacks

Denial of service (DoS) attacks can be categorized as either crashing or flooding attacks. Crashing attacks involve exploiting known vulnerabilities to trigger a system hard down state. In this type of attack, input is sent to a system with the intent of completely destabilizing a system so that it cannot be accessed. Flood attacks happen when a system receives too much traffic for the server to buffer, causing them to slow down to the point of stopping.

Distributed denial of service (DDoS) attacks occur when multiple devices are leveraged into a botnet and used to target a single system. Flooding attack methods are used in DDoS strikes to increase the volume of traffic aimed at the target. DDoS attacks can also be reflected and amplified to further increase the volume of traffic generated.

Generally speaking, reflected DDoS attacks are any attack where the attacker spoofs the source IP address to be the address of the intended target and amplified attacks are any attack where the original attack is enhanced by use of another protocol, redirection, or spoofed means. For example, Microsoft defines a DNS amplification attack as, "a type of distributed denial of service (DDoS) attack that takes advantage of the fact that a small DNS query can generate a much larger response. When combined with source address spoofing, an attacker can direct a large volume of network traffic to a target system by initiating relatively small DNS queries [8]."

3.1.2 Sockets

Sockets involve the connection between two devices on a specific port number. The listening computer waits for attempted connections to be made on a specific port before negotiating a connection to establish an active socket. Sockets are represented by an IP address and port number (i.e. 10.0.0.1:80) and operate on layer 4 of the OSI model [9].

3.1.3 Buffers

Buffers are primarily used for flow control and allow the receiving computer to temporarily store data into a dedicated memory space until it is ready to read and respond to the data packets sent to it [10].

3.1.4 Packets

A packet is a unit of transmission in the TCP protocol that consists of a header and a payload. RFC 791 defines a header as, "Control information at the beginning of a message, segment, datagram, packet or block of data [11]." In UDP, the equivalent unit of transmission is referred to as a datagram.

3.1.5 HTTP GET/POST Requests

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers. HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a website may be the server. HTTP GET requests focus on requesting data from a server and the server providing the

requested data back to the client. HTTP POST requests usually involve a client providing data to a server and the server uploading or storing that information [12].

3.2 IoT Emulation through Virtualization

To adequately measure the effects of available resources (CPU, RAM, and networking capability) on IoT botnets, we architected an IoT emulation environment utilizing virtualization techniques. Fig. 1 below outlines the high-level topology of our emulation laboratory. The goal of this emulation environment was to remove as many variables as possible and to focus on the abilities of an emulated IoT device.

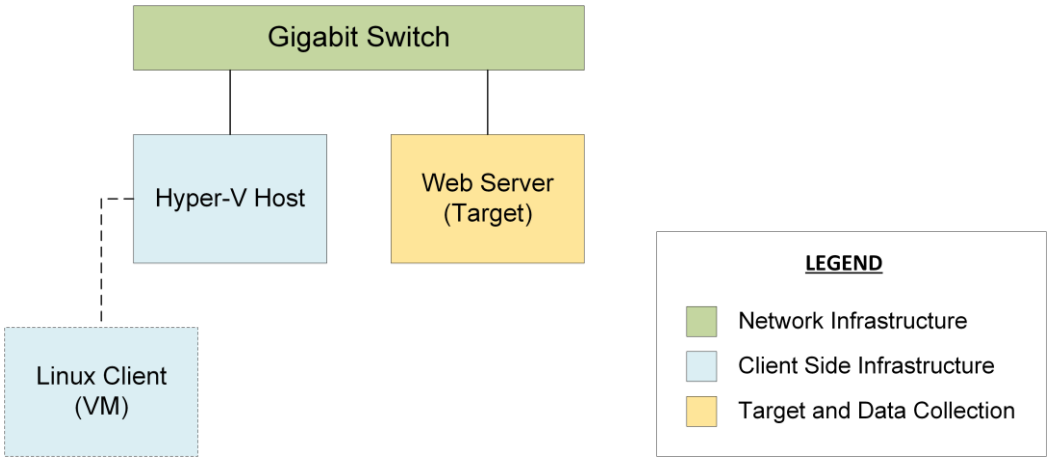

Fig. 1 IoT Emulation Topology

The IoT emulation laboratory consisted of three areas; network infrastructure, client side infrastructure, and target and data collection infrastructure. Regarding network infrastructure, we used a Cisco Catalyst 3750-E series switch with a single VLAN and no additional connected devices to minimize unnecessary traffic that may compromise our data. From a hardware perspective, the client side infrastructure consisted of a Dell T750 with 6GB of RAM and an Intel Xeon E5507 processor. The hardware hosting the Windows Hyper-V hypervisor was used to create a single Ubuntu based virtual machine with resources controlled by the hypervisor and scaled during testing. The target and data collection infrastructure consisted of a customized PC that had 24GB of RAM and an Intel i7 processor. The operating system of choice was Windows Server 2012R2. To measure the bandwidth on the target infrastructure, we used a program called BitmeterOS. BitmeterOS is a vendor agnostic application created by Codebox Software that was installed on the target infrastructure to collect data from the target perspective. The application was bound to the NIC-team interface which was composed of four one gigabit interfaces using category 6 cable. BitmeterOS was responsible for calculating maximum bandwidth achieved during the testing intervals to provide a maximum denial of service capability for each testing scenario. BitmeterOS 0.7.6 was used during all testing iterations. For additional information regarding infrastructure, refer to Appendix A.

This emulation focused on collecting data for two resource types: CPU and RAM. We chose to exclude limiting network resources during this phase of research due to our desired goal of

determining the maximum bandwidth capability for each scenario, and because it was hypothesized that these emulation scenarios would not reach the capability of 802.3 ethernet standards, or 802.11 wireless standards. To examine the effects of CPU and RAM on bandwidth generation capability, we chose to maximize the amount of a particular resource while completing incremental increases of the other resource during each testing set (i.e. CPU was maximized while incrementing RAM 5% for each testing set).

Each testing iteration consisted of a server side script written in Python, a client side script written in Python, and the measuring of bandwidth on the server side device using an application called BitmeterOS. The Python scripts use standard libraries to establish sockets and determine the size of the buffer used by the server. For details regarding the Python scripts used, refer to Appendix B.

The client side device was set to the desired specification of resources for CPU and RAM through the hypervisor prior to each iteration. Testing iterations ranged from 5-100% of the limiting resource with the non-limiting resource being set to 100% utilization. Upon verification of the environment set up, a buffer size is specified for the server, a socket is established between the client and server, the client then passes data from a text file to the server via the active socket, and the server computer measures the observed bandwidth using BitmeterOS. The test was conducted until the maximum bandwidth observed did not change for two minutes.

3.3 Data

The data collected is separated into two categories: CPU based testing and RAM based testing. In each set of testing the specific resource is being incremented to determine the effect on bandwidth generation capability. For the complete set of raw data collected, refer to Appendix C.

3.3.1 CPU Incremental Testing

TABLE I
CPU INCREMENTAL TESTING DATA

	CPU Resources Allocated (RAM 100%)					
	5%	25%	50%	80%	100%	Average
Buffer Size (B)	0.11 GHz	0.57 GHz	1.12 GHz	1.82 GHz	2.27 GHz	
512	482.83	345.88	342.71	343.88	343.01	362.29
1024	753.06	637.16	655.50	618.64	616.11	638.96
2048	1,495.04	1,218.56	1,280.00	1,259.52	1,331.20	1,279.32
4096	3,072.00	2,488.32	2,529.28	2,529.28	2,385.92	2,636.46
8192	7,321.60	4,771.84	4,935.68	4,710.40	4,403.20	5,068.80
16384	10,905.60	10,219.52	9,625.60	9,738.24	9,226.24	9,707.52
32768	22,282.24	18,647.04	18,780.16	19,087.36	18,083.84	19,030.70
65536	40,048.64	33,935.36	35,706.88	40,325.12	40,048.64	37,097.47
	Observed Maximum Bandwidth (Kbps)					

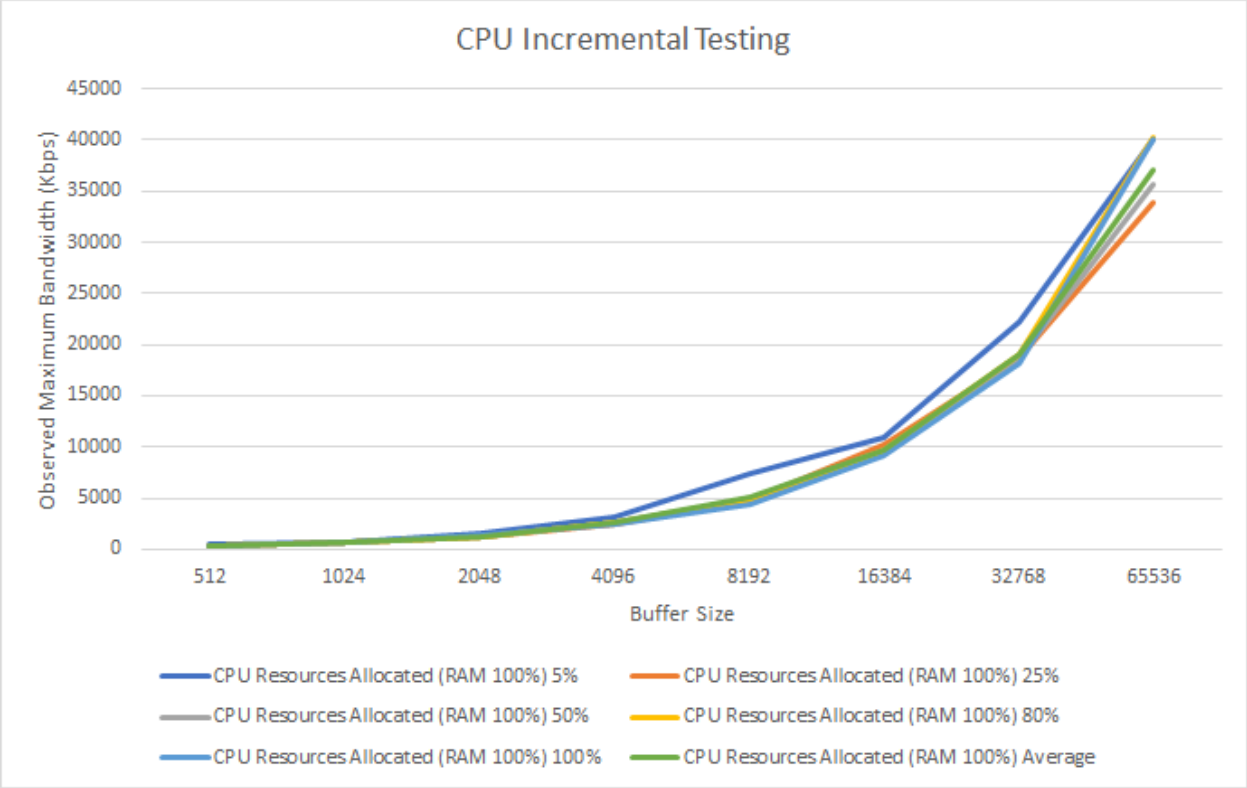

Fig. 2 CPU incremental testing data

3.3.2 RAM Incremental Testing

TABLE II
RAM INCREMENTAL TESTING DATA

	RAM Resources Allocated (CPU 100%)					
	10%	25%	50%	80%	100%	Average
Buffer Size (B)	409.6 MB	1024 MB	2048 MB	3277 MB	4096 MB	
512	345.44	349.87	345.25	350.10	343.10	345.55
1024	613.55	611.99	625.30	612.31	615.16	620.61
2048	1,310.72	1,313.09	1,228.80	1,317.10	1,313.66	1,305.91
4096	2,426.88	2,417.34	2,488.32	2,500.00	2,501.30	2,448.15
8192	4,126.72	4,201.71	5,017.60	4,127.44	4,100.63	4,272.76
16384	9,277.44	9,314.64	9,502.72	9,274.30	9,255.11	9,294.65
32768	17,991.68	17,876.34	17,899.52	17,991.66	17,898.97	17,936.71
65536	34,938.88	34,741.95	33,792.00	34,993.54	34,918.44	34,649.42
	Observed Maximum Bandwidth (Kbps)					

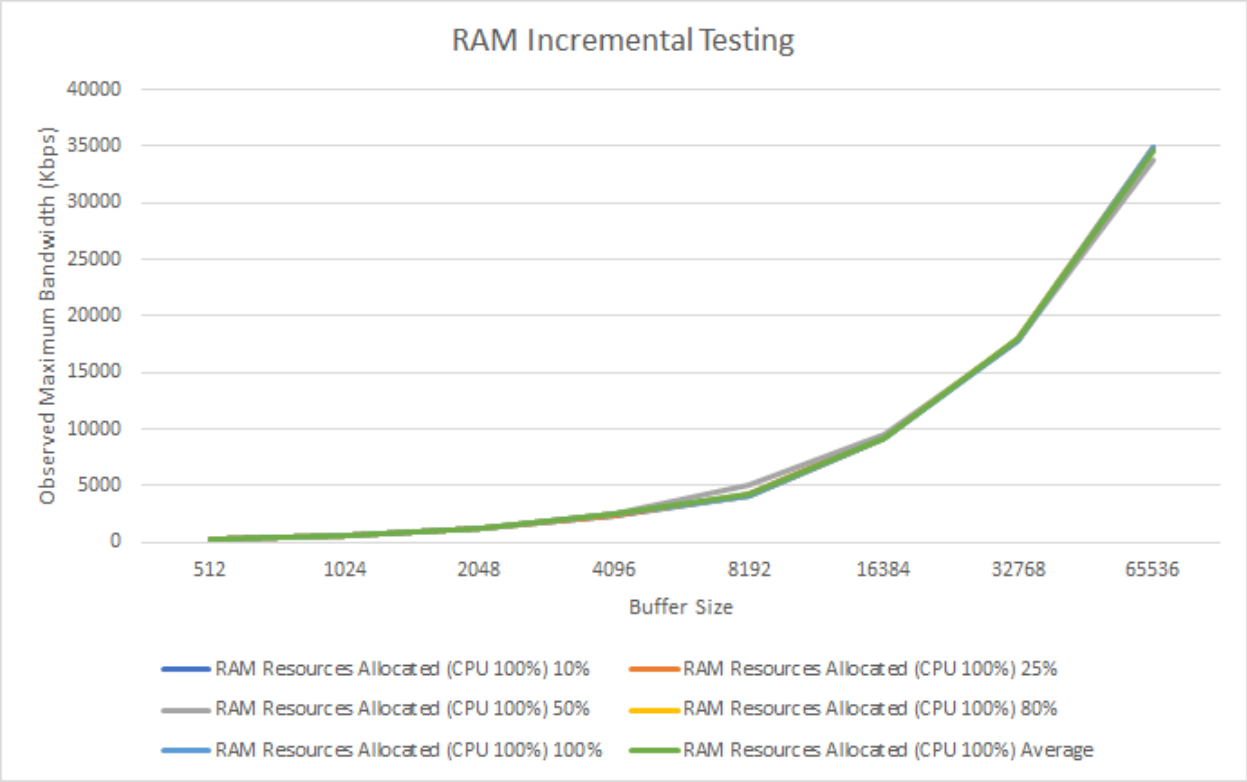

Fig. 3 RAM incremental testing

4. RESULTS

4.1 Analysis Assumptions

To analyze the raw data, we made the following assumptions:

1. Compromise of the devices is assumed, there is no emphasis on method of compromise.
2. The default buffer size for most web servers is 8,192 bytes [13]. Although a buffer size can be configured, we assume that servers in our attack analyses are using a buffer size of 8,192 bytes.
3. The average packet size of a HTTP GET and POST request is 512 bytes.
4. All compromised IoT bots are performing similar attack techniques on the target system.
5. All bots are attacking a single target.
6. If the adversary uses amplification techniques during an attack, all the bots achieve the same amount of amplified attack capability.

4.2 Hypothesis Review

As stated, we initially began this research with two hypotheses: 1) CPU and RAM will have an effect on maximum bandwidth generated, and 2) network capability will not be the limiting resource for an individual IoT device's ability to generate bandwidth. After reviewing the data collected from our IoT emulation, we determined that our first hypothesis was incorrect while our second hypothesis remained true.

TABLE III
CPU TESTING
AVERAGES

CPU Testing Raw Data	
Buffer Size (b)	Observed Average Bandwidth (Kbps)
512	362.28
1024	638.95
2048	1,279.32
4096	2,636.46
8192	5,068.80
16384	9,707.52
32768	19,030.70
65536	37,097.50

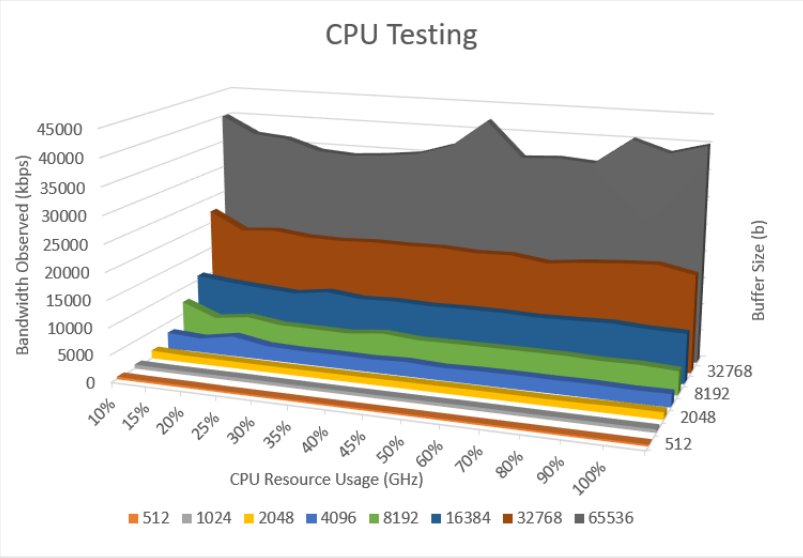

Fig. 4 Observed average bandwidth for CPU testing

Our data collection from section 3.3.1 shows that there was no meaningful change in maximum bandwidth observed while adding CPU resources to an emulated IoT device. While conducting iterative testing between 5-100% CPU allocation, we determined that CPU did not influence

maximum bandwidth generation. However, we did find that increasing buffer size during testing did have a significant effect on maximum bandwidth generation. The figure above shows a graphical representation of data collection in the CPU testing. The table above provides the average bandwidth observed in different CPU usage for the defined buffer size. Analyzing the dataset, we concluded that buffer size has the most impact on maximum bandwidth generation and that CPU allocation is independent of maximum bandwidth generation.

TABLE IV
RAM TESTING
AVERAGES

RAM Testing Raw Data	
Buffer Size (b)	Observed Average Bandwidth (Kbps)
512	345.55
1024	620.61
2048	1,305.91
4096	2,441.55
8192	4,272.75
16384	9,294.65
32768	17,936.71
65536	34,649.42

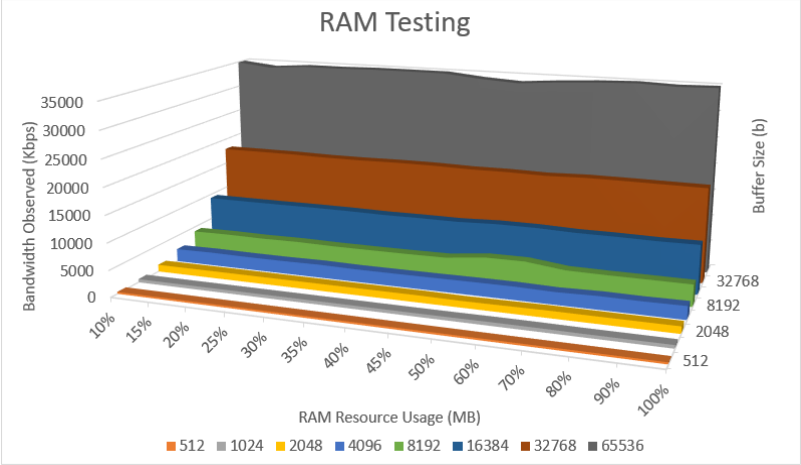

Fig. 5 Observed average bandwidth for RAM testing

Section 3.3.2 yielded the same findings as section 3.3.1 and clearly demonstrates that there is no significant relationship between allocated RAM resources during maximum bandwidth generation in an IoT emulation scenario. We conducted iterative testing between 10-100% RAM allocation and discovered no relationship between allocated RAM and maximum bandwidth generated. However, as the above figure shows, we discovered that buffer size has a significant effect on the maximum bandwidth generated by the emulated IoT device. The table above provides the average bandwidth observed in different RAM usage for the defined buffer size. Analyzing the dataset, we concluded that buffer size has the most impact on maximum bandwidth generation and that RAM allocation is independent of maximum bandwidth generation.

The analysis of testing results shows that our first hypothesis was incorrect. CPU and RAM do not appear to have significant effects on maximum bandwidth generation for emulated IoT devices. Our analysis shows that buffer size is the limiting factor in maximum bandwidth generation in a single socket system. However, our analysis did confirm that the bandwidth generated by an emulated IoT device did not approach the limitations of 802.3 (Ethernet) or

802.11 (wireless) standards.

4.3 Risk Assessment

4.3.1 IoT Device Implications

Our results demonstrate that CPU and RAM do not have a significant effect on maximum bandwidth generation in single socket target attacks. Rather, the buffer on the destination server has the most significant impact in influencing the maximum bandwidth observed from an IoT device. Based on these results, we conclude that IoT devices are not limited by the amount of resources available or the quality of the hardware used to create IoT devices. Rather, IoT devices are limited by the standards and protocols used in single socket, single target communications. This also demonstrates that all IoT devices, despite their size, resources, or intended use will contribute the same attack capability to an IoT botnet. In terms of assessing risk regarding IoT devices, we conclude that we must treat all IoT devices as having the same amount of risk in terms of their contribution to an IoT botnet.

To continue to analyze the risk associated with IoT devices in botnets, the maximum generated bandwidth for IoT devices will be expressed as:

Maximum Bandwidth Generated = [Average Bandwidth Observed at Specific Buffer Size]

4.3.2 IoT Botnet Implications & IoT Attack Implications

As stated, all IoT devices have equal contribution to botnet capability in a single socket, single target attack. In this scenario, botnets find their strength in the number of IoT bots that are included in the botnet. The DynDNS attack was said to consist of 100,000 bots and, similarly, the OVH attack was said to consist of over 145,000 bots [2]. Many DDoS attacks are utilizing the botnet herding capabilities released as part of the Mirai botnet or are variations of the Mirai botnet. Currently, the industry is speculating that there could be as many as 50 billion IoT devices connected to the internet by 2020 [1]. This means that by 2020, there could be as many as 50 billion devices that are candidates to join an IoT based botnet.

As we consider the attack capability for IoT based botnets in a single socket, single target, non-amplified attack we will evaluate attack capability as:

*Maximum IoT Botnet Capability = [Number of IoT Bots] * [Average Bandwidth Observed at a Specific Buffer Size]*

Similarly, if we include amplification as an additional factor in a single socket, single target, amplified attack where the amplified capability is observed from each device, we will evaluate attack capability as:

*Maximum IoT Botnet Capability = Amplification Factor([Number of IoT Bots] * [Average Bandwidth Observed at a Specific Buffer Size])*

To evaluate risk regarding IoT botnets, we generated the following graphical representation of

IoT botnet attack capabilities based on number of IoT bots.

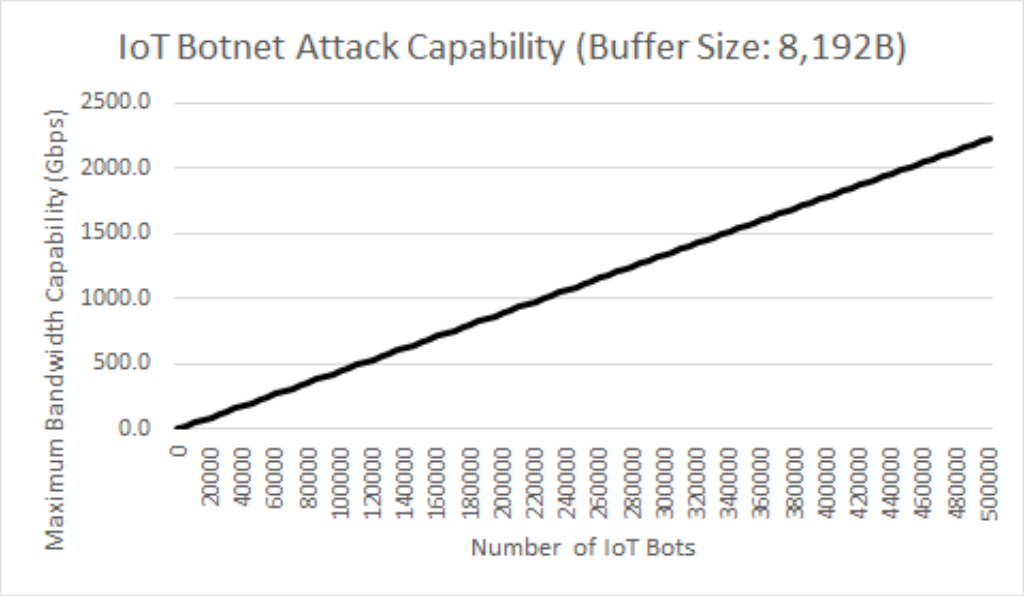

Fig. 6 IoT botnet attack capability

From the graphical representation of a non-amplified, single socket, single target attack, an IoT botnet could generate over 2 Tbps of DDoS traffic. Brian Krebs, protected by Akamai at the time of attack, was taken offline by approximately 620-630 Gbps of DDoS traffic [2]. Similarly, DynDNS and OVH experienced significant attacks of over 1Tbps that took them offline [2]. With the ability to generate over 2 Tbps of persistent non-amplified traffic with IoT devices ranging from DVRs to movie streaming devices, IoT botnets could have significant or catastrophic effects on even the largest enterprises.

4.4 Real World Scenario: Krebs on Security

The attack on Brian Krebs was the first sensationalized report of a DDoS attack generating more than 600 Gbps of persistent attack capability. Although determining the amount of DDoS bandwidth is possible with a high level of accuracy, determining the number of devices involved in a DDoS attack is not an easy task. After the Brian Krebs attack, it was reported that the botnet that attacked Krebs, now known to be Mirai or a close variant, was utilizing direct non-amplified DDoS attacks. Based on the findings of our research, we estimate that there were approximately 142,773 IoT bots that took part in the attack which would have resulted in 636 Gbps of sustained DDoS traffic [2].

4.5 Real World Scenario: DynDNS

The attack on DynDNS generated over 1.2 Tbps of sustained DDoS traffic. The estimated number of IoT bots that were used in this attack was 100,000 [2]. Analyzing the traffic observed during this attack, third party organizations determined that amplification and reflection techniques were used to enhance traffic. Based on our research and assumptions, for a botnet to achieve this level of sustained traffic with 100,000 devices, each device would gain an increase

of 276% of the original maximum traffic. This would mean that if an IoT bot sent 4,671 Kbps (4-5 Mbps) of traffic as part of a reflection attack, it would be capable of 12,845 Kbps (12-13 Mbps) of DDoS traffic. This also assumes that each device is attempting to generate the maximum amount of traffic possible. Depending on the request type, whether DNS, HTTP, or other, and the protocol, TCP vs. UDP, this is not a high threshold to meet.

4.6 Evaluating Risk for Enterprises and Individuals

As IoT devices become more prevalent for individuals and enterprises alike, the level of risk continues to grow rapidly. The Mirai botnet has quickly proven that IoT devices are not developed with security in mind and that they are easily compromised. Without drastic action, taken now, to improve the security of IoT devices, the risk associated with IoT based botnets will continue to grow at an exponential rate as we approach the estimated 50 billion IoT devices of 2020.

Home users typically have somewhere between 10-100 Mbps of internet download bandwidth available. Our findings indicate that a botnet of only 6-22 IoT devices could cause a denial of service for the average consumer. A botnet that small is well within the capability of a "script kiddie," or other person with malicious intent and a small amount of bitcoin.

In comparison, enterprises will enjoy more resiliency when it comes to withstanding a DDoS attack. However, as we have seen with DynDNS and Krebs on Security, it appears that there is no level of certainty, regarding remaining fully functional, when attacked by an adversary utilizing an IoT based botnet. As the number of IoT devices continues to grow, the number of potential candidates for IoT botnets also increases. Large enterprises no longer enjoy the sense of security that they once had.

5. CONCLUSIONS

IoT based botnet DDoS attacks continue to be a point of concern for individuals and enterprises alike. The Mirai botnet has been altered, manipulated, and improved to provide a platform capable of providing significant DDoS capability to the highest bidder. Our research takes a realistic approach to determining IoT device DDoS capability. Based on our research, the CPU and RAM resources of each individual IoT device do not have a significant impact on the maximum amount of bandwidth generated during a single target, single socket attack scenario. The established buffer size of a socket is the limiting factor in single target, single socket attacks.

This research provides insight into the anatomy of IoT botnets and the level of risk that individuals and enterprises face with respect to these devices. Our research is theoretical in nature, however, as we have seen with comparisons to attacks like the Krebs on Security attack, our research also has implications in the real world. To continue to make this research valuable to the industry and to provide defenders with a reasonable and reliable method of evaluating risk and attack capability, we must continue to expand our research in this area.

Our findings from this research have inspired many areas for future research. We intend to conduct future research in the following areas pertaining IoT device security and DDoS attack

capability:

1. Exploration into the effects of UDP and connectionless protocols on IoT botnet DDoS capability
2. The impact of CPU, RAM, and network capability of a single device attacking multiple targets directly and simultaneously
3. The impact of embedded architectures vs traditional architectures on attack capability
4. Identification of maximum attack capability for amplified and reflected attacks in IoT botnets
5. Identification of maximum attack capability in IoT botnets that include direct and indirect (amplified and/or reflected) attack methods
6. Quantifying the number of IoT bots in a botnet of unknown composition
7. Determining the resiliency of target systems during an attack and quantifying the number of devices a target system can withstand while remaining fully functional

Appendix A: Infrastructure Hardware Details

Host Details	
Client Make	Dell
Client Model	T750
Total RAM (GB)	6
CPU Make	Intel
CPU Model	Xeon E5507
CPU Cores	4
CPU GHz	2.27GHz
Operating System	Windows Server 2012R2
Hypervisor	Hyper-V

Server Details	
Server Make	Custom
Server Model	Custom
Total RAM (GB)	24
CPU Make	Intel
CPU Model	i7
CPU Cores	4
CPU GHz	2.67 GHz
Operating System	Windows Server 2012R2

Client Details	
Operating System	Ubuntu
Version	16.04 TLS
CPU	Variable
RAM	Variable

IP	Machine
10.0.0.1	Server
10.0.0.55	Host
10.0.0.56	Client
10.0.0.254	Switch

Appendix B: Python Scripts

Server Side:

```
#Script to establish a server side socket to test maximum bandwidth based on hardware resources
#Using a file to send data for an extended period of time
import socket
import os

#Variables
B_size = int(raw_input("Enter the buffer size:\n"))
Bind_port = int(raw_input("Enter the port number to connect on:\n"))

#Establish the server and listen for connections
def Socket_Server (Bind_Port, B_Size):
	s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
	s.bind(('', Bind_Port))
	s.listen(5)
	conn, addr = s.accept()
	print "Connected with", addr
	Total = 0
	while True:
		data = conn.recv(B_size)
		if not data: break
		print "Received Bytes: ", B_size
		Total = Total + B_size
		print "Total Bytes:", Total
	conn.close()

#Run the Server
print 'Prepare the server to be started'
os.system ('pause')
Socket_Server (Bind_port, B_size)
print 'Connections closed'
os.system ('pause')
```

Client Side:

```
#Script to establish a client side socket to test maximum bandwidth based on hardware resources
#Using a file to send data for an extended period of time
import socket
import os

#Variables
B_size = int(raw_input("Enter the buffer size:\n"))
Bind_port = int(raw_input("Enter the port number to connect on:\n"))
```

```
Server_IP = raw_input("Enter the IP Address of the Bandwidth_Server:\n")
Bandwidth_File = raw_input("Enter the name of your data document:\n")

#Create the Socket
def rocket_socket (Server_IP, Bind_Port, Bandwidth_File):
        fhand = open(Bandwidth_File, 'r')
        S_data = fhand.read()
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((Server_IP, Bind_Port))
        s.sendall(S_data)
        s.close()

#Run the Client
print "Verify the server side is running"
os.system('pause')
print "Beginning Bandwidth Test"
rocket_socket (Server_IP, Bind_port, Bandwidth_File, B_size)
print "Test Complete. Review data on the server."
os.system('pause')
```

Appendix C: Raw Data

Data	CPU Resources (GHz)															
	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%	60%	70%	80%	90%	100%	Average
Buffer Size (B)	0.11	0.23	0.34	0.45	0.57	0.68	0.79	0.91	1.02	1.14	1.36	1.59	1.82	2.04	2.27	
512	482.83	410.42	403.70	347.08	345.88	344.53	345.13	344.58	345.23	342.71	344.07	346.59	343.88	344.66	343.01	362.29
1024	753.06	653.25	646.67	672.06	637.16	610.21	605.24	614.38	616.23	655.50	646.30	627.04	618.64	612.51	616.11	638.96
2048	1495.04	1290.24	1280.00	1228.80	1218.56	1218.56	1218.56	1259.52	1259.52	1280.00	1259.52	1218.56	1259.52	1372.16	1331.20	1279.32
4096	3072.00	2662.40	3768.32	2662.40	2488.32	2549.76	2457.60	2682.88	2365.44	2529.28	2560.00	2488.32	2529.28	2344.96	2385.92	2636.46
8192	7321.60	4874.24	5713.92	4843.52	4771.84	4689.92	5376.00	4833.28	4904.96	4935.68	4945.92	4945.92	4710.40	4761.60	4403.20	5068.80
16384	10905.60	10188.80	9728.00	9318.40	10219.52	9512.96	9728.00	9502.72	9625.60	9625.60	9410.56	9553.92	9738.24	9328.64	9226.24	9707.52
32768	22282.24	19015.68	19517.44	18769.92	18647.04	18944.00	18872.32	18984.96	18595.84	18780.16	17868.80	18585.60	19087.36	19425.28	18083.84	19030.70
65536	40048.64	36884.48	36321.28	34385.92	33935.36	34416.64	35164.16	37181.44	42127.36	35706.88	36075.52	35584.00	40325.12	38256.64	40048.64	37097.47
	Bandwidth Observed (Kb/s)															

Data	RAM Resources (MB)														
	10%	15%	20%	25%	30%	35%	40%	45%	50%	60%	70%	80%	90%	100%	Average
Buffer Size (B)	409.60	614.40	819.20	1024.00	1228.80	1433.60	1638.40	1843.20	2048.00	2457.60	2867.20	3276.80	3686.40	4096.00	
512	345.44	346.22	345.10	349.87	343.11	344.82	346.17	345.22	345.25	343.22	344.20	350.10	345.90	343.10	345.55
1024	613.55	610.57	612.00	611.99	615.17	612.67	613.55	611.10	625.30	662.41	659.11	612.31	613.70	615.16	620.61
2048	1310.72	1313.72	1317.25	1313.09	1311.11	1313.47	1311.49	1312.99	1228.80	1274.10	1333.90	1317.10	1311.37	1313.66	1305.91
4096	2426.88	2499.12	2429.11	2417.34	2501.11	2400.84	2412.17	2437.64	2488.32	2497.33	2333.55	2500.00	2429.40	2501.30	2448.15
8192	4126.72	4200.10	4129.15	4201.71	4100.63	4101.33	4122.10	4206.30	5017.60	5050.80	4207.40	4127.44	4126.66	4100.63	4272.76
16384	9277.44	9270.11	9276.10	9314.64	9300.98	9207.47	9255.19	9236.50	9502.72	9500.44	9276.66	9274.30	9177.50	9255.11	9294.65
32768	17991.68	17999.34	17995.44	17876.34	17888.20	17997.60	18004.15	17852.46	17899.52	17774.30	17999.41	17991.66	17944.90	17898.97	17936.71
65536	34938.88	34200.77	34732.25	34741.95	34900.11	34918.80	34917.77	34219.50	33792.00	34341.20	34744.11	34993.54	34732.50	34918.44	34649.42
	Bandwidth Observed (Kb/s)														

REFERENCES

[1] “Internet of Things (IoT): number of connected devices worldwide from 2012 to 2020 (in billions)”, Statista.com, 2017. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide

[2] “Making Sense of the Last Month of DDoS Attacks”, F5.com, 2016. [Online]. Available: https://f5.com/about-us/blog/articles/making-sense-of-the-last-month-of-ddos-attacks-22608

[3] D. Peraković et al, "Analysis of the IoT impact on volume of DDoS Attacks," in XXXIII Symposium on New Technologies in the Postal and Telecommunications Traffic , Belgrade, 2015.

[4] E. Alsaadi and A. Tubaishat, "Internet of Things: Features, Challenges, and Vulnerabilities", International Journal of Advanced Computer Science and Information Technology (IJACSIT), vol. 2015, no. 4, pp. 1-13, 2016.

[5] E. Ronen and A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case of Smart Lights," in 2016 IEEE European Symposium on Security and Privacy, Saarbrücken, 2016.

[6] D. Miessler and C. Smith, "OWASP Internet of Things Project", OWASP, 2016.

[7] B. Stone-Gross et al, "Analysis of a Botnet Takeover," University of California, Santa Barbara, 2011.

[8] “DNSSEC and DNS Amplification Attacks”, Microsoft.com, 2012. [Online]. Available: https://technet.microsoft.com/en-us/security/hh972393.aspx

[9] “What Is a Socket?”, Oracle.com, 2015. [Online]. Available: https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html

[10] “Network Buffers and Memory Management”, Linux Journal, 1996. [Online]. Available: http://www.tldp.org/LDP/khg/HyperNews/get/net/net-intro.html

[11] “RFC 791 - Internet Protocol - IETF Tools”, IETF Tools, 1981. [Online]. Available: https://tools.ietf.org/html/rfc791

[12] “HTTP Methods: GET vs. POST”, W3Schools.com, 2017. [Online]. Available: https://www.w3schools.com/tags/ref_httpmethods.asp

[13] “Apache HTTP Server Version 2.4”, The Apache Software Foundation, 2017. [Online]. Available: https://httpd.apache.org/docs/2.4/mod/core.html