Investigation of Vulnerabilities with Monitoring Tools

Kazi Zunnurhain, Ankur J Patel and Mairura James
Kean University union, New Jersey
Emails: {kzunnurh, patelan5, jamesmai} @kean.edu

Abstract-
Our research is on monitoring traffic features in diverse network areas and internet of things (IoT). This research investigates existing protocols and mechanisms to secure communications between IoT and dissimilar network areas detecting threats and vulnerabilities. Network monitoring and estimation have turned out to be increasingly critical with cutting edge technologies. A denial of service (DoS) attack can be easily implemented with an assortment of tools and configuration. A denial of service (DoS) attack is generated when enormous amount of invalid network packets are send toward a target machine while the machine is exhausted by trying to validate them. Since there is not a solitary answer for DoS, this attack had propelled through the efficient, circulated and remotely controlled network with compromised personal workstations (also known as bot), so that they can be utilized for sending substantial volume of constant and concurrent invalid requests toward the target system. We used agents running on monitored systems like Zabbix, Wi-Fi network monitor, Nedio, Cacti, Angry IP Scanner etc. Though in this research study we have only utilized Wi-Fi network monitor and *Angry ip scanner*. We will also use different DoS generating tools such as Low Orbit Ion Cannon (LOIC), eXtreme Orbit Ion Cannon (XOIC) or HTTP Unbearable Load King (HULK) for DoS attack generation. Then we will compare the impact factor of these DoS attack tools and propose a resistance mechanism against DoS attack. The learning outcome of this investigation would be to choose the proper monitoring gear for a new network and development of prevention scheme. Also learn about penetration testing on a network system in the presence of diverse traffic.

CSCE Concept-• Network adapters → Network • Networks → Denial-of-service attacks; Data center networks; •Secure communication and privacy → Distributed systems security.

Keywords- IoT, Wi-Fi network monitor, Vulnerabilities, security, Denial-of-Service attacks

ACM Reference format

Kazi Zunnurhain, Ankur J. Patel, Mairura James, 2018. Investigation of Vulnerabilities with Monitoring Tools. *ACM J. Comput. Cult. Herit*

1. Introduction

The internet of things came to existence in 1999 at Massachusetts Institute of Technology (MIT). IoT concept refers to the networked interconnection of computers, devices, objects or tools. The IoT can be classified into communication, architecture technology and design methodology. In this paper we explore several network monitoring tools' traffic features involving IoT. Network monitoring is subset of network management that monitors and tracks network activity for anomalies. Most cases it is a dedicated system that is being executed from command line or *graphical user interface* (GUI) applications. Several tools use packet internet groper (ping) to test the connection by sending Internet Control Message Protocol (ICMP) echo requests to the IoT devices and monitoring the network status. Ping allows you to determine if a device is available for communication on TCP/IP network. We can also ping to know the NetBIOS name, the IP address of the computer under attack (with the ifcong or ipconfig command). *ipconfig* mostly used on windows that assist in modification of DNS server address, subnet mask, IP address, where *ifconfig* is for Linux platforms.

Now, DOS *a.k.a* denial of service attacks simply defined as a prevention of intended user from accessing a legitimate service. After this attack security is being compromised, an adversary overwhelms a system with enormous amount of invalid service requests. But due to the policy of TCP, the server is engaged in checking the validity of those illegitimate requests and eventually engaging its resources and resulting starvation of legitimate services. The adversary successfully floods internet with specific traffic (such as TCP, UDP packets) which confuses the server security in denial of services. Hence (i) website crashes, (ii) resulting system slowdown, (iii) connection lost, such as status code 404: declaring html file not found and many other consequences. A gateway router can be used to identify the attack pattern for specific network packet or from source address. Hence several open source network monitoring tools can be utilized to assess a compromised network from various aspects few are mentioned below.

Wi-Fi network monitoring can be used on a network to conduct scanning and discovering the number of Wi-Fi connections. Then the tools find the IP address

range and the host in the connected network. After finding the IP address it safeguards from possible intruders by applying access control list (ACL-filtering of IP address. It will allow the scanning of network; the status can be exported locally. We can start the scanning process by either selecting (i) manual or (ii) automatic scan. From the scan we can find IP address, host name, mac address etc. It loads all the devices status on the network, and hence we can easily identify the devices that are connected to identical network.

Angry IP scanner works faster on IP address and its port scanner module simply pings each IP address to check its status, determines the mac address of hostname and scan ports. Features include computer names, work group names and the users on windows, IP address ranges, web server detection etc. The results can be saved to .csv, .xml, .txt, or as IP port list files. For maximum speed it uses multi-threaded approach and separates scanning threads for each scanned IP address.

LOIC an open source DoS attack application was developed by praetox tech which was basically to be used on stress testing and pen testing on newly configured servers. The canon application runs on C # and JavaScript. LOIC targets server using TCP, UDP or HTTP packets in the aim of disrupting services on the target. It turns your network connection to firehose, floods the victim server with predetermined network packets (TCP, UDP or HTTP).

Rest of the paper is organized as follows section 1.1 is motivation, section 1.2 is Hypothesis we will discuss about our predictions that are to be used in monitoring tools, section 2 is related works, section 3 is experimental setup, section 4 is experimental results and section 5 describes our proposed architecture DPPN, and section 6 is conclusion and our future plans.

1.1 Motivation

Technology is developing and improving every day, and hackers are also introducing new techniques of cyberattack. Our aim is to advise different approaches of building a stronger defense mechanism as the adversaries exploit the vulnerabilities of a system. Hence in this research we will involve various cyberattacks to identify the after effects in a network system or a website and assess through previous mentioned monitoring tools.

A. Security

The major concern of IoT is safety, the need for security, how information is being passed from one channel to another without intrusion. This area is striving for safeguarding connected devices and networks. The more devices are connected online the more we introduce vulnerability.

B. Privacy

Privacy can be compromised due to integration of devices in the network environments without being aware of vulnerable issues. This is becoming more prevalent when it comes to clients' communication or tracking devices that are connected to public WIFI.

C. Confidence

To be able to assure our clients that they can trust and rely on us, if we can confirm transfer of data with integrity of data.

D. Cost

Our hypothesis is if we find too many unauthorized devices connected to public Wi-Fi then it may introduce unexpected cost for the victim.

1.2 Hypothesis

The vision is to provide optimum solutions and security to our potential clients. We plan on generating the attack in a public network from an external unit. Then we will generate sequences of attacks depending on number of threads, number of packages and multiple occurrences with different time intervals. We believe these approaches will provide us with sufficient results to determine how a device will behave in a compromised public network. We will investigate to find the transgression point of the victim where it is not processing any legitimate requests (TCP or HTTP requests). At breaking point we will also be able to measure the amount of packages and threads causing the disturbance in the victim machine. Later on we will propose an architecture, where we plan on utilizing the findings from our experiments to prevent against an external DoS in a public network. We also believe due to different protocol types the impact of attack will vary, hence we will conduct our investigation based on multiple network protocols.

2. Related Works

Many research studies had conducted surveys on internet of things and general privacy related to internet. Survey knowledge is required to understand the vulnerabilities and possible security holes to protect from intruders [1]. In this study the authors investigated key management systems for sensor networks, talked about Wi-Fi and traffic management [2]. In another study the authors studied unification of internet of things architecture. They surveyed about the ability to measure, infer and understand the prominent factors of IoT environment [3]. Server assisted key establishment protocol for WSN proposed to offload heavy computation for finding an

RSA key pair to untrusted servers [4]. Another study on forecast: the internet of things worldwide looked at finding solution in supply driven market [5]. A similar study proposed a protocol for carrying authentication for network access. Network layer transport for Extensible Authentication protocol (EAP) to enable network access authentication between clients and access network [6]. Authentication for network access engages UDP based EAP lower layer that runs between the EAP peer and EAP authentication [7]. For securing communication it deals with WSN and the receiver is able to verify that the sensor data was generated by trusted nodes [8]. Security, privacy and trust in internet of things should be the future goals security researchers. This focus with satisfaction of security and privacy requirements such as confidentiality and authentication which dealt with wireless sensor network (WSN). In [9], a research study on addressing privacy and security without a comprehensive implementation for prevention against security threats. Existing wireless networking systems and protocols (Wi-Fi, Bluetooth, near field communication, and GPS) to facilitate those objectives. In turn, this reliance will fuel the creation of even more big data. Many of these technologies and capabilities will eventually operate in the background of consumers' lives and be almost invisible to them. Their focus was only based on using Bluetooth and the near field communication as compare to our study, investigation with Wi-Fi for collecting network data from diverse community with open access to internet. In [10], twenty security considerations for cloud supported internet of things were discussed. This work focused on TLS as a security feature for cloud-providers, and can be used to assure the confidentiality and integrity of communications between IoT devices and cloud providers.

With a general view for making secure communication widely in public places, there is recent study. 'Twenty security consideration for cloud supported internet of things'. On enabling TLS on protocol stacks to better support the requirements of IoT devices, in terms of complexity and resource requirements. To mention few examples such as DTLS (Datagram Transport Layer Security), datagram-oriented protocols such as UDP (User Datagram Protocol), and LLCPS (Logical Link Control Protocol) utilizes TLS over the near field communications. Depending on the deployment, architecture and interfaces to cloud services, few methods like encryption and hashing could facilitate new ways of securing 'thing'-cloud interactions. They did not use HTTP (Hypertext Transfer Protocol) which we did to launch DoS attack from LOIC. Another work talks about routing attacks in the IoT devices [11]. Routing attacks can be spoofed, altered, attract network traffic and shorten network paths. The intension of the attacker is to gain access of a network router to control all the packets crossing through the network. The focus on this research was more on smart toys and fitness apps. They consider more web security aspects than network security.

3. Experimental Setup

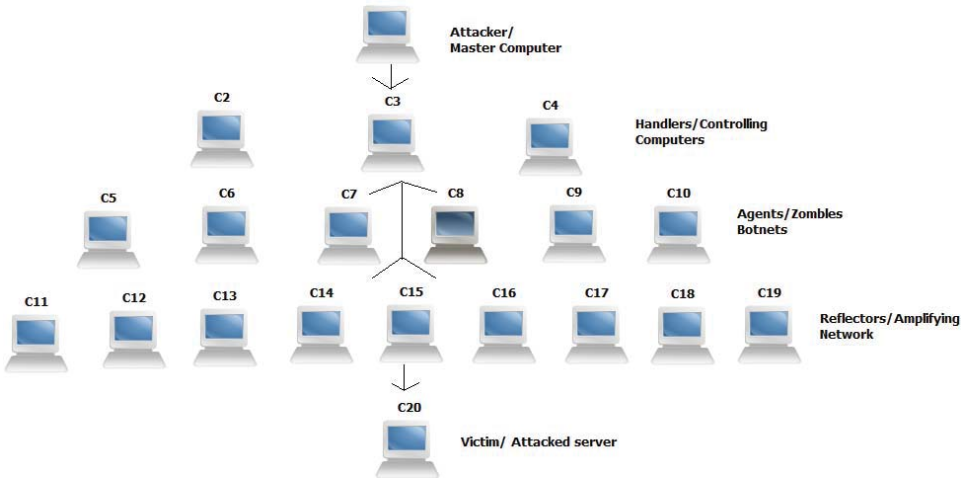


Fig 1: DDoS Attack Topology

In the above fig the architecture showing Distributed Denial of Service (DDoS) attacks. There are five segments, (1) attacker master computer, (2) handlers/controlling computers (3) Agents /zombies botnets (4) Reflectors /amplifiers and (5) victim/compromised server. Two of them are the major role players – the attacker (main PC) from where the attack was initiated and the

victim/compromised server which is at the bottom in the topology. Attack is propagated through handers and amplified by zombies and reflectors, which finally floods the victim, shutting down its resources.

The correspondence between the Attackers and the Handlers, and between the Handlers and the agents is the control traffic of the system. The correspondence between the Agents and the Victims is the surge traffic.

An attacker likes to masquerade its actual IP address behind spoof IP address for two noteworthy reasons: attacker does not want to be traced back. The second reason is the nature of execution, where propagation of attack takes place through several intermediate devices. The attacker is revoking any attempt from the victim to filter out the flooding.

3.1 Low Orbit Ion Canon

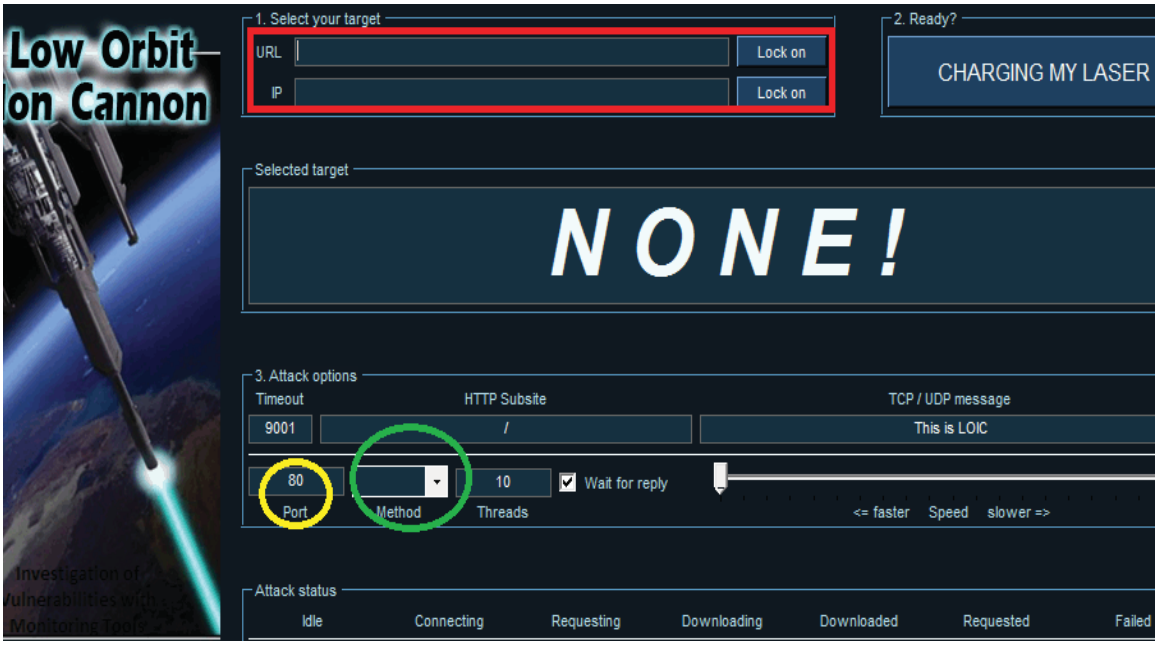


Fig 2: LOIC Graphical User Interface

LOIC is an open source network stress testing and DoS launching application. LOIC was at first created by Praetox Technologies, later was distributed for general population. It is now provided with few open source network applications. This LOIC application can perform a basic DoS attack by sending a large scale UDP, TCP or HTTP requests to the target server. It is a simple apparatus to launch a DoS attack with. No need for any fundamental learning about hacking is required to operate this tool. The major information a client needs to know for using the tool is the website URL of the target/victim. It also takes IP address. As provided in the figure above a screenshot of LOIC is the GUI. It offers the client textboxes for URL, and IP address of the victim along with few attack alternatives depending on packet types (such as TCP, UDP and HTTP), counting port, and number of threads. To investigate the output of LOIC, four analyses were setup. The initial four investigations were with 10, 15, 20 and 99 threads consecutively for TCP protocol.

4. Experimental Results

In this section we illustrated our experimental results with help of charts. We introduced popular DoS attacks for the investigations.

4.1 Investigation of the Attack

UDP Attack: User Datagram Protocol the strategy is to send enormous amount of UDP packets. It has port 80 as the default elective picked, it can be changed as per the need.

TCP Attack: This strategy is similar to UDP attack. Utilizes Transmission Control Protocol (TCP) packets. It also has port 80 as the default port, which also can be changed as per the need.

HTTP Attack: In this attack, the application sends HTTP requests to the target server. This attack is also known as HDOS or XDOS (HTML Denial of Service or XML Denial of Service) attack.

So, based on TCP, UDP, HTTP attacks we have several observations. Fig 3 defines TCP attack with a public website URL (Uniform Resource Locator) which is described in section 4.2 in details.

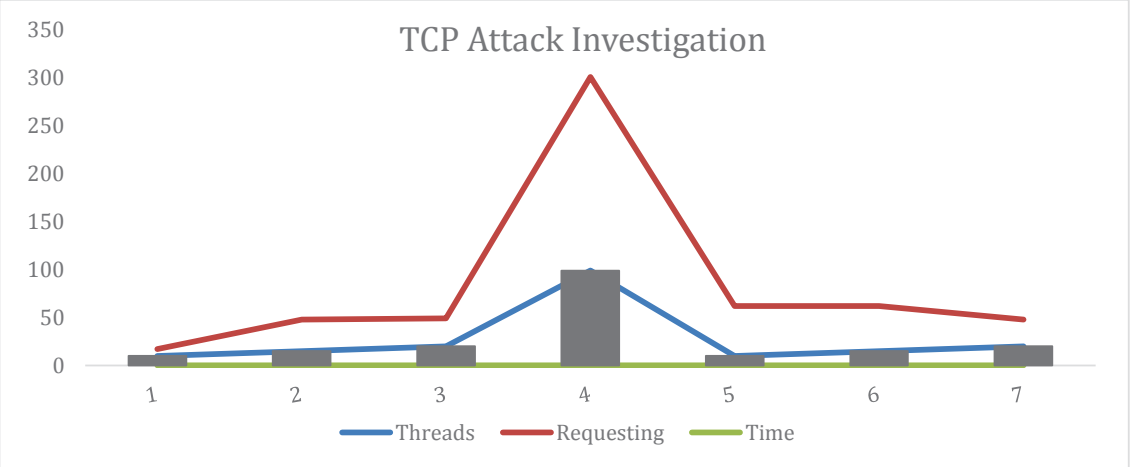


Figure 3: TCP Attack Investigation

4.2 Attack with TCP

Fig 3, demonstrates time along the horizontal axis, number of threads in blue and number of requesting packages in red (numerically along vertical axis).

The experiment was conducted by launching number of threads and response time from target website was recorded. Our observations are mentioned below:

I. In the graph, as the number of threads increases, the response time of the website also increases.
II. We tried the testing with the different number of threads attacking on the target website.
III. Maximum 99 of threads were launched and the response time of site is 300 seconds reaching the peak in the graph.
IV. We used minimum 10 threads for the attack and the response time is twice the number of threads. When the attack was generated between 15 and 20 threads it took about 50 seconds to respond.

The prominent segment in this attack is the variation in response on active users. It also happened again in the second iteration with 10 threads which took 54 seconds to respond. For the 15 and 20 threads, the response time of the website was also changed. Eventually, DoS attack will be alarming if the attackers can launch enough threads to supersede the bandwidth of the website. Hence, web server resources will be exhausted for being engaged in validating these illegitimate packets. We could not conduct these experiments in private sectors. But we plan on investigating security for private websites too. Figure 4 shows TCP attack with public websites.

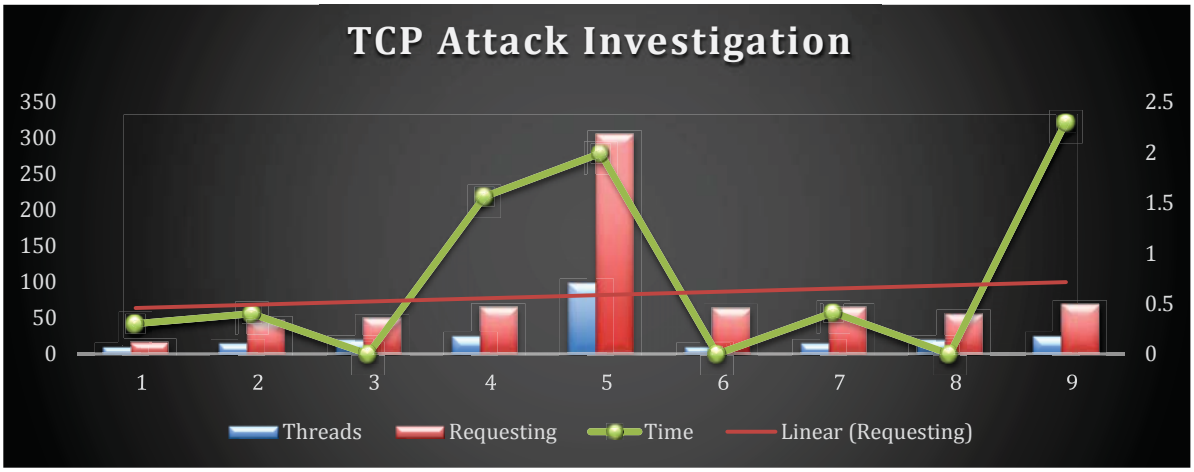
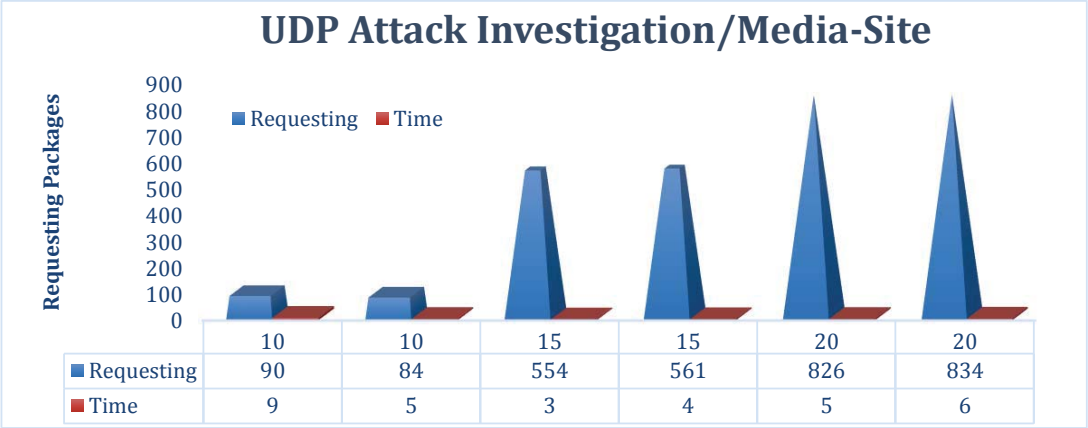


Fig 4: TCP Attack Investigation

The above chart showed the number of threads launched and the time taken by the victim site to prevent the DoS attack. In the chart, as the number of threads increases, the response time of the website increases.

We generated number of threads. Attack was launched with 99 threads and the response time was 305 seconds. Also based on HTTP requests

We launched HDOS on multiple public websites. The minimum number of threads for attack was 10 and here also the reaction time is twice the quantity of threads. Also launched attack with 20 and 25 threads and took only 1.3 second to respond. Hence it is clear that attack at the beginning is the one makes the largest impact in public websites, due to unaware of the incoming.



	10	10	15	15	20	20
Requesting	90	84	554	561	826	834
Time	9	5	3	4	5	6

Fig 5: UDP Attack Investigation/Media-Site

In Fig 5 we have conducted a UDP attack targeting a website running on similar network. The diagram showed the number of threads launched and the time taken by the victim media site to prevent.

Our observations are: (i) 3 different threads (10, 15, and 20) from port 80 requested packages using LOIC attack on a media site using HTTP. (ii) With 20 threads, 834, the largest numbers of requesting packages were launched and took 6 seconds to respond. On the other hand, 84 packages used 10 Threads to launch an attack with HTTP. It is noticeable that using 15 threads (554 and 561 requesting packages) the respond times were similar.

5. DPPN

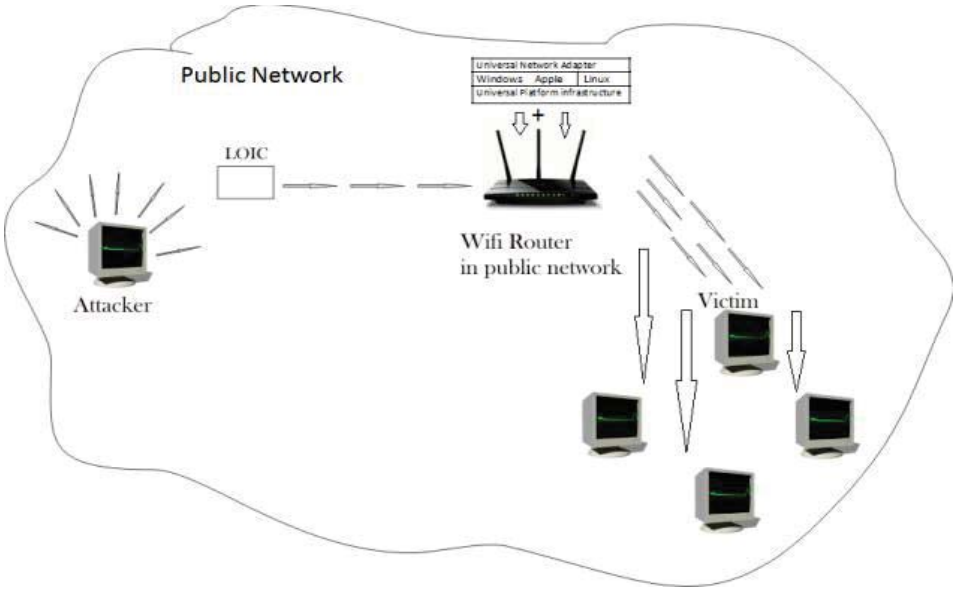


Fig 6: Dos prevention in public network (DPPN)

Fig 6 highlights our proposed architecture to defend against DoS attack. We plan on hosting the application architecture on the default gateway or the border router as shown in the figure above. Generally, a DoS attack is propelled from a solitary source which eventually uses a bot or botnet to masquerade its identity behind legitimate IP addresses. In our study we considered the attack being launched from an unsecure public network targeting the victim and eventually refraining the system from processing legitimate requests. Our web application implemented in the router will start the mitigation process of spoof packets by measuring the network packets passing through the router.

To describe our architecture, we plan on developing a universal platform which will support Windows, Apple and Linux operating systems. By supporting all these diverse OS we will easily be able to detect the platform of attack generating source to avail us with more preparation for defense. Why? Because of our universal network adaptor. This adaptor will detect the individual platform of the adversary in the network. An alarm will be generated immediately if the gateway detects the number of packages or the number of threads exceeds certain threshold. Then the router will send an *echo* message to the clients connected to that router due to a possible ongoing attack. If the clients voluntarily do not disconnect from the public network then the router will kill the connection within 5 minutes.

Now there will be a monitoring unit to measure the number of traffic on the router in a public network. We also plan on providing the public network provider the privilege of switching off/on the public access depending on the alarm generation. Hence an echo message will be send to the provider as well. These practices however are not constrained to strict packet filtering, incapacitating unused network administrations, customary refreshing and IP address changing of programming on servers. Eventually we want to implement a tracer to trace back to the adversary, but a small computing unit of router might not be able to trace back all the time due to the attack scenario. By that we are referring the possibility of large botnet involvement, amplification of attack from a masquerader behind unused legitimate IP address within the similar subnet of the ISP or may be the public network. In such situations tracing back to the original attacker is not possible for such a small unit of CPU within the gateway router. Then we have to think of hosting DPPN in the ISP router (usually large scale network servers).

In future we plan to include an additional unit to pair with the DPPN client privilege and take precautions ahead of time. DPPN-Client will be both mobile and web app and can be installed easily.

6. Conclusion

As mentioned in previous sections our intensions were to investigate the impact of denial of service attacks with multiple types of network protocols and deduce an approach to prevent against such attacks. In current world most of the tech companies are investing on security to prevent such type of attacks. As the largest market is based on SOA (Service Oriented Architecture) where uninterrupted and efficient service is the prime objective for the providers.

Our research study regarding denial of service attack reveals a noteworthy fact about number of threads, response time, package number and several other aspects of detecting penetration threshold of a webserver in a public network. With a large scale attack with either TCP, UDP or HTTP from a simple penetration testing open source tool like LOIC; can blemish the regular activities of a distributed system where resources are sharable and scalable. Due to the resource sharing nature, a distributed system offloads to its nearest available resource (server). Hence targeting only one server can result in flooding the whole system. Thus the plan is to utilize the results from this study to implement the proposed architecture (DPPN) to prevent DoS attack in public network.

In our next work we will implement the architecture we proposed to prevent against threads of packages destined from an unknown IP address. Measuring the packages will provide an extended approach to predict a denial of service attack from an open source application. We also learned the impact is large in the middle of an ongoing attack, hence the architecture will monitor an external device for longer period of time before validating and allowing the incoming network packets in the system.

References

[1] Roman, Rodrigo, et al. "Key management systems Things." *Computers & Electrical Engineering* 37.2 (2011): 147-159.

[2] Kosmatos, Evangelos A., Nikolaos D. Tselikas, and Anthony C. Boucouvalas. "Integrating RFIDs and smart objects into a UnifiedInternet of Things architecture." *Advances in Internet of Things* 1.01 (2011): 5.

[3] Nguyen, Kim Thuat, Maryline Laurent, and Nouha Oualha. "Survey on secure communication protocols for the Internet of Things." *Ad Hoc Networks* 32 (2015): 17-31.

[4] Middleton, Peter, Peter Kjeldsen, and Jim Tully. "Forecast: The internet of things, worldwide, 2013." *Gartner Research* (2013).

[5] Forsberg, Dan, et al. *Protocol for carrying authentication for network access (PANA)*. No. RFC 5191. 2008.

[6] Ohba, Yoshihiro, et al. "Protocol for carrying authentication for network access (PANA)." (2008).

[7] Raza, Shahid, et al. "Securing communication in 6LoWPAN with compressed IPsec." *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*. IEEE, 2011.

[8] Sicari, Sabrina, et al. "Security, privacy and trust in Internet of Things: The road ahead." *Computer networks* 76 (2015): 146-164.

[9] Thierer, Adam. "The internet of things and wearable technology." *Mercatus Center or George Mason University* (2014).

[10] Singh, Jatinder, et al. "Twenty security considerations for cloud-supported Internet of Things." *IEEE Internet of Things Journal* 3.3 (2016): 269-284.

[11] Mattern, Friedemann, and Christian Floerkemeier. "From the Internet of Computers to the Internet of Things." *From active data management to event-based systems and more*. Springer, Berlin, Heidelberg, 2010. 242-259.

[12] Dastjerdi, Amir Vahid, and Rajkumar Buyya. "Fog computing: Helping the Internet of Things realize its potential." *Computer*49.8 (2016): 112-116.

[13] Iliofotou, Marios, et al. "Network monitoring using traffic dispersion graphs (tdgs)." *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007.

[14] Schuba, Christoph L., et al. "Analysis of a denial of service attack on TCP." *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*. IEEE, 1997.
for sensor networks in the context of the Internet of

[15] Busschers, Rik. "Effectiveness of defense methods against DDoS attacks by anonymous." *University of Twente* (2010).

[16] Peng, Tao, Christopher Leckie, and Kotagiri Ramamohanarao. "*Survey of network-based defense mechanisms countering the DoS and DDoS problems." ACM Computing Surveys (CSUR) 39.1 (2007)*

[17] Chao-yang, Zhang. "DOS attack analysis and study of new measures to prevent*." Intelligence Science and Information Engineering (ISIE), 2011 International Conference on. IEEE, 2011.*

[18] Schuba, Christoph L., et al. "*Network protection for denial of service attacks.*" U.S. Patent No. 6,725,378. 20 Apr. 2004.

[19] Loukas, Georgios, and Gülay Öke. "Protection against denial of service attacks: A survey." *The Computer Journal 53.7 (2010): 1020-1037.*

[20] Singh, Sumeet, et al. "Detecting public network attacks using signatures and fast content analysis*." U.S. Patent No. 8,296,842. 23 Oct. 2012.*

[21] Anderson, Tom, Timothy Roscoe, and David Wetherall. "*Preventing Internet denial-of-service with capabilities*." ACM SIGCOMM Computer Communication Review 34.1 (2004): 39-44.

[22] Wang, Haopei, Lei Xu, and Guofei Gu. "*Floodguard: A dos attack prevention extension in software-defined networks*." Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on. IEEE, 2015.