International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

Attack Patterns on IoT devices using Honey Net Cloud

A R Jayakrishnan, V.Vasanthi

Abstract: ***Due to the superfluous growth of IoT devices in the current digital world, where lots of devices are becoming smart by being able to connect internet with many smart features, IoT devices have become the main target of cyber-attacks for the hackers these days. Since the IoT devices are very light in terms of processing power and memory, it has become an easy target for hackers to intrude in to the network easily. The file-less attacks, that usually doesn't require any files to be downloaded and installed gets bypassed by anti-malwares. Very less effort has been put to learn the characteristics of attack patterns in IoT devices to do the research and development efforts to defend against them. This paper deep dives to understand the attacks on IoT devices in the network. HoneyNetCloud has been made with four hardware honeypots and hundred software honeypots setup that are meant to attract wide variety of attacks from the real world. Huge range of data was recorded for the span of 12 months. This study leads to multifold insights towards developing the IoT Network Forensics Methodology.***

Keywords: IoT attack, Network Forensics, Honeypots, Digital Security, HoneyNetCloud, Intrusion.

I. INTRODUCTION

Internet of Things has rapidly enlarged popularity across a wide range of areas in home automation, industrial smart sensing and control etc [2]. In general, most of today's IoT smart devices employs the Linux (eg. Raspbian and OpenWrt) for its programmability and prevalence [1]. Connected things however incessantly emit stream of data as part of their communication in the network. Meanwhile, the cyber-attacks volume also increases rapidly. This paper therefore concentrates on Linux based IoT things and the intrusions targeting them. In general, there are two types of attacks on IoT things: File-less attacks and Malware based attacks [3].

Malware based attacks are very common these days. Examples like PNScan, Mirai and Mayday have been broadly acknowledged in IoT networks. Popular and commonly used websites like Twitter, Github were inaccessible for many hours during Oct 2016, when their DNS provider attacked by Mirai under DDOS attack. This was infected over 1.2 million IoT things over the world [4].

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author
A. R Jayakrishnan*, Research Scholar, RCAS, Bharathiar University, Coimbatore, Tamilnadu. India.
V.Vasanthi, Assistant Professor, Dept of IT, SKASC, Coimbatore, Tamilnadu, India.

Such alarming incidents created high alerts among investigators and researchers about malware based attacks on IoT systems; Their characteristics have been extensively studied over past few years and hence effective defence solutions were developed. For example, the hash MD5 or SHA-n of a malware's binary file can be calculated to fingerprint the IoT malware, and such fingerprints could be shared with the community over VirusTotal1. Static and dynamic analysis can be applied for the malwares those are not been fingerprinted to regulate their malevolence [5].

File-less attacks which is also called as non-malware attacks or zero footprint attacks on IoT things vary from malware based attacks, where they need not download and run malware files to infect the IoT devices; but instead they take benefit of existing vulnerabilities on the victims' devices and intrude in to the network. For the past few years, progressively more and more file-less attacks were reported [6]. McAfee Labs reported that file-less attacks ramped up by 430% by 2017 [7]. For the PCs, laptops and servers, defending file-less attacks can be easily done by using anti malware tools, firewalls and antivirus apps [8]; however, since due to the limited computing, storage and memory capacity for the vast majority of IoT things, such antivirus solutions are not suitable. As the result, file-less attacks stance substantial threats to the IoT ecosystem. Remember that the IoT things are often deployed in private and sensitive environments, such as banks, military areas, government offices, private residences and healthcare centres. There is very limited visibility into their appearances and attack vectors at the moment, which hampers research and development efforts to innovate new defence to battle file-less attacks [9]

A. Methodology

In this research, I've used honeypots, which are effective to capture the unknown network attacks by attracting intruders to the network [10]. Initially, four hardware honeypots were setup. Each of them was attached with a remote control power adapter, which has the ability to reset the honeypot when it is compromised to attacks. These indeed provided valuable insights in to the IoT attacks at the same time they are very expensive to setup too. In order to attract more and more real-time attacks, I've decided to scale up the number of honeypots connected in a much cheaper and effective way by installing virtual honeypots. Hundreds of software virtual honeypots were setup that are meant to attract wide variety of attacks from the real world. With the ever growing cloud setup around the globe with various providers, it seemed to be a possible solution for me in my approach as its both economical and practical to implement.

However, there are a few practical concerns to this. Firstly, the software honeypots should behave likewise to the actual IoT devices, so that it won't miss any sort of attacks. Secondly, they should also be able to depict comprehensive data of the interaction methods, so that it will become easy to categorize the difficult to track attacks. Finally, they have to go along with the diverse policies imposed by the cloud providers, so that limitations of clouds will not really impact each other for the coverage of the study results. Hence, the design of the software honeypots was customized by incorporating the comprehensions composed from hardware honeypots, such as the encrypted command disclosure, kernel information masking and data flow symmetry monitoring. I've carefully selected eight public clouds to disperse hundred software honeypots, and group of the developed system is called HoneyNetCloud. All of the software honeypots in the HoneyNetCloud have been employed with DD-WRT, which is the best and most popular Linux based OpenSource firmware for IoT devices [11] and is also appropriate for customization.In comparison with the hardware honeypot, a virtual honeypot attracts 35% lesser suspicious connections and 37% lesser attacks on an average basis. However, the maintenance cost for the software version is 12 times less compared to that of the hardware honeypot. Interestingly, all the attacks trapped by the hardware honeypots have also been captured and trapped by the HoneyNetCloud, but this is not the other way round. This proves the effectiveness of my design in practice.

B. Implications discovered from the findings

Honeypots were run for a span of one year (i.e 02/Oct/2018 to 01/Oct/2019). The suspicious connections to the honeypots were around 286 million in the time span of one year. Out of which, 34 million (11.88%) successfully logged and able to attack. The rest of 88.12% observed suspicious could not intrude to the honeypots, since they fail to crack the passwords. Among the successful intrusion attaks, 2.5 million were classified as file-less attacks and rest of them as malware attacks.

- **New security challenges by the file-less attacks could be discovered and thus leads to propose new defence directions**. Most of the captured attacks were using shell commands, hence they were all detectable by auditing the command history in the IoT devices. Most of the IoT devices use a readonly filesystem so that the malware based attacks could reduce, however this unpredictably increases the difficulty of detecting the file-less attacks. This presents a new security challenge in IoT networks. Fundamentally, it is due to the compromise among the auditability of file-less attacks and the security against attacks of malware types. Moreover, it has been observed that majority of file-less attacks (i.e 69.1%) are invoked through kill, rm, passwd, ps etc which were all enabled in the honeypots by default. For IoT devices, not all of these commands need to be enabled, which eventually creates opportunity to reduce the surface for attacks by disabling them.
- **In order to determine device authenticity, various types of information are being used by the attackers**. It has been observed that 9231 attacks executed commands like lscpu to extract very sensitive information. This is derived from the measurements taken from the hardware honeypots setup as part of the implementation. Whereas in the software honeypots HoneyNetCloud, an average of 6.9% fewer attacks were captured by the honeypot hosted on AWS than other public clouds. This could be because AWS reveals IP range of all its VM instances. Some of the malwares like Mirai will not infect specific IP range [12]. These comprehensions were incorporated to improve the effectiveness and reliability of the design of HoneyNetCloud.
- **More file-less attacks due to the unique types of attacks.** Most difficult, powerful and dangerous threats are observed to be file-less attacks. 41.2% of the attacks are considered to be gathering system information and/or performing activities shutting down firewalls, deactivating watchdog and stopping antivirus services etc. This is considered to be done so as to get clear way for consecutive attacks and also to allow more targeted attacks. The pattern of attacks shows that the most favourite attack by hackers is file-less since they are difficult to fingerprint and hence most suitable for cautious attack reconnaissance and preparations. In the attack metrics, there was also a targeted DDoS attack, that neither touches the filesystem nor run any shell commands, however it manipulates a flock of IoT devices and make the attackers not visible to victims. The anomalous pattern of outbound network traffic is the only indication of such attacks. It is highly promising for the existing host based tracking or monitoring system to catch it effectively.
- **Classification of eight types of attacks.** The detailed study of attack patterns and data captured for the duration of one year has provided the way to categorise the IoT attacks in to eight different types. Due to the lack of malware files and fingerprints, it was never an easy task to classify and also to identify the attack patterns. However, by carefully correlating the disclosed shell commands, recording the traffic flow, monitoring filesystems change and gathering online thirdparty reports, I was able to empirically classify the attack patterns.

II. IMPLEMENTATION

HoneyNetCloud – the implementation of the IoT honeypots includes hardware IoT honeypots and the software cloud based IoT honeypots. Structural overview of the HoneyNetCloud is depicted in the figure 1 below.

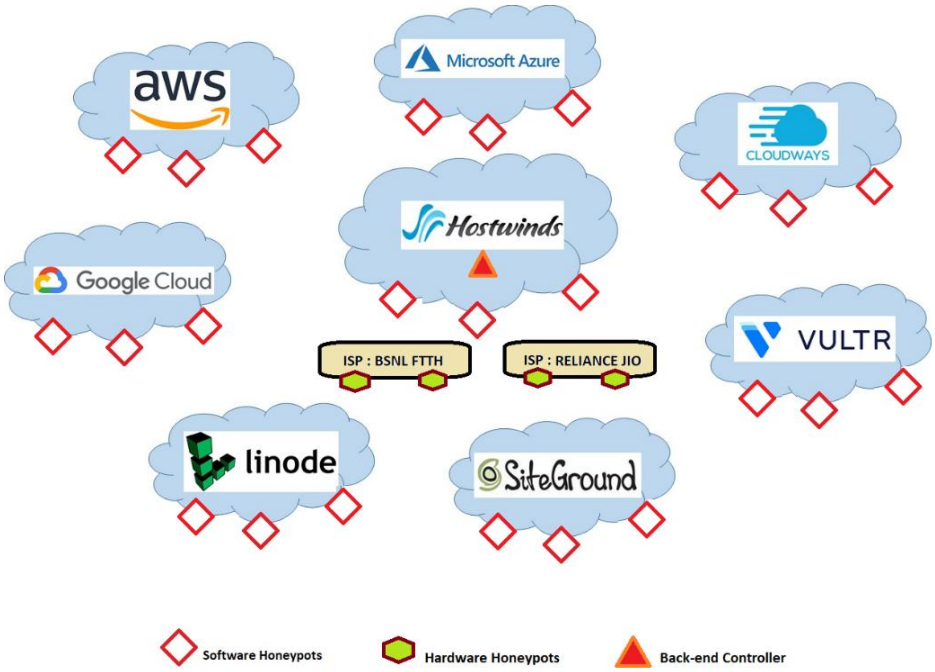


Fig. 1. Overview of HoneyNetCloud

A. Overview

Honeypot, as the name suggests is a trapping mechanism used to attract hackers and monitor the activities. It is considered to be an effective tool that helps to understand known & unknown type of attacks. They are widely deployed and are used by many on the internet [13]. It is not a production system which provides services, hence no one has really specific reasons to access it. Thus, communication packets coming in and out of Honeypots should be suspicious logically. However, this is slightly different in the case of public cloud since VM instances reports legitimate diagnostic data to the cloud providers. Those instances can be easily recognized and such instances were considered in the analysis done in this implementation. The IoT hardware honeypots were deployed in two different locations as shown in the below Table 1. Hundred numbers of software honeypots were deployed in various locations around the globe from public clouds that includes AWS, Microsoft Azure, Cloudways, Vultr, SiteGround, Linode, Google Cloud and HostWinds.

Table 1: Specifications of the hardware honeypot implementation

#	City	Device and Price	CPU Architecture	Memory	ISP
1	Kerala, India	BeagleBone, $67	1 GHz Cortex A8 Processor	512 MB	BSNL FTTH
2	Kerala, India	Linksys WRT54GL, $69	MIPS Little Endian 200 MHz	32 MB	BSNL FTTH
3	Bengaluru, India	Raspberry Pi, $40	1.2 GHz ARM Processor	1 GB	Reliance Jio
4	Bengaluru, India	NetGear R6120, $43	MIPS Big Endian 560 MHz	128 MB	Reliance Jio
*	For all centres	RCPA - Remote Control Power Adapter, $30			

B. Hardware Honeypots

My first step in this implementation was to setup the hardware honeypots. As listed in the Table 1, four hardware honeypots were installed at two locations in India on BeagleBone, Linksys WRT54GL, Raspberry Pi and NetGear R6120. The cost and the configurations were listed in the same table. For the duration of one year (Oct 2018 to Oct 2019), hardware IoT Honeypots could attract 15.4 million suspicious attacks. Out of them, 1.9 million suspicious connections could successfully log in and were successful in attacks. Malware based attacks were 0.85 million and 0.11 million of file-less attacks. The remaining 0.94 million could not be categorized because they did nothing after logging in. The cost involved in setting up and maintaining hardware honeypots are a bit expensive. However, they are effective in attracting the hackers.

The design of the IoT hardware honeypot installation is illustrated in the figure 2. It consists of three portions i) basic hardware things that includes NIC, RAM and CPU ii) OpenWrt OS and iii) services to attract intruders. Hardware honeypots tracks all the actions of attackers which include the typed commands or the programs executed by the hackers, and reports all of these to the data collector (as in Figure 2).

BusyBox, a software suite providing Unix utilities is used in the implementation. Shell script named ash is exposed to the attackers, so that all their operations can be recorded when they run the shell commands. The shadow file is modified and set the root password to root for making it more vulnerable for hackers. The Telnet service is setup on port 23 by BusyBox and the SSH service at port 22 by DropBear.

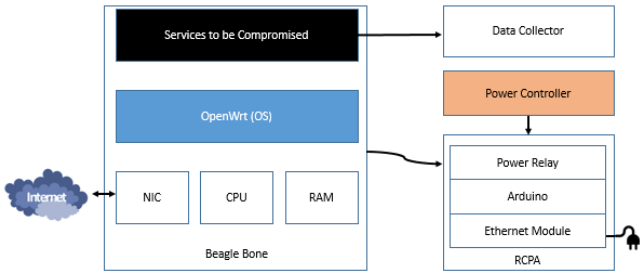


Fig. 2. Architecture of IoT Hardware Honeypot based on Beagle Bone

When an attack is identified that intrudes to the honeypot, threat information is immediately captured and the honeypot will be reset to original state. Initramfs is used to mount the root filesystem. Initially its loaded from SD card or flash memory and then loaded in to RAM. All the modifications to system state will get lost post reset. If the hardware honeypot is unresponsive or it doesn't report to data collector, then the honeypot is rebooted to bring it back to normalcy. Even though there is a watchdog to monitor and automatically reboot the honeypot, its normally been attacked by hackers using malware (for e.g Mirai) to disable watchdog. Hence, RCPA – Remote Control Power Adapter is built to physically reboot the honeypot as shown in the figure 2. It is made of Power relay, Arduino and Ethernet Module. MQTT protocol is being used by the RCPA [14]. When it gets the reboot command, it enables the power relay to switch off and then switch on the Hardware honeypot.

C. Software Honeypots

Setting up of hardware honeypots and maintaining them is very expensive compared to software honeypots. Hence, I've decided to scale up the number of honeypots to 100, so that more volume of attacks can be obtained for the research. For the one-year duration, all the 100 honeypots were run on eight public clouds namely AWS, Microsoft Azure, Cloudways, Vultr, SiteGround, Linode, Google Cloud and HostWinds. As of Oct 2019, there were 260 million records of suspicious attacks to the software honeypots. Out of this, 14.2% were successful in logging in and effective in attacks. There were 16.2 million malware based attacks and 1.8 million file-less type of attacks. 13.6 millions of attacks were not successful as nothing happened after logging in, hence they were not classified to any known types of attacks. Compared to the hardware honeypots, software honeypots are quite cheap as they need only low end hardware and the only thing needed was to rent the base level VM instances to run the software honeypots. Each VM instance host single software honeypots. So, hundred numbers of VM instances were setup with the configuration of single core processor of 2.2 GHz, 1GB RAM, 20GB of storage with 100 Mbps of internet bandwidth. Figure 3 depicts the geographical distribution of software honeypots on cloud.

Comparatively, software honeypots are easy to scale up than hardware. Since hardware honeypots require more resources and cost involved is high for setting up and maintenance, software honeypots need only software based solutions. The main impediments I faced during hardware deployment is the infrastructure dependencies. These were the reasons behind smoving on to the cloud based solution for setting up software honeypots so that the limitations faced during the setup of hardware honeypots were resolved. Cloud solution offers the most reliable, scalable and economical answer to a global scale implementation on a virtual infrastructure. While designing the software based IoT honeypots, my main intention was to get the reliability as same as that of hardware based honeypot that was setup before. The same was achieved in the software honeypots too. It has been observed that 9321 attacks were attempted to get sensitive data via software commands like *cat/proc/cpuinfo* and *lscpu*, such commands enable hackers to find out the software honeypots running in VM instances. The software honeypots setup was capable to forge the CPU/system information, making it lookalike real IoT devices. It was noticed that there were 192 attacks using commands for listing connected usb devices (like *lsusb*) so as to detect the potential honeypots, thus decided to enable common buses to make sure the fidelity.

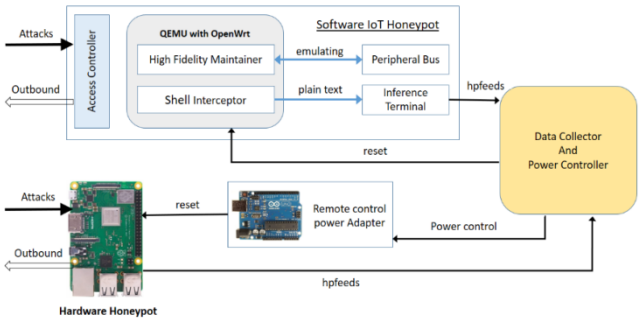


Fig. 4. Architecture overview of software IoT honeypot and new system HoneyNetCloud.

The architectural overview is illustrated in the figure 4, where the power controller and data collector are integrated with the hardware IoT honeypots too. Resetting capability of software honeypot in the power controller is also illustrated in the implementation diagram. There are mainly three modules in the software honeypots: Access Controller, Shell Interceptor cum Fidelity Maintainer and Inference Terminal.

Fig. 3. Geographical distribution of IoT Software Honeypot on public cloud

Access Controller: There needs to be a mechanism to make sure re-attack by hackers are controlled when a software honeypot is already compromised. Access Controller does the job, which will stop more attacks in to the IoT devices. Setting up proper policies is a big challenge here since too stringent policies will block benevolent requests, whereas liberal policies could create high risk too. Hence, a data driven approach was used here to dynamically understand which packets are more malicious prone and which are not. A tool called snort is used in the implementation that runs in host VM instance to monitor in and out going packets traffic. Asymmetry in the data flow is noted as the potential attacks blocked by public clouds.

Shell Interceptor: Traffic flows in through SSH session. Firstly, the data packet gets decrypted to plain text. Then the packet processor will find out the type of packets (CHANNEL_DATA, CHANNEL_WINDOW_ADJUST). Plain text data and the resize event data are sent to Inference Terminal for analysis.

Fidelity Maintainer: Its job is to implement the strategies to protect from attackers in recognizing the honeypots. QEMU - Quick emulator is an open source virtualizer and emulator. It has been used in the software honeypots to enrich the reliability. The configuration is customized so as to resemble its emulated IoT device. Though QEMU provides many CPU profiles, the best match with the CPU metrics of emulated device is selected. No need of emulating disks, since initramfs is used to achieve in-memory filesystem. Moreover, most of the IoT devices do not have disks. Sensitive information is masked in the IoT devices by forging /proc/cpuinfo in OpenWrt, since hackers have the tendency to check through whether the IoT thing is emulated or not by checking /proc. Also, the IP addresses of VM's in the eight public clouds were periodically changed by opting elastic IP feature in the cloud services. This is done because IP ranges of public cloud can usually be obtained fully or partially, it could be possible for the hackers to understand the system based on IP addresses.

Inference Terminal: Even though the shell interceptor acquires plain text data, there may be escape sequences and control characters to be handled in the data feed. An open source tool called pyte is used, which is a terminal emulator. Pyte is customised to get the history of feeds instead of screen by screen. Special control chars and escape sequences were converted to plain text, so that we will know the actual text converted.

Reset Manager: It's a heartbeat based reset mechanism used in the system, whenever the back end controller receives three consecutive missed heartbeats, it will send out the reset command to QEMU. Malwares like Mirai can attack and kill the Telnet process service, hence the reset manager regularly tries to connect QEMU via SSH/Telnet. Whenever it notices the connection fail, then it will reset the QEMU also.

III. RESULTS AND ANALYSIS

This portion details the result analysis of the data recorded for the duration of one year i.e from 01/Oct/2018 to 01/Oct/2019. The overall statistics and the working flows of the captured attacks are introduced first, then the detailed study of the attacks and suggestions for defence strategies against IoT attacks are explained below

A. Generic Topographies, Statistics and Analysis

It's been generally classified that there are two types of IoT attacks are seen, Malware types of attacks and file-less attacks. Malware type downloads malware file from internet, whereas file-less doesn't download any from internet. Work flow of the attacks captured in this implementation is portrayed in the figure 6 below. There are three stages in it; Intrusion, Infection and Monetisation.

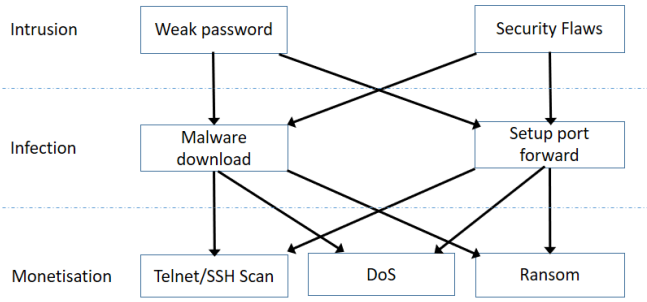


Fig. 6. Work flow of the attacks captured.

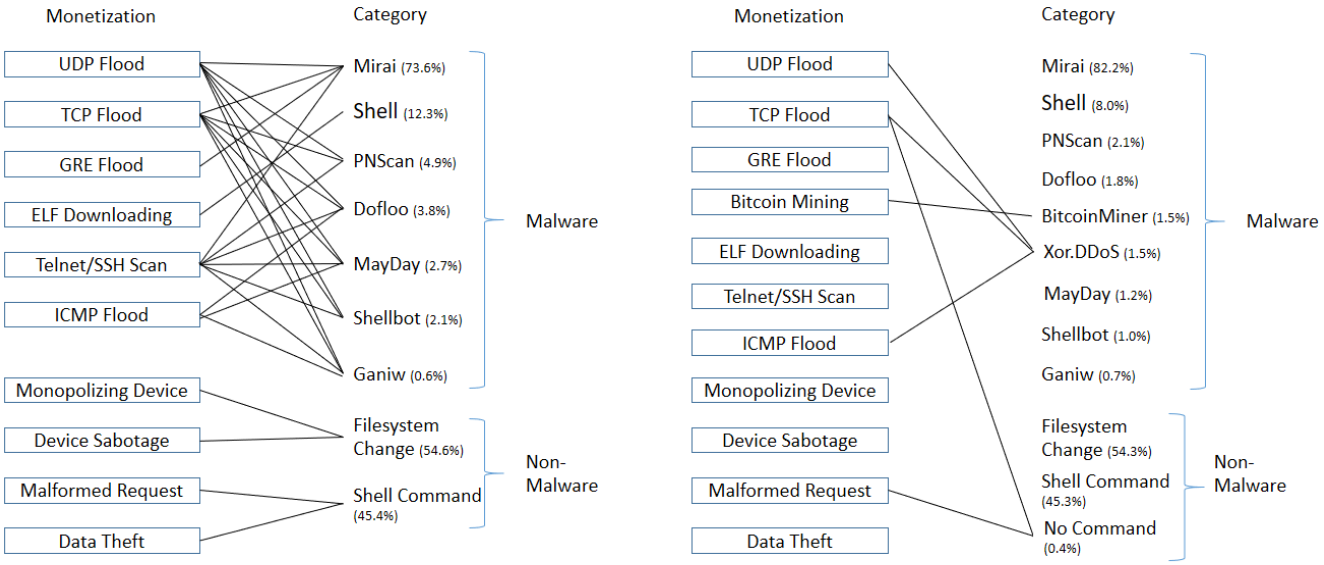
Intrusion: In this phase, the most common method of intrusion observed is brute force, where the attacker tries to login the IoT device with common username and passwords. It's also found that majority of the IoT devices use default credentials that's been set from the factory.

Infection: After the attacker gets in to the IoT device, they use wget or tftp to download the malwares. Usually malwares connect to the C & C (command & control) server for the commands. Once it gets the commands, it can perform various types of attacks like Telnet/SSH, Dos etc. In the file-less attacks, nothing is downloaded, therefore traditional methods of fingerprinting cannot be applied here. Certain system files are modified and/or port forwarding is setup in place of this.

In comparison with the volume of attacks between software and hardware honeypots, it has been observed that the software honeypots had 34% fewer suspicious attacks. This is because of two things; clouds prevent certain kinds of attacks (by means of attack filtering) before it reaches the honeypot. Secondly, the attackers would have tried some advanced techniques to gather the VM identity, though many steps were done on newly developed HoneyNetCloud to mitigate exposing the VM identity. Nevertheless, the impact of this on the study is least affected since all types of attacks on hardware honeypots were also captured on software cloud honeypots too.

Malware based attacks captured in hardware honeypots can be classified in to seven categories as illustrated in the figure 7a. There were 427 various types of malwares downloaded by the hackers. Whereas software honeypots have collected 595 types of malwares and they can be classified in to the nine categories as depicted in figure 7b. Among these, the Mirai takes the major portion – 73.5% and 81.2% for hardware and software honeypots respectively. Attacks were targeting multiple architectures of the IoT devices.

In addition to the malware based attacks, HoneyNetCloud has captured eight various types of file-less attacks too. The classification is based on the behaviours and intentions of the attacks. Figure 7 depicts the categories of the attacks captured and the percentage of them. Table 2 details each type of attack and its analysis.



(a) Attacks captured by hardware IoT honeypots. **(b) Attacks captured by software IoT honeypots.**

Fig. 7. Categories of the attacks captured by hardware and software honeypots in HoneyNetCloud. In figure 7b, only new connection lines relative to those in figure 7a is drawn.

Table 2: Attack types and its behaviours observed

Type I (1.8%)	• Attacker occupy the end system by taking control of the system. • Modify the password of the IoT device using *passwd*. • After modifying the password, the attacker gets full control • Also this will make sure that no one else log in to the device other than attacker.
Type II (54.6%)	• In this type, attacker damages the systems data. • Done by deleting or modifying certain config files and programs using *rm* and *dd* • Scenario is removing the watchdog daemon, once removed watchdog device will not function (*/dev/watchdog*). • Since the device will not reboot when it malfunctions, attacker gets longer time to utilize the system.
Type III (8.3%)	• Prevention of the auditing or monitoring devices. • E.g is killing the firewall service or watchdog service using *kill/service* • After stopping the services, attacker gets opportunities to exploit the vulnerabilities that are known and can do whatever intended to do.

Type IV (7.2%)	• Gathering system information using *lscpu* to get the hardware details and using *uname, ps* and *netstat* to retrieve system information. • Details gathered would be useful for attackers to carry out specific purposeful attacks. • For e.g downloading and running system/platform specific malwares.
Type V (23.7%)	• Stealing sensitive and valuable information by reading passwords and configuration files using *cat* command. • Though the sensitive details stored (in */etc/shadow*) are hashed, however attackers could analyse the behaviours to recover the passwords using tools available in market.
Type VI (0.4%)	• This type is basically launching network specific attacks. • Done by sending HTTP requests to exploit vulnerabilities of the target systems to launch DDoS attacks • Other most common attacks are SQL injections and OpenSSL Heartbleed
Type VII (3.4%)	• It is unclear that some commands are used for unknown reasons. • Commands including *lastlog, help, sleep, who* etc were used which is presumed to be used to collect and analyse other system resources.
Type VIII (0.4%)	• In this type, no shell command is used to attack. • SSH Tunnelling is an example where hacker has compromised a device. • When hacker gets the credentials of SSH service by bruteforcing, another device in the network is converted to proxy server by configuring the SSH port forwarding on the compromised honeypot. • In order to prevent other machines connecting to the service, attacker binds the SSH port forwarding service to a loopback address and then the attacker launches attack from the compromised IoT device. • The real IP address of the attacker is hidden and hence it is really difficult to track the attacks. • This type is quite challenging due to the anonymity and also that shell commands are not used.

B. Key Understandings

With the detailed study done on IoT attacks analysis using honeypots, I could gather many insights.

a) Previous study on IoT attacks were very limited and it was focused mainly on malwares.
b) Number of file-less attacks is 10% more than that of malware attacks in IoT devices.
c) Study reveals that the weak authentication is the widest reason of the attacks.
d) Habit of users not changing the default username and password worsens the criticality. This paves way to the hackers to remotely get in to the network for malicious activities.
e) Figure 8 lists the top shell commands used by attackers and its frequencies of usage.

Fig. 8. Shell commands used and its frequencies in percentage

f) It's also observed that some of the shell commands are obsolete in IoT device. So, the manufactures can disable them so the surface of attack can be reduced to minimum.
g) Majority of the file-less attacks (41.2%) are collecting the system information or de-vaccination operations, (i.e shutting down the firewall or killing the watchdog service etc) so that more attacks can be done effectively.

Even though the attack vectors studied in this paper are applicable to general purpose systems (i.e PC/Servers), effects of attacks are entirely different. Since the PCs and Servers are highly protected by strict policies imposed by organizations, enhanced credentials etc, they are less attacked.

C. Defence Strategies and Challenges

Firstly, the portion of non-malware attacks that modified the filesystem of the attacked IoT devices was 54.3% (Type I – II). In order to resist such attacks, manufacturers should have a policy set to use a non-root default system user. Secondly, 99.6% of Type I-VII file-less attacks were using shell commands. Such attacks were detected by auditing the shell command trail of IoT devices. Generally, IoT device uses read only filesystem like SquashFS which reduces the malware based attacks, however it eventually increases the risk of detecting the file-less attacks. But, if a hybrid filesystem is used where certain parts of the filesystem is writable like OverlayFS to enable logging, it will enable malwares to download and initiate attacks. Hence this is a tricky situation and is a challenge among the IoT device manufacturers. If the IoT device is not capable in using a unique strong password, then such device should not be used connecting to the public internet. Instead VPN can be used if users wish to use such device to connect remotely. This defence strategy would be helpful to harden the security of IoT devices and would assist the manufacturers and administrators to validate effectively.

IV. CONCLUSION

Attacks in general has increased abnormally due to the many facts including the advancement of technology and the increase of users and smart devices connected to the internet globally. IoT attacks has become drastically dangerous since the surface to attack increased. Past researches were mainly focused on malware based attacks, however in this paper, detailed attack analysis of file-less, malwares and other category of attacks were studied. Malwares like Mirai can quickly spread among IoT devices, however they can be restricted by fingerprinting. However, file-less creates a major threat in IoT networks, since they are silent intruders and is difficult in noticing them. To understand the patterns of attacks, a new infrastructure environment implementation is setup named HoneyNetCloud with four hardware honeypots and hundred numbers of cloud based software honeypots deployed in multiple public clouds. The attacks were captured widely for a period of one year up to Oct 2019 and the data collected were analysed in detail. A good range of various types of attacks were able to be captured with different profiles, characteristics, behaviours and influences. Actionable defence strategies could be derived from the study. It is time to think out of the box from the traditional attacks mainly lurking on malware based. We need to take the file-less types of attacks too in to consideration going forward and a universal defence framework for IoT attacks is terribly needed at this point in time.

REFERENCES

1. Dang Fan, Zhenhua Lii,Yunhao Lui, Jingyu yang, “Understanding File-less Attacks on Linux-based IoT Devices with HoneyCloud” 17th International Conference MobiSys (2019)
2. Zimu Zhou, ChenshuWu, Zheng Yang, and Yunhao Liu. 2015. Sensorless Sensing with WiFi. Tsinghua Science and Technology 20, 1 (Feb. 2015), 1–6.
3. New Trends in the World of IoT Threats. https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/
4. Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, et al. 2017. Understanding the Mirai Botnet. In Proceedings of USENIX Security.Vancouver, BC, Canada.
5. Ekta Gandotra, Divya Bansal, and Sanjeev Sofat. 2014. Malware Analysis and Classification: A Survey. Journal of Information Security 05 (Jan. 2014), 56–64.
6. MMD-0062-2017 - Credential Harvesting by SSH Direct TCP Forward Attack via IoT Botnet. http://blog.malwaremustdie.org/2017/02/mmd-0062-2017-ssh-direct-tcp-forward-attack.html
7. McAfee Labs: Cybercriminal Tactics Shifting From External Malware Threats to ‘file-less’ Attacks. https://www.dqindia.com/mcafee-labs-cybercriminal-tacticsshifting-external-malware-threats-file-less-attacks/
8. Now You See Me: Exposing File-less Malware – Microsoft Secure. https://cloudblogs.microsoft.com/microsoftsecure/2018/01/24/now-you see-me-exposing-file-less-malware/
9. Tips for Guarding Against Untraceable, “File-less” Cyberattacks. http://www.govtech.com/security/Tips-for-Guarding-Against-Untraceable-File-less-Cyberattacks.html
10. Lance Spitzner. 2003. Honeypots: Tracking Hackers. Vol. 1. Addison-Wesley Reading
11. dd-wrt support documentation https://dd-wrt.com/support/documentation/
12. Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein,et al. 2017. Understanding the Mirai Botnet. In Proceedings of USENIX Security. Vancouver, BC, Canada.
13. Lance Spitzner. 2003. Honeypots: Tracking Hackers. Vol. 1. Addison-Wesley Reading.
14. ISO/IEC 20922:2016 Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1. http://www.iso.org
15. ISO/IEC 20922:2016 Information technology – Message Queuing Telemetry Transport (MQTT) v3.1.1. http://www.iso.org
16. Mallikarjunan. E “Security Operations and Automation in IoT networks” International Cyber Security Conference Hyderabad (2019)
17. Rohin I.E Pujari Prema T “IoT Based Smart Secured Home Systems” 2nd International Conference on Intelligent Data Communication Technologies And Internet Of Things (ICICI 2019)
18. Sundreshan Peramal, Norita Md Norwami and Valiappan Raman. Internet of Things(IoT) digital forensic investigation model: Top-down forensic approach methodology. In proceedings of International Conference on Digital Information Processing and Communications, Oct 2015
19. Bogdan Copos, Karl Levitt, Matt Bishop and Jeff Rowe. Is Anybody Home? Inferring Activity from Smart Home Network Traffic. In proceedings of the conference 2016 IEEE Security and Privacy Workshops (SPW) May 2016

AUTHORS PROFILE

A.R Jayakrishnan received his masters degree MCA from Mahatma Gandhi University in 2001, PG Diploma in Cyber Security and Forensics from IFS in 2014 and currently pursuing PhD from Bharathiar University. He started his career as a Lecturer in Computer Science with University of Calicut. He has been working in the IT industry since 2003 as a Technical Architect and Consultant. His research interests are Cyber security, Forensics, IoT attacks, Intrusion detection and prevention. He has published papers in Springer series, CSI publications and other international journals. He is a life member of Computer Society of India and IEEE.

Dr (Mrs) V Vasanthi pursued M.Sc (CS), M.Phil and Ph.D in Computer Science from Karpagam University, Coimbatore in 2014. She is currently working as an Assistant Professor in Computer Science with SKACAS, Coimbatore, India. She has published many papers in reputed international journals including Thomsons Rheuters (SCI & Impact factor) and conferences including IEEE and Springer. Her main research focus areas are Adhoc and Sensor Networking.