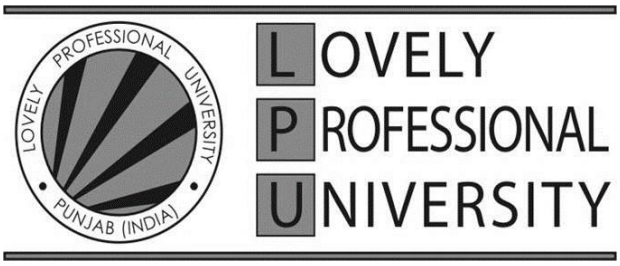

Capturing Attacks on IoT devices by making a Honeypot using Raspberry Pi

A Dissertation Proposal submitted

by

Bhavkanwal Kaur

to

Department of

Computer Science & Engineering

in partial fulfillment of the Requirement for the

Award of the Degree of Master of

Technology in

Computer Science & Engineering

Under the Guidance of

Er. Pushpendra Kumar Pateriya

November, 2017

INDIA'S LARGEST UNIVERSITY

LOVELY PROFESSIONAL UNIVERSITY

Transforming Education, Transforming India

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering

Program : P172::M.Tech. (Computer Science and Engineering) [Full Time]

COURSE CODE : CSE548 CSERGD0330 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :**

Supervisor Name : Pushpendra Kumar Pateriya **UID :** 14623 **Designation :** Assistant Professor

Qualification : _____ **Research Experience :** _____

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Bhavkanwal Kaur	11604925	2016	K1637	7589420414

SPECIALIZATION AREA :Networking and Security **Supervisor Signature:** _____

PROPOSED TOPIC : A Harmonic Security Scheme for Resource constrained Internet of things platforms

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	8.00
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.67
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.33
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	8.00
5	Social Applicability: Project work intends to solve a practical problem.	7.33
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.33

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): Yes
PAC Member 2 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 3 Name: Raj Karan Singh	UID: 14307	Recommended (Y/N): NA
PAC Member 4 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 5 Name: Sawal Tandon	UID: 14770	Recommended (Y/N): NA
PAC Member 6 Name: Aditya Khamparia	UID: 17862	Recommended (Y/N): NA
PAC Member 7 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): NA
DAA Nominee Name: Kuldeep Kumar Kushwaha	UID: 17118	Recommended (Y/N): NA

Final Topic Approved by PAC: **A Harmonic Security Scheme for Resource constrained Internet of things platforms**

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal **Approval Date:** 04 Nov 2017

11/29/2017 11:21:55 AM

Abstract

In the past few years, the use of Internet of Things appliances has increased drastically, so does the security issues related to them. The number of attacks happening on IoT devices is increasing day by day. Most of the malicious attacks happening on IoT devices are carried out by botnets. Now different devices require different security measures. If different types of security options are available, we need to decide which solutions are best for a particular IoT device or application. It is important to understand the requirements for the threat prevention of a particular device. Individual appliances or a whole enterprise need a system which can detect and respond to different threats, malware or hacking of the system. Here we propose a Raspberry Pi based Honeypot which can be used with other attack detection applications to attain more efficient and better cyber- security plan. In our research work, we are going to collect the data from our honeypot and will do device identification after scanning from VirusTotal, in order to better understand the characteristics and the insights of the device. Once we come to know about its characteristics, we can even make the additional device signatures and can do device identification using various scanning technologies.

Acknowledgement

I would like to take this opportunity to express my deep sense of gratitude to all who helped me directly or indirectly during their thesis work.

Firstly, I would like to thank my supervisor, **Mr. Puspendra Kumar Pateriya**, for being a great mentor and for constantly supporting and guiding me in the successful completion of this dissertation. The confidence shown on me by him was the biggest source of inspiration for me. It has been a privilege working with him.

I also wish to express my sincere thanks to all the faculty members of computer science and engineering department for their support and encouragement.

I would like to express my sincere appreciation and gratitude towards my friends for their encouragement, consistent support and invaluable suggestions at this time I needed the most.

At last, I would like to thank my family for their love, support and prayers.

-Bhavkanwal Kaur

Declaration

I hereby declare that the dissertation proposal entitled, "Capturing Attacks on IoT devices by making a Honeypot using Raspberry Pi", submitted for the Master of Technology degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or Diploma.

November, 2017

Bhavkanwal Kaur

Reg No.11604925

Contents

List of Figures

Chapter 1
Introduction

Internet of Things facilitates many gadgets that are meant for everyday use to communicate with each other by means of Internet. These gadgets are considered as smart gadgets since they are able to deliver the message or data to a streamlined system whose work is to supervise the received information and take measures on the basis of assignment given to it. For the future novelties, IoT is like a stimulant since the growth in this field is drastically increasing by time.(19) Internet of Things has its applications in various areas of expertise like Wearable devices, Management of Traffic, Power houses, Smart homes and cities, Information Sciences, Behavioral Sciences.(20)

Since various computing devices embedded with IoT communicate with each other using Internet which has large amount of data con-corded with it, security is a major concern over there. For an intruder, targeting the IoT device is very easy task. The intruder usually attacks the network layer and once it is endangered, the attacker can now easily access the device and also he can hack various nearby devices as well. Providing security in an IoT infrastructure is a very tedious task since in an IoT network we don't only have traditional appliances like laptops and computers but in its network we also have real world appliances like refrigerator, cars, door locks, television, washing machine and many more. Now these appliances do not have any safeguard against various viruses and malware. So, these real world devices are highly vulnerable to be used by the attacker as a "Internet bot" or a "Web Spider" to spread the malignant code to corrupt other devices. According to the analysis made by International Data Corporation, more than twenty million real world devices will be connected by Internet. But eventually with such rapid growth in IoT, the opportunity for attackers and hackers also grow to a very large scale (21). The scope of performing attacks like "denial of service", spoofed emails and or spreading any other harmful worms or viruses has increased drastically.

IoT devices are considered as the most compromised devices in terms security. Recently a test was conducted by Hewlett Packard Enterprise on various IoT devices, in which it was found that more than ninety percent of the devices possessed at least a fraction of secret data through the migratory application or cloud or by the device itself. Now the attacker can perform cyber attack on this secret information or can get unauthorized access to it and the private information can get compromised. So this will cause reduction in security in all aspects of it, which are integrity, confidentiality and authenticity. Hence the end users are very cautious to adopt this new technology. In order to make this new technology reliable, these IoT appliances need very big improvement in terms of security and privacy. Privacy here refers to the fact that the data is very personal to the user and no one else can access it except the user. Security refers to the fact the personal information is secure against any sort of unauthorized access. Along with the security, we need to ensure that data transmission between the various IoT devices should be efficient such that there should not be any loss or modification in the data. If there is loss in the data or say wrong data is exchanged, this can lead to many deliberate consequences if the data or information is very crucial.

1.1 IoT Layers and Security Problems Related to them

The IoT model consists of the following layers as shown in the figure 1. These layers are actually responsible for establishment of an IoT.

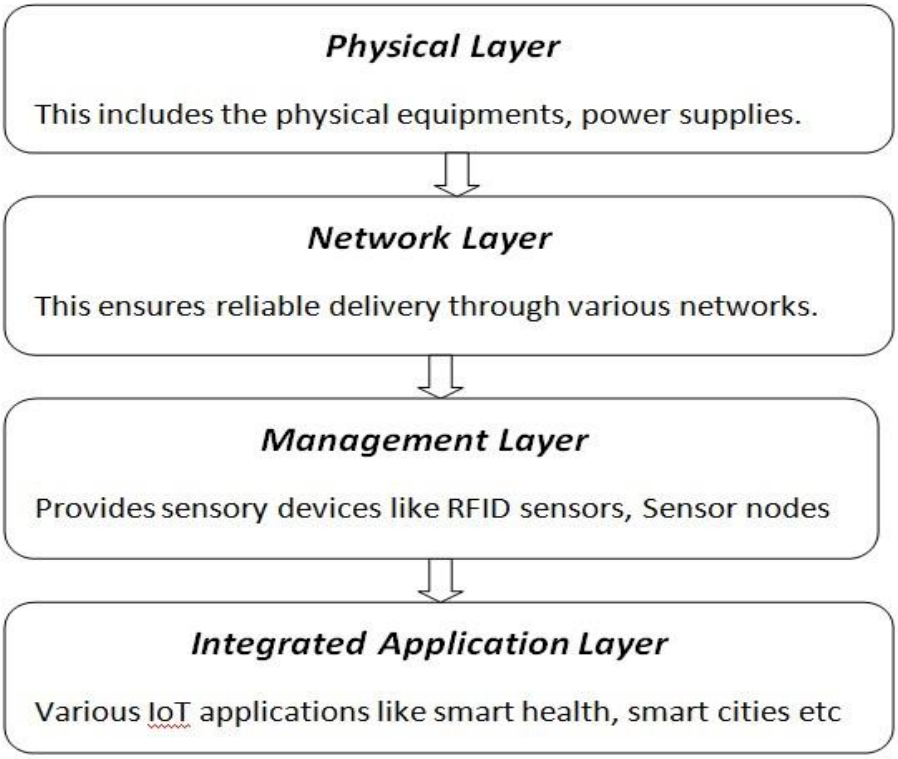

Figure. 1

1. Physical Layer: This layer consists of the hardware components like power houses, smart equipments and many other physical devices. The networking among the various smart gadgets happens when they have a strong foundation of these hardware components as their backbone.

   In IoT, providing security is major concern in constraint resource availability. At the physical layer, many security problems are being faced. With the rapid growth in the technology every day, the need for enhancing the security of the power generators and many hardware security machineries is increasing drastically. The gadgets should be protective such that they must be able to face any sort of physical invasion. The devices need to have a long battery lifetime so that if they is any power cut or blackout, they must be efficient enough to work better on their battery power.(22)(23)(24). Now the security at the physical layer can be breached by doing any physical harm to sensors or nodes, by any entity which acts as an intruder. The entity may have the potential to destroy the equipments physically and the equipments may lose their working abilities and this can

cause havoc to an IoT system. Another thing can happen that the devices may get damaged by natural calamities like rain, snow, floods. The devices need to robust enough to face these calamities. Now, if the device runs out of power, it can cause the whole network to get down because every node or sensor is connected to one another. So if anyone goes out of power, this will lead to denial of service attack themselves. To prevent this, the devices should enter power saving mode when not in use in order to save their power. On the other hand, the malicious node can send continuous wrong requests to the sensors or other intermediate nodes, which hinders them to go into the sleeping mode. So this can also called as sleep starvation attack.

The hardware structure is the actual reason for the existence of any device. If there is any problem in the hardware, such that due to which the device may not work properly or it leads to the failure of the device as a result it may send any wrong data or may not be able to send any data at all. For instance if any malicious party performs cyber attack on the smart home and is able to breach the security of home which can lead to theft or much more worse consequences. Also, in the big organizations, there are many automated systems planted within it. For example if the intruder is able to disrupt the functionality of the swipe card attendance systems of all the organization by attacking the main operating system at the back end, this can cause real work loss to the company.

2. Network/Internet Layer : The composition of this layer has both hardware as well as software elements for instance networking devices used for communication, intermediate nodes, sensors, end servers, regional anatomies and many more. This layer is responsible for reliable transmission of data between intermediate nodes, among different networks and even between a network and an end user.

   The network layer is responsible for data transmission, so the attackers always have their eye on it. Hence it is very much prone to attacks. Due to large volumes of data that it carries, it is very much susceptible to congestion. The major concern over here is that the data should not lose its integrity, which means the sent data should not be modified and

the data needs to be authenticated, which refers to the fact that it is coming from the reliable sender. If there are compromised nodes present in the network then it can cause havoc to the security of the network.

The attacker can continuously send malicious data causing the system to respond recklessly. This technique is usually followed to perform a physical attack by masquerading with such threats. The attacker can also attack the storage devices which contains data, which may or may not be confidential, in very large amounts. The attacker can either perform a passive attack by only looking at the data and using the information later for the attack or it can perform an active attack by disrupting the integrity of the data. If the same malicious data is sent again and again by the attacker to the different users may enlarge the area for disruption by the attacker.

At the network layer, the attacker can also prevent the a number of client to access various services by performing denial of service attack. The attacker while performing DoS attack can be sending many number of malicious requests to the server such that the server do not get free enough to respond to the actual users, who actually need the services. The attacker actually does the closure of exchange of information between the clients and the server. For instance, the attacker can capture the database of any health institution and if the services are executed at a low frequency network of IoT, this may lead to critical situations which can be life intimidating and can cause huge collapse in the business. (25)

Another thing is, the owners of devices may have a misconception that if the devices are physically under their eye then they are safe and they are only prone to attack if they are physically accessed by the intruder. However this is not true. There are devices which are usually planted at the places which are rarely attended or the places where human go very rare. So, in that case, physical eye on devices is not possible but they do require security, for example pace-maker. Pacemaker usually controls irregular heart rhythms. This is implanted in the chest. It makes the heart to beat at normal rate using electrical signals. It has a battery, a computerized generator and wires at the end of which sensors

are connected. Usually sensors keep a check on the beat rate of the heart. If the heartbeat is not normal, the sensors send a data to the computer in the generator through the wires and then the generator send electrical pulses to the heart and the heart comes to normal mode. Pacemakers are little bit unsafe for the users since they require unvarying time to send control signals after definite equal intervals of time. These are designed such that they can connect with other devices for sending and receiving information. Some unauthenticated devices may pretend them as authenticated ones and connect to these devices and hence get the access to them without having authorization and the device hence get compromised. The can cause great danger to the heath of the patient.

Some attackers make their prey gateway between the Internet support and the sensor nodes. The attacker can either perform routing attack or denial of service at the gateway which can either stop the communication between the client and the server or can send malicious data to the client from the frequency which is meant for providing Internet facility. This attack can then substantially cause loss to the sub-domains like smart cities or VANETS. (26)

3. Perception Layer: This layer is comprised of numerous forms of sensing mechanisms for instance sensors for capturing temperature change, sensors for detecting air pressure, RFID sensors which are used for sensing different devices.

The main risks in the management layer occur at the host level. Here the hosts are sensors. The main purpose of the attacker here is to hack the sensor by replacing the software of the sensor with their own software. The majority of the attacks happening at this layer is done by the foreign entities. The attack is mostly done on sensors and other information collecting parties. (22)(23)(24)

Now, when the information is delivered through a wireless medium, over wide areas, it is possible that the information may contain noise. The noise can cause any of the data packets to get lost or the data can also get changed or modified due to the change

of bits due to noise. Now, if the information is very crucial, any modification or change in data can cause great loss.

In order to gather the data being exchanged between different nodes, malicious nodes settle down near the reliable nodes of the network. Now the IoT devices are capable of capturing human information without their consent. This means they are able to analyze or determine the person and are able to store the information related to them in a profile. The attacking nodes can settle down near such nodes and try to get their access by connecting with them maliciously, hence gathering information from them. Now, humans can interact either with each other or with an IoT device. The reliable IoT devices are able to sense physical trails of the information being exchanged but in very small amounts.

In a wireless network, the devices are not under any supervision. So they are more prone to attacks such as eavesdropping because the devices here, only communicate through wireless connection using Internet. For instance, if the sensors that are being hacked by the attacker can send push message to the user and can gather personal information from them.

4. Application Layer: This includes many types of utilizations and servings presented by IoT for instance smart health, wearable devices, smart transportation, traffic management, smart cities and many more.

Security problems at the application layer can make the applications to get completely disrupted or some of its features may stop responding or working, hence making the application to get compromised very badly. The worst thing can happen, if its functionality which is providing authentication is corrupted, it may cause the application to provide privileges to an unauthenticated party which could be there with the motive of performing a severe attack. The application can become the victim of malfunctioning if the intruder is able to cause errors in the programming code of the application. For the

appliances which are categorized as application layer items, such attack is a huge threat for them.

Attacks at application layer, can even lead to economic losses. Consider a scenario in which attacker has planned a burglary in a house. The house is using a smart meter here, whose main function is to deliver the information about the data consumption to the utility operator dynamically, which uses this data for billing. This is considered as a secure process unless someone is able to get the access to the data. If an attacker gets the access to the transmission of the data, he will be able to get the patterns about when there are people at home and at what time the home is mostly empty, by checking the patterns of power consumption. If the attack on the smart grid becomes successful, then it can lead to huge economic loss.

Also, in areas of atomic power plants, if there is a mobile node carrying many software viruses and the softwares are not updated in time, it can lead to many disastrous results.(27)(28).The attackers can also do tampering with many other node based applications. The attackers mostly exploit the application by attacking the nodes of the device and they either disrupt their functioning or they replace them by their own malicious nodes. The security of the device should be such that, it doesn't allow the attacker to do any temperament. Providing protection to only few parts of the device may not be sufficient. Now, even if we provide secure protection to the nodes, the attacker now instead of attacking the nodes, will attack on the local environment of the device by manipulating it such that it can lead to the malfunctioning of the device. For instance, the attacker can fix the temperature of the environment in which a temperature detection sensor is deployed. As a result the sensor will give the wrong data since the temperature of the environment is manipulated now. Another instance could be say that the smart camera deployed in a home gives the output of out-of-date photos instead of the real ones.

Now, if the attacker wants to attack not only one device but a network of devices, he can spread "worms" among the devices using the Internet. Since the IoT devices are Internet enabled devices, it is easier for the attacker to attack many devices at one time

using Internet. The attacker can attack security devices like cameras, routers, sensors. Also the devices to be attacked with a particular type of worm should be using the operating system for which the worm is designed to attack. For example the worm can attack only Linux based operating system, so all the attacked devices should have the same operating system. Such attacks are really dangerous because they can take the whole control of the system. For example, if the attacker is able to hack the Car's Wi-Fi, he can take control of the car like even of the steering wheel and can make the car to do accidents resulting in even loss of life.

TABLE I
SECURITY CONCERNS OF EACH IOT LAYER

IoT Layer	**Security Threats related to it**
Physical Layer	Failure of the Device itself, Drainage of the power of the device, Damaging the gadgets physically, Environmental Calamities like floods, storm
Network/Internet Layer	Attack is possible both on cloud carrying user's crucial data or on the storage device, Negligence of owner in providing security to the devices, Attacks on the gateway between the networks, Intruder can send wrong data to the system, DoS attacks
Perception Layer	Data may containing wrong in-formation leads to loss of data, placing malicious nodes near to netwrok nodes for data sniffing, Masquerading attacks
Application Layer	Non-updation of security bugs in the software of the device, Changing the IoT device environment in order to make it receive wrong data

1.2 Different Security Providing Protocols for IoT

We need standard protocols for making a secure connection among different IoT devices. Various standard protocols are used for the making of an IoT device as well. For the compatibility of smart devices, Internet Protocol (IP) is used as a standard, which is supported by Internet Engineering Task Force (IETF), an International Organization. Since IPv4 addresses are coming to a finish line, IPv6 is the new introduced solution for providing communication facilities among various smart objects. (29) The Internet protocols being used by the IoT devices are usually same as that used by the classical Internet devices so that it becomes

possible to build the ”Extended-Internet”, which is the conglomeration of the IoT with the Internet. The integration of the IP protocols with the smart devices can happen only if the architecture of the smart devices is able to support the standard IP architecture. The IoT devices should be portable such that they must be able to adopt to the already defined security algorithms. (30)

IPSec (Internet Protocol Security) provides secure exchange of data at the network layer among the different IoT nodes.(31) IPSec, used in IPv4, was actually developed for the IPv6. IPSec is impacted in IPv6. IPSec can provide security during the data exchange among the different hosts or among a host and a network or among different networks. IPSec can provide confidentiality, authenticity and integrity for each and every IP packet. Authentication Header (AH) and Encapsulated Security Payload (ESP) are the protocols that provide these security requirements. ESP provides integrity, authenticity and confidentiality while AH provides authentication about the sender and protection against replay attacks.

At the Transport Layer, the security is provided by the Transport Layer Security Protocol or the Datagram Layer Security Protocol. This protocol provides confidentiality utilizing symmetric key encryption, protection against replay attacks using message authentication code and peer to system authentication by applying asymmetric cryptography. A three way handshake is done at the beginning for providing peer-system authentication. Here for end to end reliability, it is dependent on the reliability of the intermediate nodes. This means that for end to end secure transmission, security at the intermediate nodes is very necessary. This is the major issue in the Transport Layer and IPSec advances.

An alternative to this problem is providing security for the reliable transmission at the application level. This in turn reduces the overall consumption of resources at each node for flow and error control, data processing, thereby reducing the cost as well. Also, the encryption of data done at the application level makes the security implementation much easier. But the limitations are also there breaking the

security at the application level is not a tedious task for the attacker and also the code for building the application becomes quite complex. Implementing the security protocols at the application level is complicated during the development of the application.

1.3 Different Security Schemes for IoT Protection and the Limitations Associated With Them

Since security is the major concern in IoT because we are not able to provide enough security to the IoT devices due to lack of resources. There are many schemes being introduced to provide security to the IoT devices, some of which are discussed here.

Renu Aggarwal introduces a security scheme in relation to the " Internet of Things ". The scheme here shows an improvement in the security provided by RIFD systems. The efficiency of a low-cost RIFD authentication protocol is observed and found that this protocol has a limitation that does not provide any security against disclosure attacks and desynchronization attacks. A new scheme is introduced in this paper that overcomes the limitations of this protocol to some extent. But this still is prone to attacks because RIFD tags are easily hacked by the professional hackers, this is the reason "RIFD hacking" is increasing these days. (32)

Lui et al., introduces an approach to protect the system against eavesdropping, man-in-the middle attack, repaly attacks by mending the weak spots in maintaining the integrity of the data and providing security to the device.In this it proposes a solution such that whenever a user wants to connect to an IoT device, it has to get an approval from a "Registration Authority" to access that device. If the registration authority approves the user, the user is now considered as authenticated one and is allowed to make a connection. (33)

A.Dohr introduces a new progressive anatomy which contributes to the life of elderly people by making them live a secure and independent life. This new anatomy is

“Ambient Assisted Living (AAl)”. This new scheme makes old age people to live a comfortable life at home with smart objects. But the main drawback of this scheme is it does not have privacy and security features, making it prone to attacks. (34)

A.Sardana and S.Horrow introduces a scheme to provide authentication of the information exchanged between cloud and the devices that are connecting with it. This approach is efficient but the main problem is the protocols required to practically implement it has not been formulated yet. (35)

A preference based security protection framework was introduced by Tao and Peiran, in which a third party estimates the security need and amount of privacy required by the operator and provides security to the user according to the need only. This approach does provide a efficient mechanism to use the resources in a very sensible manner but still the security techniques introduced in it require more advancement in order to make a strongly secure IoT network. (36)

You-guo and Ming-fu did an improvement in enhancing the security of the exchange of data among the two parties via Middleware techniques. Middleware is the new scheme which uses various cryptographic techniques for data privacy and security for instance authentication, data integrity, digital signatures, user identification, communication is happening between reliable devices. But Middleware is a newly introduced scheme and a lot of work need to be done in this area. (37)

All the above schemes are better at preventing the attacks. But what if we had a much better security scheme that can prevent the attacks from occurring so that we don’t have to detect them. Honeypots are the instance of such systems.
In the last few years the majority of the attacks happening on the IoT devices are DDoS attacks which are mounted after infecting the devices and then used for spreading the infection further. The infected devices are used to attack the other devices later.

The most extrusive of such attacks was marai, which is a malware that attack the devices with weak login credentials and then use these infected devices to spread the infection further. There are a lots of such malware which are used for attacking and infecting such systems. But the major limitation is that these are only detected after a large scale attack. In order to capture such attacks at the early stage honeypots are used.

1.4 Honeypots

Honeypot is basically a system designed to attract and capture the attacker. It attracts the attacker by making itself vulnerable to attacks. It attracts the attacker by showing him that it contains information which of use of the attacker. Honeypots do this in order to make log information about all the activities of the attacker which can be used to make patterns and prevent further such attacks.

1.4.1Classification of honeypots

1.4.1.1 Based on interaction

High interaction honeypots (1) provide full interaction of the attacker with the complete system. In order to collect the information regarding the peak levels of vulnerability possible a particular system. **Low interaction honeypots** provide only a limited set of functionalities to the attacker in order to collect a log of attack patterns. Such honeypots are basically used to find the source from where the attack is happening rather than finding the methods being used in these attacks. **Medium interaction honeypots** lay in between high and low interaction honeypots.

1.4.1.2 Based on deployment

Research honeypots are used for academic purposes by the researchers. They are used to collect the malicious activities of the attacker. **Production honeypots** are set up by the production network along with other servers for the use of corporations.

These are in the category of the low interaction honeypots. Research honeypots are more interactive than the production honeypots.

1.4.2 Honeynets

Network of honeypots is called a honeynet (2). It is a highly controlled environment used to collect and analyze data from the attackers. It consists of the following modules:

1. **Data Control**: Data control ensures that even if the honeynet is compromised, it should not be used by the attackers to perform further attacks. In order to take care of this flow of the data in and out of the system is regulated.

2. **Data Capture**: This is the technique of the activities of the attacker who has accessed the honeynet. This data is then stored for further analysis.

3. **Data Analysis**: This analyzes the collected data and then uses the information to draw patterns or different conclusions. This is then used to do modification in the existing honeynet.

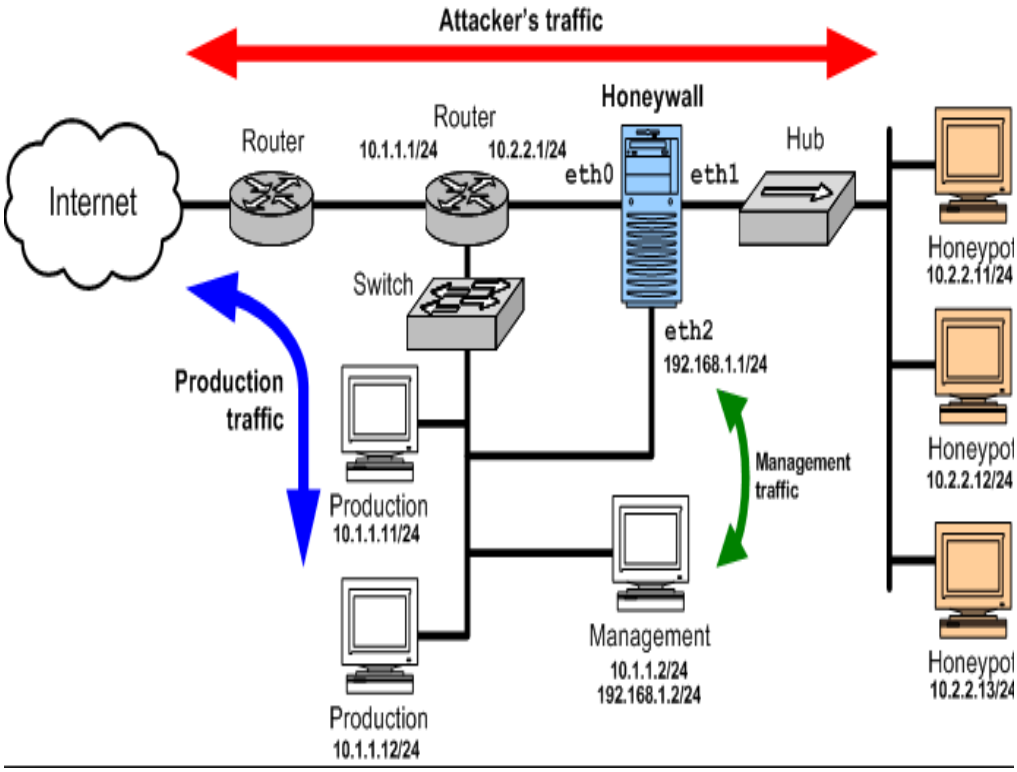

Figure 2. Honeynet

Chapter 2

Previous Work

The concept of honeypot is a very old concept. It was first introduced in 1990. But still it has not gained that much popularity. Deception Toolkit (DTK) (3) is one of the earliest honeypots to be public. It has the capability of masquerading different hosts as well as it can simulate wide categories of UNIX vulnerabilities. These two characteristics make it being a honeynet. It uses TCP wrappers for processing the incoming requests. It is written in Perl. Although it is not so complex, but it not even a high interaction honeypot so get compromised easily.

After Deception Toolkit, Cyber Cop (4) Sting was introduced. It was the first commercial honeypot to be released. It was a Windows Honeypot. It usually simulates the Windows machine, Solaris and the sub network of routers. This was also able to simulate Telnet. For an intruder, it was like the part of the network.

Honeypots started becoming popular with the establishment of the Honeynet Project in 1999 (5). The main goal of the this project is to create an awareness about the existing threats to the Internet and to provide the necessary tools and techniques to the general public so that they can also be aware of the attacks happening on the internet everyday and how to preserve their credentials from such threats.

2.0.1 Popular Honeypots

Nowadays, many open source honeypots are available. These can be downloaded easily and can be used. Here we give a brief review on some of the most popular honeypots being used these days.

2.0.1.1 Kippo

Kippo is a medium interaction SSH honeypot (6) which was designed to collect the log information of the brute-force attacks happening on the system. It is written in Python language using a twisted framework (7). This is a very popular model and is used by others to create a many other types of honeypots. It has a fake file system and also simulates a shell. It has the capability of adding files to the system. Although it has many useful features, it is very slow and is easily compromised.

2.0.1.2 HonSSH

HonSSH (8) is a high interaction honeypot. Instead of being like a server, it works more like a proxy. It acts as a SSH proxy by lying in between the attacker and a honeypot. It firstly accepts the connections from the attacker and then makes a connection with the honeypot. Any data from the attacker is passes through the honeypot and vice versa. Log information is maintained for every data passing through it.

2.0.1.3 Glastopf

Glastopf (9) is a low interaction honeypot used for capturing attacks on Web applications. It has the capability of imitating various vulnerabilities which can be used by the attacker for performing various attacks. It gathers the data from attack which targets the web applications. It basically sends a reply expected by the attacker when the attacker is trying to access the web service. Vulnerabilities are like HTML injection by emulating POST requests, local file inclusion by providing files from a virtual file system and providing a build-in PHP sandbox for remote file inclusion.

2.0.1.4 Thug

Thug is a client side honeypot (10). It behaves like a client and it seeks out malicious servers instead of waiting for being attacked. It emulates a web browser. It is written in Python.

2.0.1.5 Cowrie

Cowrie is a fork of Kippo (11). It supports both Telnet and SSH. It emulates SFTP and SCP protocols. It supports more of the Linux commands.

2.0.1.6 Dionaea

Dionaea is a python based honeypot (12). It detects the shellcodes using libemu. It offers vulnerabilities to the attacker for capturing the malware. It also supports ipv6 and TLS.

2.1 IoT Honeypots

2.1.1 HoneyThing

HoneyThing (18) was designed as a part of GSoC project. It was created for the Internet of TR-069 things (13). It emulates a router which supports CWMP protocol and which has a web interface. CWMP protocol is used to control IoT machines using an Auto Configuration Server and it has a technical specification of TR-069. It emulates some of the most famous vulnerabilities in IoTs.

2.1.2 Telnet-iot-honeypot

Telnet-iot-honeypot is used for IoT devices to capture Telnet attacks (14). It is mainly used to capture malware of botnets and binaries. It is basically written in Python. Instead of offering a live terminal, it simulates a telnet session. For the further analysis, the binaries are generally uploaded to VirusTotal.

2.1.3 MTPot

MTPot (17)is open source honeypot introduced by Cymmetria Research (15). It is used to detect marai malware. It is a light weight honeypot. It finds out the machines infected by the marai malware and collects samples of marai malware if possible. It emulates Telnet servers and the settings are changed according to the version marai which is trying to infect the particular system.

2.1.4 IoTPot

IoTPoT came into existence with the collaborative hard work of the researchers from the Germany and Japan (16). It possesses a sandbox for Telnet attacks and an IoT honeypot. It emulates the telnet services of many types of devices. It has two parts:

- A Frontend, which act like a low-interaction respondent.
- A Backend, which act like a IOTBOX.

IOTBOX provides a high interaction virtual environment. It supports about eight different architectures of CPU, consisting of MIPS and ARM. The frontend sends the commands of the attacker to the backend by establishing a connection with it and the reply is sent to the attacker from the backend. But the main problem is, it is not an open source right now.

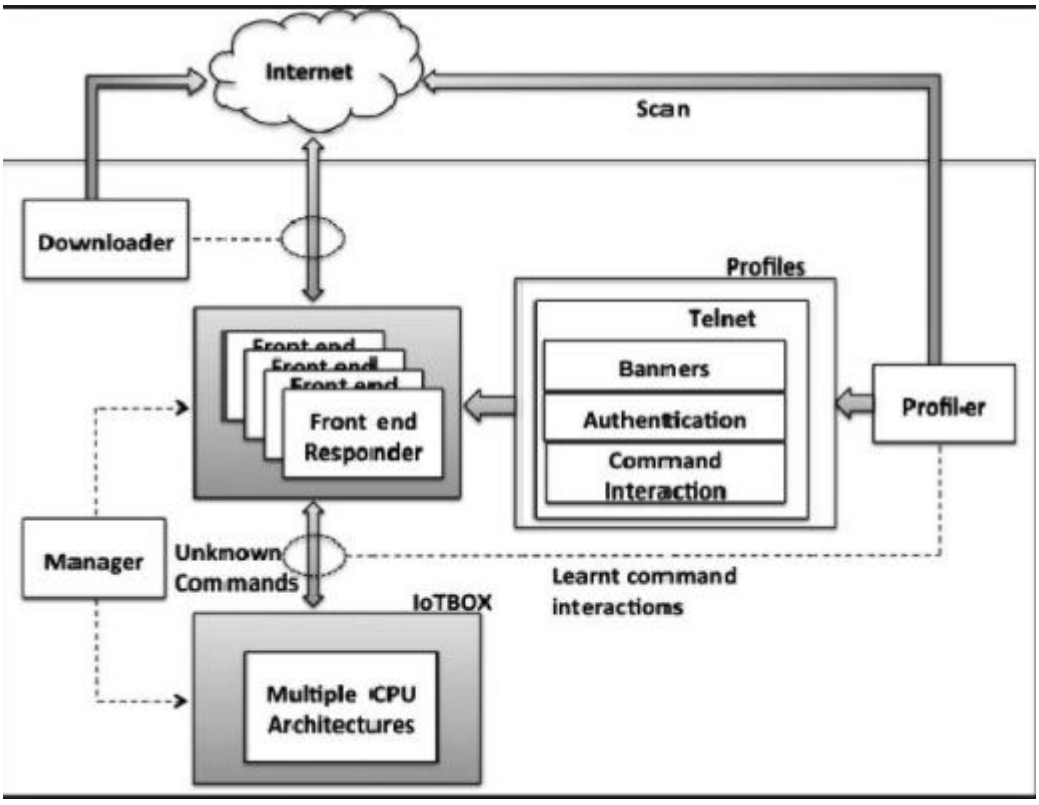

Figure 3. IoTPOT design

Chapter 3

Scope of Study

The aim of the honeypot is to make the intruder believe that he or she is able to access the real network and can access all the information from the network. However the network administrator closely monitors the activity on the honeypot in order to observe the operations performed by the attacker. This in return, enables the administrator to secure the system further from such attacks.

When so many security options are available, the user must be able to decide which solutions are best for their particular system.

Understanding different threat intelligence tools is very important in order to understand that which tool is able to meet the security requirements according to the needs of a particular application.

Whether it is an individual or a enterprise, all need a particular system to detect, analyze and respond to certain type of malware, threat or attack.

Raspberry Pi Honeypot is one of the tools which can be used in many ways for either threat detection or can be used as a combination with other threat detection tools in order to attain a higher level of security.

Once you understand the basic security necessities of different tools, we can access the specific offerings and can go with what is best for it.

Honeypots are used to observe the intruders. By learning about the means of access (the purpose of the attacker and by what particular means, it is able to access the network) and the source of the traffic, we can recognize the attacker and can block them. By looking at the attack patterns can help us to improve our security levels.

How Raspberry Honeypot is different than the other honeypots?

Raspberry Pi is a microcomputer powered by the ARM processor. Making Raspberry Pi, a honeypot device is a very interesting concept. This is because it is relatively very small in size, so it does not accommodate a lot of space. It consumes very small amount of power and also the Pi is very inexpensive. It is designed such that it is able to work with other tools in the Modern Honey Network (MHN is like a centralized server which collects and manages data from different honeypots. It helps in the easy deployment of the sensors and also collects the data which is viewable from a neat web interface.) MHN is the open source framework that allows downloading software code for free.

The Raspberry Pi honeypot can be used along with other honeypots. The information collected by the Pi honeypot about the network can be shared with other honeypots and patterns of the traffic and the source information can be compared in order to detect, say, a particular virus attack. You just need some skill to configure the honeypots but the cost of hosting and labor is very less as compared to that available cost before the open source solutions. If see the benefits the overall cost is negligible.

It is better to detect the attacks at the early stage, before becoming the victim of the attacker. An individual or a company may have lots of crucial information on the servers. The attackers can exploit your network and can modify or steal your personal data for instance medical history, images, financial information etc. If you have very important data or asset but do not have a team for full time threat analysis, consider a MHN like a Raspberry Pi Honeypot.

Another advantage is that the operating system used in this is based on Debian Linux, which provides the users to access lots of open source network and computing security packages like Snort, Cowrie, Dionaea , Glastopf and many more. All these collect the vulnerabilities which are exploited by the malware. The goal is to collect the patterns of the malware.

Now, alongside Raspberry Pi there are many other low-cost microcomputers available, so why did we choose Raspberry Pi?

We reviewed many other microcomputers to check their efficiency. CubieBoard is microcomputer which costs about 45$, has CPU clock speed of 1 gigahertz , RAM of 1 gigabits, Flash memory of 4 gigabits and requires a power of 5 watts. Beagle Board was an another microcomputer which costs about 150$, has CPU clock speed of 720 megahertz , RAM of 256 megabits, Flash memory of 2 gigabits and requires a power of 2 watts. Via APC was an another one, which costs about 49$, has CPU clock speed of 700 megahertz , RAM of 512 megabits, Flash memory of 2 gigabits and requires a power of 13.5 watts. Now the Raspberry Pi costs about 25$, has CPU clock speed of 700 megahertz, RAM of 512 megabits and requires a power of 5 watts.

So from this information it is very clear that Raspberry Pi has the lowest cost as compared to others and the power consumption is still doing well. Looking at the cost factors we can eliminate Beagle Board and Via APC from our list. CubieBoard provides much better features as compared to Raspberry Pi by sending just 20$ extra on it. But since in our research work we are going to use Dionaea honeypot, which runs on old Pentium processor, we decided to use Raspberry Pi for running Dionaea for our project.

Honeypots are basically classified into two broad catagories:

- Malware Honeypots
- Cloud Honeypots

We are going to deploy a malware honeypot. The most important feature of the malware honeypot is that they are able to detect botnets by capturing malware. The collection of the malware follows “know your enemy” policy by creating signatures from the malicious data. The collection of malicious data is a non- trivial task. In order to overcome the difficulties of the manual approach, Nepenthes platform was designed.

In our research work we are deploying Dionaea via the honeypot management system MHN (Modern Honey Network). This management system makes the deployment of the honeypots very quick and easy, just by executing a bunch of simple commands. It was first efficient honeypot designed to collect malware. It falls in the category of low interaction honeypot. It was really good scheme as it was able to emulate the most vulnerable services, those which were

even opposed by the high interaction honeypots. Nepenthes was the first attempt in collecting malware for detecting botnets.

Dionaea is the successor of Nepenthes. It is also a low interaction honeypot which is used to capture malware. Malwares are then collected, analyzed and then sent to online sandboxes like CWSandbox, Virus Total and Norman Sandbox.
The services emulated by Dionaea are:

- ftp(port 21/tcp)
- http/https(port 80/tcp and port 443/tcp)
- sip/sip-tls(port 5060/tcp and 5061/tcp)
- mysql (port 3306/tcp)
- mssql (port 1433/tcp)
- tftp (port 69/udp)
- smb(port 445/tcp)
- nameserever (port 42/tcp)
- msrpc (port 135/tcp)

After collecting the malware from the honeypot, the malware need to be scanned which could be done with any of the available online scanners like:
Metascan Online, (38) which has the following features:

- Time it takes to scan and upload 400 KB File: 76 seconds
- It does hash searching but does not scan remote files.
- Report Page Information: MD5/SHA1/SHA256, file size, detection, Analysis date, detection ratio via badge, individual AV engine scan time and definition date used.
- Sharing of uploaded files with antivirus vendors: YES
- Upload progress meter: NO
- Upload method: Web + SSL
- Max upload size: 50 MB
- Antivirus Engine: 42

VirScan, which has the following, features:

- Time it takes to scan and upload 400 KB File: 270 seconds
- It does not hash searching but does not scan remote files.
- Report Page Information: MD5/SHA1, file size, detection, Analysis date, detection ratio, individual AV engine definition date and engine version
- Sharing of uploaded files with antivirus vendors: YES
- Upload progress meter: Yes with detailed progress
- Upload method: Web
- Max upload size: 20 MB
- Antivirus Engine: 37

Jotti, which has the following, features:

- Time it takes to scan and upload 400 KB File: 55 seconds
- It does hash searching but does not scan remote files.
- Report Page Information: MD5/SHA1, file size, detection, Analysis date, detection ratio via badge, individual AV engine scan time and definition date used.
- Sharing of uploaded files with antivirus vendors: YES
- Upload progress meter: YES
- Upload method: Web
- Max upload size: 25 MB
- Antivirus Engine: 20

NoVirusThanks, which has the following, features:

- Time it takes to scan and upload 400 KB File: 76 seconds
- A separate box is made in order to scan remote files by entering a direct link in it without downloading the file to your first computer
- Report Page Information: MD5/SHA1/SHA256, file size, detection, Analysis date, detection ratio via badge, individual AV engine version used to scan.
- Sharing of uploaded files with antivirus vendors: OPTIONAL
- Upload progress meter: NO

- Upload method: Web
- Max upload size: not known
- Antivirus Engine: 14

VirusTotal, (38) which has the following features:

- Time it takes to scan and upload 400 KB File: 76 seconds
- It does hash searching but does scan remote files.
- Report Page Information: MD5/SHA1/SHA256, file size, detection, Analysis date, detection ratio via badge, individual AV engine scan time and definition date used.
- Sharing of uploaded files with antivirus vendors: YES
- Upload progress meter: YES
- Upload method: Web + SSL, Email Attachment, Desktop Browser, Android, Windows Context menu
- Max upload size: 32 MB
- Antivirus Engine: 46

Now, after studying about all the Scanners, it was observed that VirusTotal is the most suitable to use for malware analysis, since it is able to scan up to 46 Antivirus Engines and it is leading in all aspects like speed, URL scanning, multiple languages , voting and comment.

Shodan Search Engine

Shodan(39) is search engine which is used to discover particular type of computer based devices like routers, webcams and servers and many other devices which are connected to internet. It generally collects the data mostly using web servers, especially from port 80, 8080, 443, 8443, which are http/https ports, ftp port 21, telnet port 23, ssh port 22, snmp port 161, sip port 5060. It was designed by John Matherly, a computer programmer in 2003 but it was launched in 2009.

Chapter 4

Objective of Study

Earlier all the work has been done on both active and passive techniques for the attack detection. While these studies were good in finding patterns from the malicious data and these patterns were used for making signatures which were used for pattern matching by various fraud detection systems. But there is less work done on understanding the characteristics of the devices from which the attack is done or in understanding the characteristics of the compromised devices which are made to perform attacks, for instance the botnet compromised devices. Based on this, our objective is to understand the characteristics of these compromised devices, so that we come to know what sort of malware is being sent from particular device and we can create signatures according to that. This is an enhancement in the security of the system, since we will be able to retrieve patterns according to the compromised device instead of just the malware.

Chapter 5

Research Methodology

Setting up honeypots like Dionaea is not an easy task and is very much time consuming. There are generally two ways by which Dionaea can be deployed on Raspberry Pi. One is the easy one and the other one a little bit tedious but much reliable.

One is using Pi-pots, which are pre-loaded Raspberry Pi images. Pi-pots were designed by team of Indian Honeynet Project (37). Pi-pots contain various honeypot clients like Kippo, Dionaea, Glastopf and also many other softwares which are needed to run honeypot sensor. We just have to download these raspbian distributions and write it to the memory card. Then we can set up the sensor in very less time.

Firstly the basic requirements for deploying the pre-requisite set up:

- Raspberry Pi
- A SD Card(of 4gb or larger)
- HDMI cable
- A monitor with a HDMI input
- Ethernet Cable or a Network Connection
- Router or a Switch with an Ethernet Port
- USB Keyboard
- USB Mouse
- Power Supply (5 watts)

Installation on Windows:

1. Download zip file and extract the image file.

2. Insert the SD card into your SD card reader and check which drive letter was assigned. You can easily see the drive letter (for example G :) by looking in the left column of Windows Explorer.

3. Download the Win32DiskImager utility and extract the executable from the zip file and run the Win32DiskImager as administrator.

4. Select the image file you extracted above.

5. Select the drive letter of the SD card in the device box.

6. Click Write and wait for the write to complete. > Exit the imager and eject the SD card. > Now insert the SD into raspberry pi's slot and switch it on.

7. Use an NMAP ping scan to find out the IP address of raspberry pi. > Use port 2222 to make an SSH connection. The default username: password is pi: raspberry

8. Run "sudo raspi-config" and select "Expand Filesystem"

9. Click on finish and reboot, once rebooted ssh into the pi again. You are now ready to run honeypots.

The following commands can be used to run different honeypots:

Dionaea

=======

Run dionaea using the following commands:

> cd /opt/dionaea
> sudo ./dionaea -u nobody -g nogroup -r /opt/dionaea -w /opt/dionaea -p /opt/dionaea/var/dionaea.pid

Note: If you want to run it in the background then use: nohup dionaea -u nobody -g nogroup -r /opt/dionaea -w /opt/dionaea -p /opt/dionaea/var/dionaea.pid &

Kippo

=====

Use the following commands to run kippo:

> sudo su kippo

> cd

> cd kippo

> ./start.sh

Glastopf

=======

To run glastopf use the following commands:

> cd /opt/myhoneypot

> sudo glastopf-runner

Another method of installing Dionaea is:

Firstly we have to download the NOOBS LITE operating system for the Rasberry Pi. For this we require a formatted SD card. The SD Card firstly needs to be formatted using SD formatter 4.0. Then NOOBS LITE is downloaded, which is in the form of a Zip File. The file is then extracted on the SD card. Insert the card in the Pi setup and power on and install Raspian. After configuring all the settings of the operating system, we will install Dionaea.

1) In the terminal, run the command “ifconfig”. Check the ip address of your device. Note it down.
2) We are done with the raspberry Pi configuration. Now we need a host machine with MHN (Modern Honey Network) installed on it.

MHN server is supported by Ubuntu 14.04, Ubuntu 16.04 and Centos 6.9.

The following steps need to be followed to install MHN:

Install Git

```
# on Debian or Ubuntu
$ sudo apt-get install git -y

# on Centos or RHEL
$ sudo yum install -y git
```

Install MHN

```
$ cd /opt/
$ sudo git clone https://github.com/threatstream/mhn.git
$ cd mhn/
```

Run the following script to complete the installation. While this script runs, you will be prompted for some configuration options. See below for how this looks.

```
$ sudo ./install.sh
```

Configuration

```
============================================================
MHN Configuration
============================================================
Do you wish to run in Debug mode?: y/n n
Superuser email: YOUR_EMAIL@YOURSITE.COM
Superuser password:
Server base url ["http://1.2.3.4"]:
Honeymap url ["http://1.2.3.4:3000"]:
Mail server address ["localhost"]:
Mail server port [25]:
Use TLS for email?: y/n n
Use SSL for email?: y/n n
Mail server username [""]:
Mail server password [""]:
Mail default sender [""]:
Path for log file ["mhn.log"]:
```

Running

If the installation scripts ran successfully, you should have a number of services running on your MHN server. See below for checking these.

```
user@precise64:/opt/mhn/scripts$ sudo /etc/init.d/nginx status
 * nginx is running
user@precise64:/opt/mhn/scripts$ sudo /etc/init.d/supervisor status
 is running
user@precise64:/opt/mhn/scripts$ sudo supervisorctl status
geoloc                           RUNNING    pid 31443, uptime 0:00:12
honeymap                         RUNNING    pid 30826, uptime 0:08:54
hpfeeds-broker                   RUNNING    pid 10089, uptime 0:36:42
mhn-celery-beat                  RUNNING    pid 29909, uptime 0:18:41
mhn-celery-worker                RUNNING    pid 29910, uptime 0:18:41
mhn-collector                    RUNNING    pid 7872,  uptime 0:18:41
mhn-uwsgi                        RUNNING    pid 29911, uptime 0:18:41
mnemosyne                        RUNNING    pid 28173, uptime 0:30:08
```

Deploying honeypots with MHN:

MHN was designed to make scalable deployment of honeypots easier. Here are the steps for deploying a honeypot with MHN:

1. Login to your MHN server web app.
2. Click the "Deploy" link in the upper left hand corner.
3. Select a type of honeypot from the drop down menu (e.g. "Ubuntu Dionaea").
4. Copy the deployment command.
5. Login to a honeypot server and run this command as root.

If the deploy script successfully completes you should see the new sensor listed under your deployed sensor list.

After installing MHN on the host machine, open the terminal and run->

- ssh pi@{IP address we noted earlier} (Here, in the curly brackets enter the ip address of the Raspberry Pi)
- You will receive an alert about the authenticity of the host. Type “Yes” and press “Enter”. This will happen only one time when your host machine will make a ssh connection with the Raspberry Pi.
- Then type your password and press “Enter”.
- Open the MHN web interface into the browser. Click on “Deploy” Tab and select “Raspberry Pi Dionaea” from the displayed menu.
- Copy the deploy command on the terminal and run it. Since we have made a ssh connection with the Raspberry Pi, this command is run on the Pi actually.
- Once the script has run successfully, click on the “Sensors Tab”. If we find Raspberry Pi in the list of the sensors, this means Raspberry Pi is successfully deployed as a Dionaea honeypot.
- Dionaea stores the all the malware information in SQLite database residing on the honeypot.

- This Raspberry Pi honeypot will be deployed for few days and a web front-end called DionaeaFR can also be used in order to observe the status of the honeypot

Analysis after the collection of the Malware:

For the analysis of the malware collected, we are using VirusTotal tool. It is an online service and is freely available. It is scanner which is able to identify malicious files and URLs also. Here we are using to analyze the data captured by the honeypots. The copy of the Dionaea malware is then automatically submitted to the either to VirusTotal API or through email or web or VirusTotal uploader for the analysis. In the VirusTotal, the malware data will be analysed using almost 60 antivirus engines and the resulted scanned data will be stored in the honeypot database.

Identifying the Devices:

After the submission of the malware analysis and collection of all attack information, a Pyhton script will be used to get the IP addresses of the attackers and these IP addresses will tell us about the devices from which the attack is coming. The identification of the devices will be done by Shodan Search engine.

This research is unique as in this we are using a very cost effective Raspberry Pi based honeypot along with VirusTotal scanner and Shodan search engine to study the characteristics of the compromised devices. The administrator can improve the security postures by using these findings and will come to know about the vulnerabilities of the device.

Bibliography

1. Peter, E. and Schillar, T.(2008). A practical guide to honeypots. *http://www.cs.wustl.edu/~jain/cse571-09/ftp/honey.pdf.*[accessed on 26/10/2017]

2. honeynet Project (2006). Know your enemy: Honeynets. *http://old.honeynet.org/papers/honeynet.* [accessed on 26/10/2017]

3. Piscitello, D.(2001). Honeypots: Sweet idea, sticky business. *http://www.corecom.com/external/livesecurity/honeypots.html.* [accessed on 22/10/2017]

4. Furche, J. and Elingehausen, R. (1999). Cybercop sting,getting started guide version 1.0.

5. Schneier, B. (1999). Honeypots and the honeynet project. *https://www.schneier.com/crypto-gram/archives/2001/0615.html#1*. [Accessed on 28/10/2017]

6. Desaster (2014). Kippo -ssh honeypot. *https://github.com/desaster/kippo.* [Accessed on 29/10/2017]

7. Wikipedia (2017j). Twisted(software). *https://en.wikipedia.org/wiki/Twisted_(software).* [Accessed on 29/10/2017]

8. Nicholson, T. (2016). Honssh. *https://github.com/tnich/honssh/wiki.* [Accessed on 21/10/2017]

9. Lukas Rist, Sven Vetsch, M.K.M.M(2010). Know your tools: Glastopf. *http://honeynet.org/sites/default/files/files/KYT-Glastopf-Final_v1.pdf.* [Accessed on 11/10/2017]

10. Dell'Area, A. (2016). Thug. *https://github.com/buffer/thug.* [Accessed on 22/9/2017]

11. (2016). Cowrie ssh/telnet honeypot. . *https://github.com/micheloosterhof/cowrie.* [Accessed on 25/9/2017]

12. Welcome to dionaea's documentation! (2015).*httpss://dionaea.readthedocs.io/en/latest/.* [Accessed on 21/9/2017]

13. Wikipedia (2017i). Tr-069. *https://en.wikipedia.org/wiki/TR-069.* [Accessed on 12/10/2017]

14. Phype(2016). Pyhton tenet honeypot for catching botnet binaries. *https://github.com/Phype/telnet-iot-honeypot.* [Accessed on 15/10/2017]

15. Wikipedia (2016b). Cymmetria. *https://en.wikipedia.org/wiki/Cymmetria.* [Accessed on 01/10/2017]

16. Yin Minn Pa Pa, Shogo Suzuki, K. Y. T. M. T. K. C. R. (2015). Iotpot: Analysing the rise of iot compromises. *http://christian-rossow.de/publications/iotpot-woot2015.pdf.* [Accessed on 17/10/2017]

17. Cymmetria (2016). Open Source Telnet Honeypot. . *https://github.com/Cymmetria/MTPot.* [Accessed on 30/10/2017]

18. Omer Erdem (2015) HoneyThing. *https://github.com/omererdem/honeything.* [Accessed on 22/9/2017]

19. D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," Ad Hoc Networks, vol. 10, no. 7, pp. 1497–1516, 2012.

20. L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

21. S. Horrow and A. Sardana, "Identity management framework for cloud based internet of things," in Proceedings of the First International Conference on Security of Internet of Things. ACM, 2012, pp. 200–203.

22. H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: a review," in Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on, vol. 3. IEEE, 2012, pp. 648–651.

23. X. Xiaohui, "Study on security problems and key technologies of the internet of things," in Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on. IEEE, 2013, pp. 407–410.

24. D. Kozlov, J. Veijalainen, and Y. Ali, "Security and privacy threats in iot architectures," in Proceedings of the 7th International Conference on Body Area Networks. ICST (Institute for Computer Sciences, Social- Informatics and Telecommunications Engineering), 2012, pp. 256–262.

25. R. M. Savola, H. Abie, and M. Sihvonen, "Towards metrics-driven adap- tive security management in e-health iot applications," in Proceedings of the 7th International Conference on Body Area Networks. ICST (Insti- tute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 276–281.

26. A. Kanuparthi, R. Karri, and S. Addepalli, “Hardware and embedded security in the context of internet of things,” in Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles. ACM, 2013, pp. 61–64.

27. D.-Y. Kim, “Cyber security issues imposed on nuclear power plants,” Annals of Nuclear Energy, vol. 65, pp. 141–143, 2014.

28. D. E. Denning, “Stuxnet: What has changed?” Future Internet, vol. 4, no. 3, pp. 672–687, 2012.

29. S. Alampalayam and A. Kumar, “An adaptive and predictive security model for mobile adhoc networks,” Wireless Personal Communications, vol. 29, no. 3-4, pp. 263–281, 2004.

30. E. Rescorla and N. Modadugu, “Datagram transport layer security version 1.2,” 2012.

31. T. Dierks and C. Allen, “Rfc 5246-the tls protocol, 2008,” 2014.

32. R. Aggarwal and M. L. Das, “Rfid security in the context of internet of things,” in Proceedings of the First International Conference on Security of Internet of Things. ACM, 2012, pp. 51–56.

33. J. Liu, Y. Xiao, and C. P. Chen, “Authentication and access control in the internet of things,” in Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on. IEEE, 2012, pp. 588–592.

34. A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier, “The internet of things for ambient assisted living,” in Information Technol- ogy: New Generations (ITNG), 2010 Seventh International Conference on. Ieee, 2010, pp. 804–809.

35. H. Tao and W. Peiran, “Preference-based privacy protection mechanism for the internet of things,” in Information Science and Engineering (ISISE), 2010 International Symposium on. IEEE, 2010, pp. 531–534.

36. L. You-guo and J. Ming-fu, “The reinforcement of communication security of the internet of things in the field of intelligent home through the use of middleware,” in Knowledge Acquisition and Modeling (KAM), 2011 Fourth International Symposium on. IEEE, 2011, pp. 254–257.

37. Indian Honeynet Project. *http://honeynet.org.in/raspberry-pie-sensor-setup/*. [Accessed on 29/10/2017]

38. Seven Online Multi-Engine Antivirus Scanners to Scan Suspicious Files. *https://www.raymond.cc/blog/battle-of-the-6-online-malware-file-scanners/*. [Accessed on 26/9/2017]

39. Shodan(Website). *https://en.wikipedia.org/wiki/Shodan_(website)*.[Accessed on 22/10/2017]