A Survey on Security Concerns in Internet of Things

Bhavkanwal Kaur
School of Computer Science and Engineering
Lovely Professional University
Jalandhar-144411
bhavkanwal2193@gmail.com

Pushpendra Kumar Pateriya
School of Computer Science and Engineering
Lovely Professional University
Jalandhar-144411
pushpendra.mnnit@gmail.com

***Abstract*—Internet of Things is the kinship using Internet of computing equipments ingrained in everyday gadgets which facilitate them to deliver and retrieve data. But the use of such gadgets was not encouraged to much greater extent due to many security concerns arising with them. 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. In this paper, we are producing a review on various security outbreaks that can take place on different layers that constitute IoT. Distinct mechanisms are examined that present clarifications to these security concerns along with their disadvantages. Also we have discussed about honeypots which are a much better security scheme that can prevent the attacks from occurring so that there is no need to detect them. Here we have also explained setting up a honeypot for resource constrained IoT devices using a simple illustration. Future work is also discussed such that strength and safety of IoT should enhance and their various devices get more protection against many familiar attacks.**

Keywords- Internet of Things, Preservation methods, Attacks and threats, Security concerns

I. 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.[1] 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.[2]

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 a 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 an "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. [3] 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. In order to make this new technology reliable, these IoT appliances need very big improvement in terms of security and privacy. The written article is formulated in the following manner. Firstly, IoT layer model is discussed explaining the functionality of each layer which builds up the complete IoT model and in this section, the concerns regarding the security of each layer is also discussed. In the next section we discussed about the various protocols available for providing security in IoT and what techniques can be adopted for IoT protection. Finally we present the Future scope in enhancing security in IoT, then summary and conclusion.

II. IOT LAYERS AND SECURITY PROBLEMS RELATED TO THEM

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

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

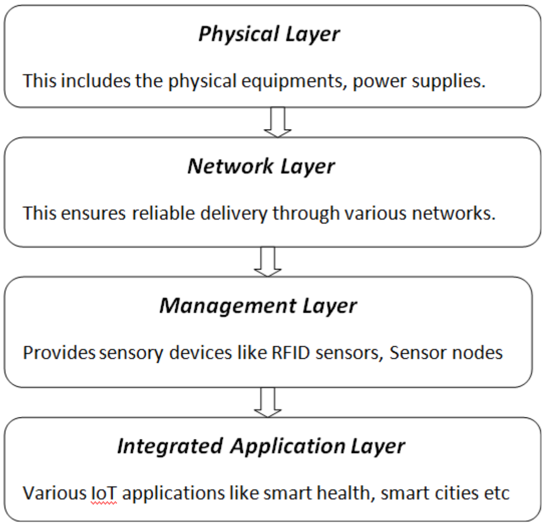

Physical Layer
This includes the physical equipments, power supplies.
Network Layer
This ensures reliable delivery through various networks.
Management Layer
Provides sensory devices like RFID sensors, Sensor nodes
Integrated Application Layer
Various IoT applications like smart health, smart cities etc

Fig. 1. IoT Layered Model

foundation of such hardware components.

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 there is any power cut or blackout, they must be efficient enough to work better on their battery power.[4][5][6].

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. 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. [7]

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.[8]

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. [4][5][6]

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. Also, in areas of atomic power plants, if there is a mobile node carrying many software viruses and the software are not updated in time, it can lead to many disastrous results. [9][10]

Table: I give a brief review of security concerns of all the IoT Layers.

III. 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.[11]The integration of the IP protocols with the smart devices can happen only if the architecture of the smart devices are 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. [12] IPsec(Internet Protocol Security) provides secure exchange of

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 information leads to loss of data, placing malicious nodes near to network nodes for data sniffing, Masquerading attacks
Application Layer	No updation of security bugs in the software of the device, Changing the IoT device environment in order to make it receive wrong data

data at the network layer among the different IoT nodes.[13] IPsec, used in IPv4, was actually developed for the IPv6. IPsec is intact-ed 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.

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. 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.

IV. 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.[14]

Lui et al., introduces an approach to protect the system against eavesdropping, man-in-the middle attack, replay 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.[15]

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.[16]

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.[3]

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. [17]

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.[18]

All the above schemes are better at preventing the attacks. But what if there is 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 is marai, which is a malware that attack the devices with weak login credentials and then use these infected devices to spread the infection further. In order to capture such attacks at the early stage honeypots are used.

V. HONEYPOTS

Honeypot is basically a system designed to attract and capture the attacker. It attracts the attacker by making itself vulnerable to attacks. 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.

A. Classification of honeypots

1) Based on interaction: High interaction honeypots[19] 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. Medium interaction honeypots lay in between high and low interaction honeypots.

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.

PREVIOUS WORK ON HONEYPOTS

Deception Toolkit (DTK) [20] 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 is not even a high interaction honeypot so get compromised easily.

After Deception Toolkit, Cyber Cop 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.

A. 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.

1) Kippo: Kippo is a medium interaction SSH honeypot 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. This is a very popular model and is used by others to create a many other types of honeypots.

2) HonSSH: HonSSH 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.

3) Glastopf: Glastopf [21] 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 basically sends a reply expected by the attacker when the attacker is trying to access the web service.

4) Thug: Thug is a client side honeypot [22]. 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.

5) Cowrie: Cowrie is a fork of Kippo [23]. It supports both Telnet and SSH. It emulates SFTP and SCP protocols. It supports more of the Linux commands.

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

B. IoT Honeypots

1) HoneyThing: HoneyThing[24] was designed as a part of GSoC project. It was created for the Internet of TR-069 things. It emulates a router which supports CWMP protocol and which has a web interface. It emulates some of the most famous vulnerabilities in IoTs.

2) Telnet-iot-honeypot: Telnet-iot-honeypot is used for IoT devices to capture Telnet attacks. It is mainly used to capture malware of botnets and binaries. It is basically written in Python.

3) MTPot: MTPot is open source honeypot introduced by Cymmetria Research. 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.

4) IoTPot: IoTPoT came into existence with the collaborative hard work of the researchers from the Germany and Japan [25]. 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.

VI. EXPLANATION OF SETTING UP A HONEYPOT IN RESOURCE CONSTRAINTED ENVIRONMENT USING AN ILLUSTRATION

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. 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.

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.

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 dollar, 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 dollar, 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 dollar, 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 dollar, 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 dollar 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.

In our research work we are deploying Dionaea via the honeypot management system MHN (Modern Honey Network). 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:

1) ftp(port 21/tcp)
2) http/https(port 80/tcp and port 443/tcp)
3) sip/sip-tls(port 5060/tcp and 5061/tcp)
4) mysql (port 3306/tcp)
5) mssql (port 1433/tcp)
6) tftp (port 69/udp)
7) smb(port 445/tcp)
8) nameserever (port 42/tcp)
9) 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:

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

VirScan, which has the following, features:

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

Jotti, which has the following, features:

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

NoVirusThanks, which has the following, features:

1) Time it takes to scan and upload 400 KB File: 76 seconds.
2) 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.
3) Report Page Information: MD5/SHA1/SHA256, file size, detection, Analysis date, detection ratio via badge, individual AV engine version used to scan.
4) Sharing of uploaded files with antivirus vendors: OPTIONAL
5) Upload progress meter: NO
6) Upload method: Web
7) Max upload size: not known
8) Antivirus Engine: 14

VirusTotal, (38) which has the following features:

1) Time it takes to scan and upload 400 KB File: 76 seconds
2) It does hash searching but does scan remote files.
3) Report Page Information: MD5/SHA1/SHA256, file size, detection, Analysis date, detection ratio via badge, individual AV engine scan time and definition date used.
4) Sharing of uploaded files with antivirus vendors: YES
5) Upload progress meter: YES
6) Upload method: Web + SSL, Email Attachment, Desktop Browser, Android, Windows Context menu.
7) Max upload size: 32 MB
8) 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.

VII. A SIMPLE METHODOLOGY FOR SETTING UP A RASPBERRY PI BASED HONEYPOT FOR IOT DEVICES

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. [26] Pi-pots contain various honeypot clients like Kippo, Dionaea, Glastopf and also many other software 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:

1) Raspberry Pi
2) A SD Card(of 4gb or larger)
3) HDMI cable
4) A monitor with a HDMI input
5) Ethernet Cable or a Network Connection
6) Router or a Switch with an Ethernet Port
7) USB Keyboard
8) USB Mouse
9) 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 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 File system
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 Dionaea honeypot:

- 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

Another method of installing Dionaea

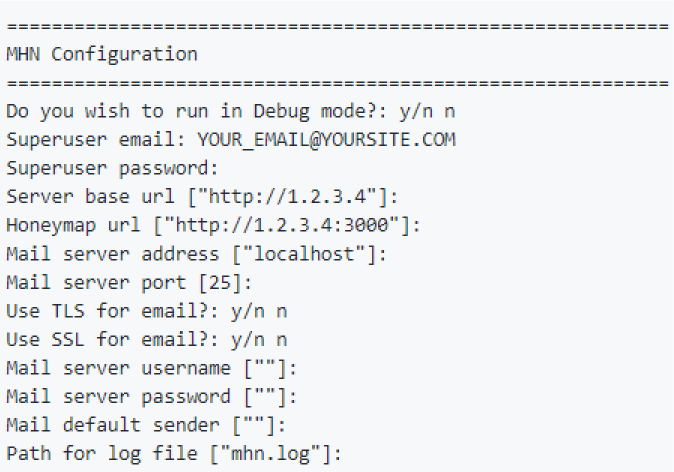
Firstly we have to download the NOOBS LITE operating system for the Rasberry Pi. For this we require a formatted SD card. The SD Card firstly needs to be formatted using SD formatter 4.0. Then NOOBS LITE is downloaded, which is in the form of a Zip File. The file is then extracted on the SD card. Insert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

- In the terminal, run the command ifconfig. Check the ip address of your device. Note it down.
- 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:

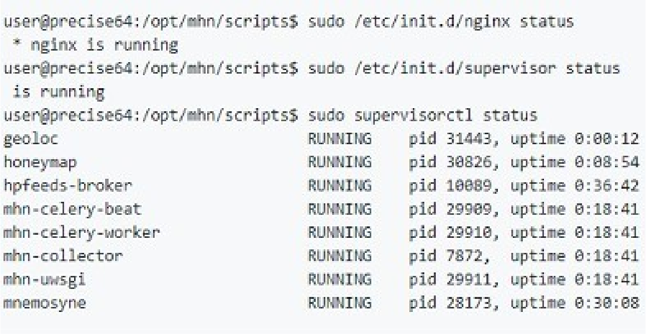
Install Git
On Debian or Ubuntu:
sudo apt-get install git -y
On Centos or RHEL:
sudo yum install -y git

Install MHN
cd /opt/
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 ofhoneypots 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:

1) ssh pi@IP address we noted earlier (Here, in the curly brackets enter the ip address of the Raspberry Pi)
2) 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.
3) Then type your password and press Enter.
4) Open the MHN web interface into the browser. Click on Deploy Tab and select Raspberry Pi Dionaea from the displayed menu.
5) 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.
6) 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.
7) Dionaea stores the all the malware information in SQLite database residing on the honeypot.
8) 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 of collection of malware

For the analysis of the malware collected, VirusTotal tool can be used. It is an online service and is freely available. It is scanner which is able to identify malicious files and URLs also. The data captured by the honeypots can be analyzed using VirusTotal. 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 analyzed using almost 60 antivirus engines and the resulted scanned data will be stored in the honeypot database.
Shodan 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. 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 can be done by Shodan Search engine.

VIII. FUTURE WORKS

A lot of work needs to be done at the security level in IoT. New approaches need to be embarked to provide security at the IP and the transport layer such that they do not rely on the intervening nodes for the reliable transmission of the data.

Man in the Middle attack and eavesdropping is quite common in IoT devices. Such schemes should be introduced which uses security protocols that generate an alarm rate in the real time when these type of attacks are crossing a certain threshold which is set every time according to a particular threat index value.

For the authentication of the data and authorization of the user, many security techniques are used which involve encryption algorithms, but these require a lots of resources and memory and much time to implement. Such encryption need to be introduced which consume less power and faster than the existing ones.

The proposal made regarding the security of IoT devices 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.

IX. CONCLUSION

As the IoT based appliances are increasing day by day and there is tremendous growth happening in this field, the growth in the number of attacks on them will also increase because these appliances are Internet based, as they can access connectivity through Internet only, the more and more attacks are possible on them when they expand. In this paper, we have discussed about each layer of IoT model and their vulnerabilities. Then we discussed which security protocols are there for IoT. At the end we discussed about a few of the security schemes being introduced and what are the limitations associated with them. Further, a lot of work needs to be done in the order to remove the limitations of the above discussed schemes and many protocols need to be developed for the implementation of the above introduced schemes.

REFERENCES

[1] 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.

[2] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] R. M. Savola, H. Abie, and M. Sihvonen, “Towards metrics-driven adaptive security management in e-health iot applications,” in *Proceedings of the 7th International Conference on Body Area Networks*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 276–281.

[8] 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.

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

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

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

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

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] E. Peter and T. Schiller, “A practical guide to honeypots,” *Washington University*, 2011.

[20] D. Piscitello, “Honeypots: Sweet idea, sticky business,” *TISC Insight*, vol. 3, no. 2, 2001.

[21] L. Rist, S. Vetsch, M. Kossin, and M. Mauer, “Know your tools: Glastopf-a dynamic, low-interaction web application honeypot,” *The Honeynet Project*, vol. 4, 2010.

[22] R. Lizzi, “Expo 2015: An opportunity to relaunch italy against the background of local and global challenges,” *Italian Politics*, vol. 31, no. 1, pp. 209–224, 2016.

[23] P. Krishnaprasad, “Capturing attacks on iot devices with a multi-purpose iot honeypot,” Ph.D. dissertation, INDIAN INSTITUTE OF TECHNOLOGY KANPUR, 2017.

[24] Ö. Erdem, “Honeything: nesnelerin interneti için tuzak sistem,” Ph.D. dissertation, 2016.

[25] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, “Iotpot: analysing the rise of iot compromises,” *EMU*, vol. 9, p. 1, 2015.

[26] L. Spitzner, “The honeynet project: Trapping the hackers,” *IEEE Security & Privacy*, vol. 99, no. 2, pp. 15–23, 2003.