Pressing Importance of IoT Security

Malav Vyas

Abstract— IoT Devices, can be found almost everywhere.Due to misconfigurations, IoT devices allow attackers to wreak havoc. In this Paper, I try to elucidate these misconfigurations and possible impact of them and try to explain the pressing need of IoT security and explain my findings.

1 INTRODUCTION

According to proposed defination by Haller et al, IoT is "A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active partcipants in business process"[1]

Now-a-days, any uniquely addressable objects can be interconnected through network [2], [3]

Due to broad applications, IoT devices are used in a lot of industries, including healthcare, life sciences, municipal infrastructure, agriculture, smart home, retail, manufacturing, education and automation

Use of IoT devices is increasing at an alarming rate. According to former chief futurist at Cisco and CIO/VIP of Technology at the Computer History musem,

"Today, literally anything can be connected, including tennis rackets, diapers, clothing, vehicles and, of course, home."

As per a report by forbes, by 2020 annual revenue of IoT vendoes could exceed $470B [4]. Due to wide spread usage and less awareness about effects of an IoT device compromise, has attracted hackers.

The IoT security is one of the wings of Information security, which concerns with safeguarding connected devices and networks in the Internet of things (IoT)

IoT technology involves giving every device a unique identifier, which enables it to connect over the internet.

If not configured properly, due to the ease of connectivity, anyone can perform malicious activities and lead to financial and in some cases even loss of lives.

The overview of the same can be better elucidated with an example

Not just simple IoT ecosystems like smart home can be hack, past is the witness that devices that are critical to humans can also be hacked and can deliver much more consequencies.

Insulin pumps, which are just like a normal pager in size, can provide very ease to patients, minimizing the hassle of injecting insulng 4-7 times a day,

These devices can deliver insulin doses as small as 0.25 units that can be given more constantly.

These devices have some wireless capabiliies which makes data entry from an external blood glucose easier.

Which also makes it easier for an attacker to take advantage of the improper configurations and gain access to settings of device. Attacker can also configure device in a way that it'll lead to a higher amount of insulin, much higher than safe, and put victim's life in danger[5].

This attack was explained and elaborated in a blackhat USA conference more than a decade ago, still this research on wireless security on medical devices has only scratched the surface of what vulnerabilities exist.

There are a lot of threats related to these innocent looking IoT devices.

On the surface IoT related threats can be devided in two sections:

1) Threats against IoT and 2) Threats by IoT

1.a Threats against IoT

These are the threats directly aimed towards IoT devices for compromise. It can result from complete takeover of a device to complete takeover or compromise of an IoT architecture (e.g. A critical Infrastructure like an electrical grid).

Various other Iot devices have also been hacked in the past. IP cameras can be hacked through buffer overflow attacks, Philips Hue lightbulbs were hacked through its ZigBee link protocol.

While attacker can make a device completely useless, it is also possible for possible stealing of data and Identity theft leading to financial consequences.

1.b Threats by IoT

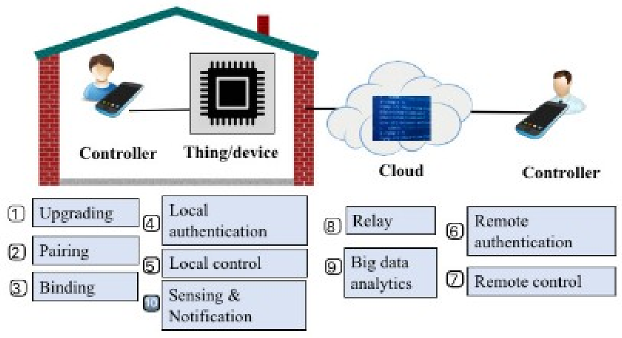
These are the threats posed by IoT devices to other computer systems.

An Unmanned Arial Vehicle (UAV) can fly far away and compromise the privacy and security of the people on the ground. UAV can also be made to fly over sensitive infrastructure with national importance and can result in numerous consequences.

Understanding it better with an example, The Mirai Botnet, attacked a huge number of devices, compromising them and using them to perform a Distributed Denial of Service attack on popular websites, and making them inaccessible. This outage resulted in the loss of billions of dollars. [

Such vulnerabilities still exist in our ecosystem.

This possibility and lack of awareness of possible impact of the compromise of Iot devices, can lead to another big attack like Mirai attack, affecting households as well as businesses.



2 Building blocks for IoT systems

In this section, view on building blocks of an IoT system is presented. Focus is on a standalone IoT system as shown in Figure 1.

2.1 Sensors:

They can be considered as the front end of IoT Devices, they collect data from the surroundings or give data to their surrounding, i.e. actuators.

These sensors need to have unique identity, i.e IP Address, for them to be easily identified in a network.

They collect data in real time actively while working autonomously or user controlled, depending on requirements.

2.2 Processors:

These are the brains functioning to process collected data and making a sense out of them, working in real time and instructing other componenets on how to perform.

2.3 Gateways:

Just as the name suggests, gateways provide a route to data to traverse through the network delivering to proper location for proper utilization.

2.4 Applications:

Possible cloud-based applications are responsible for utilizing and rendering meaning out of the collected data.

They also provide user with an interface to control as well as monitor the behavior of an IoT system.

Let us understand the structure by example.

Consider an IoT system for automatic door.

For the sensors, there could be one or multiple IR Blasters (InfraRed Blasters), which constantly collects data from surroundings, which is, the information of if anyone is in the proximity of the door.

This Information is then passed to the processor (i.e. MicroController).

The processor analyzes data and renders if anyone is in proximity or not. If yes, then it indicates other sensors (i.e. acumlators) to open the door.

It can also command the sensors to open or close the door via the same cloud medium.

3 Attack surface for IoT devices

The sum Total of all the potential security vulnerabilities in IoT devices, their software as well as infrastructure in a given network is the attack surface for IoT devices

Attackers can use vulnerabilities in IoT devices to gain access to a network for various purposes compromising privacy and security of users.

So, it becomes viable to assess and map out all the surfaces, i.e. interfaces present in an IoT network and fix them before they are exploited.

According to Joshua Corman, Chief technology officer at Sonatype,

"You are taking things that weren't connected and weren't vulnerable and putting vulnerability and connectivity in all of them."[6]

Concerned with the threats presented by increased use and scope of IoT devices as well as rapidly growing IoT attack surface, the FBI released a public service announcement, "Internet of Things poses opportunities for cybercrime" which warns about potential vulnerabilities and advises protective measures that can mitigate risk associated with them.

There are a number of possible IoT attack surfaces due to wide spectrum of use cases, which can be deviced into four major componenets[7]:

1. Mobile
2. Communication
3. Cloud
4. Device

3.1 Mobile

Being the source of insights into the state of physical world for users, mobile acts as one of the important user interfaces for IoT.

Mobile, while containing applications for communication with IoT eco-system, for sending commands and a medium for data, becomes one of the entry point into IoT system.

If compromised, mobile can lead attacker to manipulate IoT architecture to their needs and can also extract sensitive information like account information, tokens, etc.

These can be considered sensitive attack surfaces, i.e entry points, available in a mobile device.

- Authentication
  - If not properly implemented, this can lead to un-verified and un-authenticated access to IoT system.
  - i.e. no password or easily guessable password on a mobile device
- communication
  - If not properly implemented, this can allow attackers to intercept the traffic to-and-fro IoT/Mobile device
- Encryption
  - Passwords and keys stored in plain text can allow attackers to use them after a successful attempt at extracting them out of an insecure storage or communication medium
- Storage
  - If misconfigured, i.e. open SD card slot, easy and unrestricted access to the memory, can allow attackers to steal data from the device.
- Standard Mobile vulnerabilities
  - Standard vulnerabilities like Owasp Top 10, if exploited, can result in complete mobile device as well as IoT device compromise[8]

3.2 Cloud

Cloud, where all data of an IoT system converges, is a very interesting attack point.

Cloud additionally has the privilages to send commands to all connected devices and can be used to manipulate the behavior of them if compromised. This can also lead to another botnet or even more severe attack like mirai malware.

In addition to attack surfaces like Authentication, Encryption, Storage and communication present in mobile component, Cloud contains APIs and Generic Web/Cloud vulnerabilities

- APIs
  - Vulnerable API endpoints can lead to compromise of complete cloud instance as well as IoT devices connected to it
- Generic Web/Cloud Vulnerabilities
  - Generic web/ cloud vulnerabilities like mentioned in OWASP web top 10 produced by misconfiguration of cloud instances or flaws in web design, can also lead to the same consequences as API attack surface[9]

3.3 Device

Devices can be attacked via storage component or user

interfaces.

Improper configuration SDcard, USB or volatile/Non-volatile memory can be used to perform the attack.

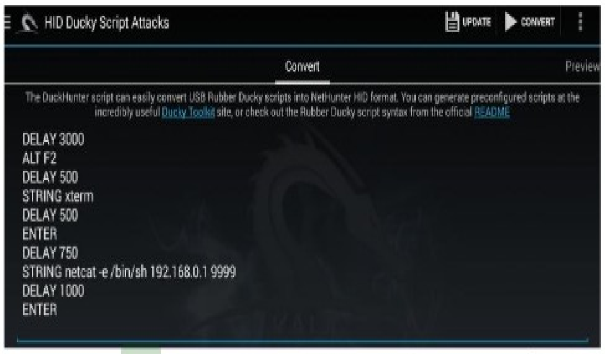
Micro-controller's internal memory can also be overwritten to exploit the IoT system.

Improper design, and easy access to Storage, can lead to firmware extraction.

Apart from devices communicating over network, different hardware components, of the same board, need to communicate to each other and to the outside world.

Most common interfaces like Universal Asynchronous Receiver Transmitter (UART), or Microcontroller debug ports, used for run-time debugging like, Joint Test Action Group (JTAG), Compact JTAG (CJTAG), Serial Wire Debug (SWD) can be leveraged to perform attack when configured improperly as they are capable of read/write firmware and microcontroller internal memory while also controlling microcontroller pins post production.

Apart from storage, Interfaces like, Human Machine Interface (HMI), i.e touch screens, push buttons, touchpad allowing unrestricted and unauthorized control to the system can also be considered as a threat to IoT eco-system.[10]



The DuckHunter script can easily convert USB Rubber Ducky scripts into NetHunter HID format. You can generate preconfigured scripts at the incredibly useful Ducky Toolkit site, or check out the Rubber Ducky script syntax from the official README

```
DELAY 3000
ALT F2
DELAY 500
STRING xterm
DELAY 500
ENTER
DELAY 750
STRING netcat -e /bin/sh 192.168.0.1 9999
DELAY 1000
ENTER
```

3.4 Communication

Communication, implemented with various possible protocols and mediums, can contain severe vulnerabilities if not implemented correctly. It can result into loss of data or complete takeover of the device if attacker can sniff the network and capture the data bits travelling towards/from the device.

In addition to Authentication and encryption attack surfaces, communication componenet has 2 additional attack surfaces.

Deviation from standard protocol and anomalies in protocol implementation, both resulting in either Denial of service or complete takeover of the device

Many IoT devices use wireless or Bluetooth as a medium for communication.

IoT, being a networked system, the whole system has to be secured from end to end. All traffic traversing in the network should be encrypted to prevent the leak of the sensitive information while implementing authentication carefully.

Most IoT devices allow any controller in proximity for pairing, while risk posed by this practice is small in a private setting like home, however, for deployment on large-scale in a public environment, this practice could pose serious threat.

Anyone in the proximity and with access to device can reconfigure the system and may break into the system.

Notorius attack, mirai DDos was made possible by such weak authentication on various IoT devices.

4 Exploiting IoT devices via storage componenet

Apart from households, and businesses, IoT devices are also used in production chains.

In large scale infrastructures implementing IoT devices on huge scale, it becomes a necessity to constant keep them in check and under repeated security audits.

One of the most comman way industrial Iot devices are attacked, is exposed storage components.

Many devices like collaborative robots contain exposed SD card port or USB port that can be used to compromise the whole infrastructure.

Keeping SD card and USB ports open can lead to some serious consequences.

Some of them are, Denial of Service due to theft of SD card,

Theft of data and sensitive information stored on SD card.

Theft of API keys used to program the device, which can also lead to compromise of cloud service connected, in turn compromising the whole IoT device Infrastructure.

There are whole lot of other attack vectors associated with expose USB Port.

Ease of access of USB port can lead to Human Interface Device Attacks, abbrivated as HID Attacks. [11]

HID attacks are performed by a tiny chip that masks itself as a USB drive, making victims belive that it is just a simple USB drive.

When someone plugs it in victim system, it acts as a USB keyboard device. It sends malicious keystrokes on victim system while delivering and executing payload.

This attack is still undetectable.

The USB Rubber Ducky device by Hack5 is available and anyone can use it to perform the attack.

However, devices like Arduino Uno, Digispark Digiborard etc. can be programmed to work and deliver payloads just like USB rubber ducky, even at cheaper cost.

While doing my research in HID attacks, I found a way to build multiple devices to perform the attack on the large scale. I also built a program to perform the same which can be found out on my GitHub repository.

HID devices generally perform the task with the help of a library called “keyboard.h”.

This library enables USB devices like Arduino to act as a keyboard.

While not every ASCII character can be sent with the library, the characters available are enough for an attacker to perform the attack.

While typing stuff on victim device is possible, attacker can also execute malicious command with opening terminal or command prompt on victim device with administrative permissions.

Impact of the attack performed with HID devices can be drasticly increased if it is targeted towards critical infrastructure.

HID attack combined with a little bit of social engineering can wreak some serious havoc.

HID devices can get attacker access to some files, a system or in some cases complete takeover of whole infrastructure, However, another type of attack is also possible.

Devices named USB killer, while not giving access to any files or system to attacker, can empower attacker to completely wreak system, performing a Denial of Service (DOS) attack on system.

This USB killer device, when plugged into a device, rapidly charges its capacitors from power lines of victim device’s USB port.[12]

After capacitors are charged, -200V is discharged to the same USB port’s DATA line.

This cycle of charge discharge continues until USB killer is removed or device circuit is completely broken.

So, there are numerous ways an attacker can compromise an IoT device or even takeover complete infrastructure if USB ports are kept open and kept accessible to anyone for use.

Mitigation:

While attacks by HID devices are mostly dependent on users of victim devices, it can be reduced to a level if USB ports on device are kept obfuscated.

There are several devices like USB killer test shield, USB condom etc. to prevent USB killer as well as juice jacking by completely disallowing data connection passed over a USB cable.

5 Case Study

According to a security research of United States based natural gas utility operator by Sepio Systems, some sensitive documents were stolen from an air-gapped network.[13]

According to initial investigation, As the network was air-gapped, there was no possible way that the documents were leaked through internet.

While usage of all removable storage devices was prohibited in the premises, the possibility of data theft by copying file into removable media was also ruled out.

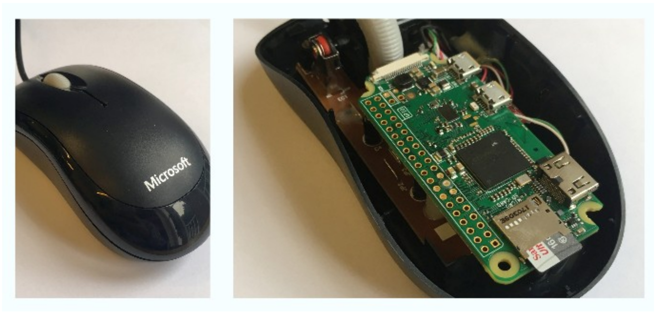
While doing through investigation, a maliciously altered USB device was discovered.

When connected to the system, the host system detected malicious mouse as a combination of fully a functional mouse and a keyboard.

By sending keystrokes, malicious device opend a powershell interface and sent keystrokes to type a malicious PowerShell script which in turn executed a covert chanel on communication stack.

Using the mouse's wireless interface, it created an out of band connection while bypassing the air-gap network security.

While being the perfect example of attacks and threats presented by IoT devices, mitigation for this attack, requires the network and devices to be continuously monitored for unintended behavior and alert operators and users to prevent this type of attacks.



Multiple USB devices can be used to perform this attack to make sure to reach the result. As example, a bunch of malicious devices can be dropped in the proximity of the infrastructure. When a user working at that organization plugs the same USB device curiously to the system accessible, he unintentionally compromises the same system.

While it may seem unlikely to belive that users will pick up devices and just use it, there is a study conducted by Elie Bursztein, the presentor of the talk, "Does Dropping USB devices really work?" seems to contradict that belief.[14]

In that study, he dropped 297 USB keys on the University of Illinois campus and analyzed result

USB keys varied in appearance. Some were just plain USB devices, some had physical keys attached to them, some with both keys and a note to return.

Some USBs had tempting notes attached to lure people to lure into picking up and plugging those USBs in their system. Notes were "confidencial" and "Final Exam Solutions".

Analyzing the effects and responses, results were astonishing.

45% of the all keys phoned home. Number can be fully subjective to the location where these USB keys were dropped. This number of 45% can be a lot higher and effective if the attack is performed on the criticatl infrastructure and provide a reverse shell access to the pc connected with usb and in turn complete infrastructure take over after privilege escalation attack.

6 Collaborative robots (COBOT)

6.1 Introduction to Cobots

According to Wikipedia, Collaborative Robot or a Cobot is generally understood as one that is intended to work alongside and / or directly interact with humans in a shared space.

In the initial phase of industrial development, Industrial machines were big and robust devices designed to work on specific taskes.

They needed fences and guards for safety and warning signs to represent dager presented by them.

While on the other hand, collaborative robots are complete antithesis of that, being compact, lightweight and dexterous.

Cobots equipped with new technologies, possess 7 degrees-of-freedom allowing better configuration and control over parts.

Being more flexible, they can perform more tasks and even do whatever a human can do to some extent.

Some cobots, while integrated with machine learning capavilities, can perform repetitive intelligently.

A good example of that would be GrowBot[15], which is developed to let non-experienced users work with cobots for repetitive tasks and to assuage the shortages of seasonal labor. GrowBot uses machine learning ability to learn to flexibly and skillfully perform the handling of seedling, herbs and other plants without user interaction.

6.3 Security Threats to Cobots.

The place and importance of cobots in industry as well as household while handling expensive equipments, increase the pressing need of sound security measurements for cobots.[16]

The most basic problem with these cobots is the ease of finding their presence and IP Address to guide the attack.

These robots use multicast DNS frames to advertise their presence over a network.

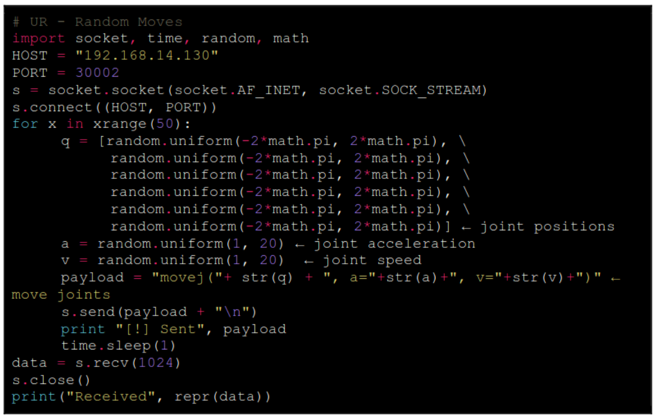
While actively listening for in the same network, one can identify and resolve their IP addresses even if no other name resolution services are present, due to their same unchanged default hostname.

As example, a NAO [17] robot can be identified with it's default hostname "nao. local" and a Universal Robot [18] with the hostname "ur. local".

There are also multiple authentication as well as authorization issues.

Generally, UR robot listening service run on port number 30002 and it is left unchanged and uniform to all UR robots.

Following exploit can be used to command random movement operations to a robot without any authentication with just it's IP address and default port number 30002.



```
# UR - Random Moves
import socket, time, random, math
HOST = "192.168.14.130"
PORT = 30002
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
for x in xrange(50):
      q = [random.uniform(-2*math.pi, 2*math.pi), \
           random.uniform(-2*math.pi, 2*math.pi), \
           random.uniform(-2*math.pi, 2*math.pi), \
           random.uniform(-2*math.pi, 2*math.pi), \
           random.uniform(-2*math.pi, 2*math.pi), \
           random.uniform(-2*math.pi, 2*math.pi)] ← joint positions
      a = random.uniform(1, 20) ← joint acceleration
      v = random.uniform(1, 20)  ← joint speed
      payload = "movej("+ str(q) + ", a="+str(a)+", v="+str(v)+")" ←
move joints
      s.send(payload + "\n")
      print "[!] Sent", payload
      time.sleep(1)
data = s.recv(1024)
s.close()
print("Received", repr(data))
```

The python exploit uses socket library to build a socket and to connect with UR robot service listening on IP address 192.168.14.130 and port 30002.

Then uses random and math libraries to generate random movement co-ordinates and then builts a payload along with joint acceleration and speed.

The exploit sends payload via the socket and listens for responses on the same communication medium.

While it may seem harmless to perform movement operations on cobots, it generally is not.

Attackers can use specific movements and controls to harm operators and cause injuries and alos wreak havoc on costly materials.

Some collaborative robot's protocols and/or softwares doesn't require users to have any authentication to have control over device.

For example, GR-001 from HPI Japan running V-Sido Operating System can be controlled using "V-Sido-Lite" without any authentication.

Same way, RoboPlus protocol of Robotis lacks authentication, so RoboPlus software can be downloaded by anyone to control motions for every Robotis product without any authentication by just using IP address and default TCP port 6501.

In addition to software vulnerabilities, some devices posses, design flaws, that causes visible and easily accessible USB, SD card, and Lan Ports.

Some devices even keep their debug ports like JTAG etc open and accessible, due to that attacker can extract as well as overwrite firmware.

7 Data Extraction Exploits in Arduino

7.1 Introduction to types of memory in Arduino

There are 3 important types of memory in an Arduino device[19].

- Flash or program Memory
- SRAM
- EEPROM

7.1.1 Flash

Program Image and any initialized data are stored in Flash memory of device.

Data in flash memory is read only, so it cannot be modified by executing code. To modify the data, it needs to be copied in SRAM

7.1.2 SRAM – Static Random-Access Memory.

SRAM, can be read and written from executing code

SRAM contains block of reserved space, named Static Data, it contains all global and static variables from program

Heap block is also provided for dynamically allocated data items in SRAM.

Finally, the stack block is provided for local

7.1.3 EEPROM

EEPROM is another type of non-volatile memory with read and write capabilities from executing program.

While it can only be read byte- by-byte, it is slower than SRAM and only 100,000 write cycles.

7.2 Extracting Data from Arduino

There are several possible ways to compile and upload firmware, program and bootloader to an Arduino device.

Simplest ways are,

1. Using Arduino GUI
2. Using Arduino CLI

Arduino provides software development kit with an Integrated Development Environment (IDE).

Arduino IDE takes source code, compiles it and links it with provided libraries as well as custom libraries relevant to the development board selected.

It also provides a repository for downloading toolkit for development boards not shipped with Arduino SDK.

While Arduino does not provide a way or interface to download uploaded file from Arduino device, in my research I found a way to download and analyze compiled hex file from Atmel's AVR microcontrollers. Using AVRDude one can program the Flash and EEPROM, and where supported by the serial programming protocols, it can program fuse and lock bits. In the process of extracting and cloning the AVR boards, I developed a python program, extractiot[20]

The python program takes input via -c or -w flags.

Input if provided -c indicates that user wants to download data from the connected device.

And input is provided as -w flag, it indicates that user wants to write to the board from files provided in the directory.

If directed to copy via command line argument, the program calls a function for copy.

This function presumes that code was compiled by Arduino compiler, and tries to extract data from all memories available.

The function creats a folder named copied in the program directory.

This python program has capabilities to extract data on many levels.

It can get data from FLASH, EFUSE, EEPROM, HFUSE and LFUSE.

Everything is stored in copied folder. Every file of the type ".hex"

Same way the fuction for write looks for files provided in the folder. If not specified, the function fetches files copied from the device attached before and writes it to the currently connected device, that is, makes a clone of device.

This program can be used to make copies on larger scale too, with a simple hacky code in default linux command shell, bash.

" python extractiot.py -c;

for i in `seq 1 100`; do \

sleep 6000;

python extractiot.py -w;

sleep 6000; \

done

"

This program assumes that you are trying to exract data from atmega 328p, if you want to perform on another type of arduino, you can do that by changing option from the program.

7.2.1 Consequences of Data extraction.

IoT devices are developed to operate and interact over network.

To properly commute over a network, the device needs to have credentials to properly authenticate over network.

In some cases, Iot device operates on Cloud platform.

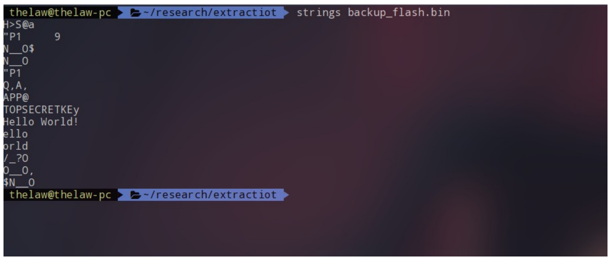
There are high possibilities that a number of IoT devices of same type are connecting to the same cloud, sharing and retriving information.

To enable IoT devices to do that, they need to have authentication keys for cloud platform.

Cloud platform can also contain database and other sensitive information.

After extracting the hex file, attacker can alayze it with "**Strings"** Unix utility or "**Binwalk",** the tool for searching a binary file.

As demonstrated in the image below, a simple command, "**Strings backup-flash.bin"** can give us all strings passed to the program while compiling the program uploaded to the device which also included sensitive secret key, **"TOPSECRETKEy"**



```
thelaw@thelaw-pc  ~/research/extractiot  strings backup_flash.bin
H>S@a
"P1     9
N__O$
N__O
"P1
Q,A,
APP@
TOPSECRETKEy
Hello World!
ello
orld
/_?O
O__O,
$N__O
thelaw@thelaw-pc  ~/research/extractiot
```

Attacker can use this key to access and command the cloud-based application and compromise all devices connected to it.

Consider another scenario, where an IoT based device is required to authenticate and access certain premises.

While attacker having momentatry access to that authentication IoT device can not enter premise, he can easlily make copies of that key and bypass the digital security.

So, ensuring that no secret keys are stored on IoT device as well as physical security of device is very important.

8 Case studies directly related to IoT security

- **Siberian Pipeline Explosion (1982)**

This is considered as the first cyber incdent involving safety-critical infrastructures which resulted in the explosion of a gas pipeline in Siberia 1982[21].

It is belived that a Trojan Horse malware was planted into the SCADA system incharge of regulating the gas pipeline.

By gaining access to the SCADA system, attackers changed the co-ordination of pumps, turbines and valves resulting into the internal pressure of pipeline far beyond the acceptable level, leading to an explosion with the power of 3 Kilotons of TNT

- **Slammer Worm Denial of Service (DOS) attack on Ohio Nuclear plant Network (2003)**

In first month of 2003, a worm named slammer, penetrated a private computer network at Ohio's nuclear power plant disabling a safety monitoring system for nearly 5 hours[22].

The slammer worm spread to the SCADA network by exploiting the vulnerabilities of MS-SQL present in the version used in systems.

- **Stuxnet Virus (2010)**

Stuxnet, a computer worm, discovered in 2010, was primarily written to target Iranian nuclear centrifuges[23].

Stuxnet was designed to specifically target Programmable Logic Controllers (PLCs), which allow for the automation of electromechanical processes like centrifuges for separating nuclear material.

Stuxnet used 4 Zero-day vulnerabilities to compromise Iraniagn centrifuges.

This attack resulted in infection of over 200,000 computers and physically degraded 1,000 machines.

References

[1] S. Haller, S. Karnouskos, and C. Schroth, "The Internet of Things in an Enterprise Context," in Future Internet – FIS 2008 Lecture Notes in Computer Science Vol. 5468, 2009, pp 14-28.

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

[3] things (iot): A vision, architectural elements, and futuredirections,”Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 92013.

[4] L. Columbus, “Roundup of internet of things forecasts and marketestimates,”https://www.forbes.com/sites/louiscolumbus/2016/11/27/roundup-of-internet-of-thi

[5] https://media.blackhat.com/bh-us-11/Radcliffe/BH_US_11_Radcliffe_Hacking_Medical_Devices_WP.pdf

[6] https://internetofthingsagenda.techtarget.com/definition/IoT-attack-surface

[7] https://www.owasp.org/index.php/IoT_Attack_Surface_Areas

[8] https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10

[9] https://www.owasp.org/images/3/3f/OWASP_Cloud_Top_10.pdf

[10] https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1152&context=adf

[11] http://www.iiisci.org/journal/CV$/sci/pdfs/ZA340MX17.pdf

[12] https://www.sciencedirect.com/science/article/pii/S0167404817301578

[13] https://www.infosecurityeurope.com/__novadocuments/480572?v=636632820679870000

[14] https://www.blackhat.com/docs/us-16/materials/us-16-Bursztein-Does-Dropping-USB-Drives-In-Parking-Lots-And-Other-Places-Really-Work.pdf

[15] https://www.growbot.eu/

[16] https://ioactive.com/pdfs/Hacking-Robots-Before-Skynet.pdf

[17] https://en.wikipedia.org/wiki/Nao_(robot)

[18] https://en.wikipedia.org/wiki/Universal_Robots

[19] https://www.arduino.cc/en/tutorial/memory

[20] https://github.com/malavvyas

[21] Daniela, T. 2011. Communication security in SCADA pipeline monitoring systems. Roedunet International Conference (RoEduNet), 2011 10th.

[22] European Conference on Information Warfare and Security: National Defence College, Helsinki, Finland, 1 - 2 June 2006. Academic Conferences Limited.

[23] Farwell, J.P. and Rohozinski, R. 2011. Stuxnet and the Future of Cyber War. Survival. 53, 1 (Feb. 2011), 23–40.

Malav Vyas Student at Gujarat Technological University,

Information Security Specialist – Infovys Inc.

Final Year Information Technology Engineering.

Speaker / Contributor– Null Ahmedabad

Volunteer – Owasp Seasides.