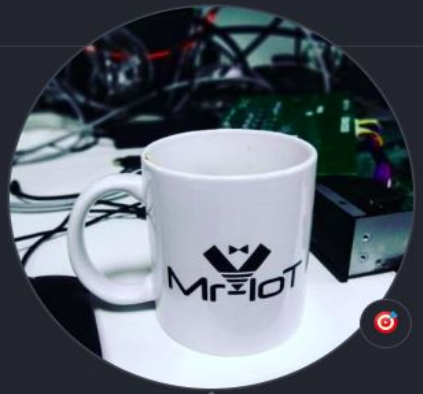
IoT Pentesting Simplified

Speed up the work

Agenda …!

1. IoT Attack Surfaces
2. IoT Pentesting vs Regular Pentesting
3. IoT Researchers life
4. Secrets to start testing different attack vectors
5. Mind Mapping your work
6. Automated tools which help us easy tasks
7. Standards and Conclusion

About me..



Change your avatar

Mr-IoT
V33RU

curl -sL https://tinyurl.com/mr-iot|sh
Donate :
https://www.buymeacoffee.com/v33ru

Edit profile

Overview | Repositories 51 | Projects 1 | Packages | Stars 70 | Sponsoring

V33RU / README.md

Hi all 👋,

I am well known as Mr-IoT rather than my original name, I'm the person who makes IoT Security Knowledge into opensource like IoT pentesting OS , and curated list and blogs as well. IoTSecurity101 Community successfully maintained since 4 years to till today in Reddit,Telegram and Discord too.

For contact : curl -sL https://tinyurl.com/mr-iot|sh

DISCORD | TELEGRAM | REDDIT

TWITTER | LINKEDIN | me WEBSITE

Buy me a coffee

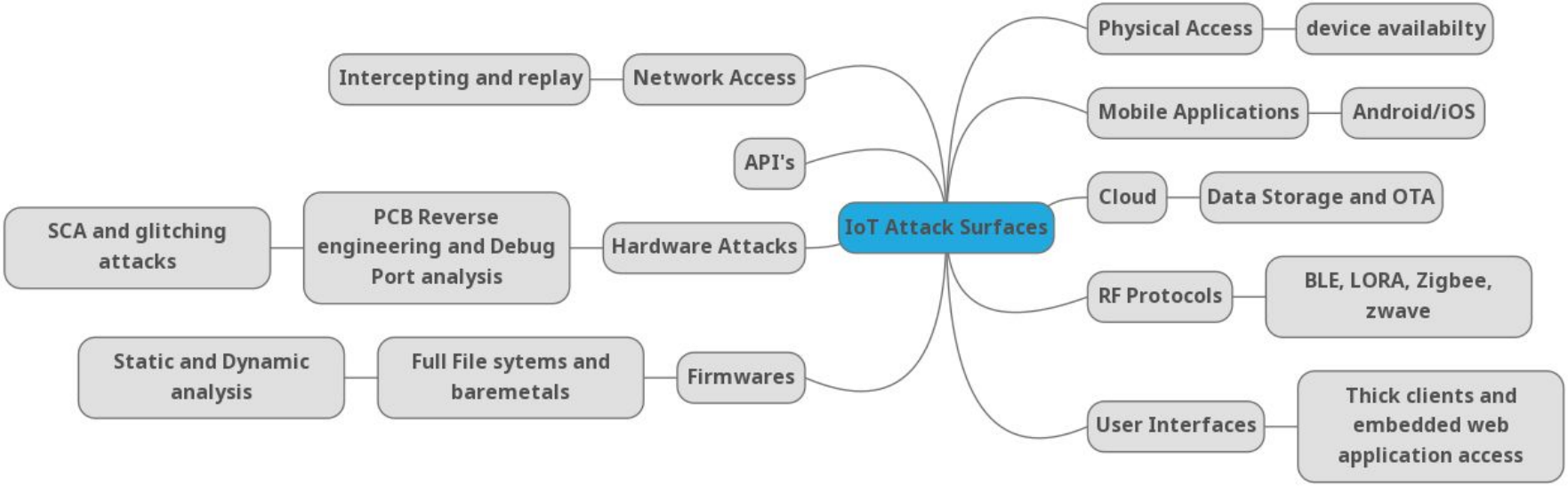
Profile views 4,549

Talks and Trainings

Others few work of mine and a little about me..

1. Created IoT-PT OSv1 , and v2 and v3 coming soon
2. Made a blogs and resources for problem solving of current trend
3. Check my github very clearly all your questions have already answered there
4. IoT Security 101 - Telegram , Discord , Reddit actively working since 4 year

IoT Attack Surfaces



IoT Pentesting vs Regular Pentesting

Regular Pentesting

1. Mostly follow by given target reconnaissance (generic)
2. Mostly depends technology implementation attacks

E.g : SQL,GRAPHQL,MYSQL etc

3. If you know technology and input locations and tricks mostly solve your problem

IoT Pentesting

1. Will start understanding the functionality then recon
2. Most of the IoT Device developed under Linux / RTOS /SELinux

E.g: OS Cmd injection,file path manipulation

3. Understand device as much as you can , like testing device standalone vs with fully configured

IoT Researchers life


TOL-12042
dtify


MONGOLIA
CHINA
PHILIPPINES
Tigard
CC Debugger
MPLAB
PICkit 4
MICROCHIP
PICkit
POWER
ACTIVE
STATUS
USB TO TTL
USB
offbeat

Secrets to start pentesting different attack vectors

IoT Attack vectors are bit more exhausted.

1. Map the service based vulnerabilities as per technologies
2. Buy the relevant and supported device to pentest IoT Protocols
3. Check deprecated tools and look for tools actively support is there currently or not
4. Understand network level reverse engineering / Replay concepts
5. Fuzzing will help you to find cool bugs in IoT devices
6. Work on daemon services inside firmware
7. Breaking Into hardware

Map the service based vulnerabilities as per technologies

IoT Technology	Common Service-Based vulnerabilities
Wi-Fi	Attacks mostly like Client AP attacks and Access Points
Bluetooth	Authentication and DOS , MiTM. Chipset based Vulnerabilities and Version Based Vulnerabilities
Zigbee	Insecure key storage , plaintext key NWK, DOS , MiTM, Selective Jamming Attacks
Hardware	Check for debug ports and possible simple attacks
USB	Depends on device , ADB over USB, Keystroke injections, USB Rubber ducky attacks
Firmware	Static and Dynamic analysis, busybox vulnerabilities ,3rd party libraries version based bugs

Buy the relevant and supported device to pentest IoT Product technologies

V33RU Delete IoT Security Lab Setup - Independent Researchers - Sheet1.pdf a2ac342 on Feb 10 **28** commits

Enterprise	Update setup.md	3 months ago
Independent-Researchers	Update setup.md	3 months ago
README.md	Update README.md	3 months ago

README.md

IoT-Security/Development-Lab-Setup

- Enterprise : For companies internally building their labs for security team
- Independent : For independent researchers and enthusiastic people purpose

https://github.com/IoT-PTv/IoT-Lab-Setup

Check for tools or scripts

This branch is 15 commits ahead of devttys0:master. Contribute Sync fork

V33RU Update README.md 5777537 on Sep 2, 2021 17 commits

LICENSE	Initial commit	8 years ago
README.md	Update README.md	2 years ago
baudrate.py	Update baudrate.py	2 years ago
values.txt	Update values.txt	2 years ago

README.md

baudrate values added more

By Craig Heffner, http://www.devttys0.com

```
#sudo python3 baudrate.py -h
-p <serial port>        Specify the serial port to use [/dev/ttyUSB0]
-t <seconds>            Set the timeout period used when switching baudrates in auto detect mode [5]
-c <num>                Set the minimum ASCII character threshold used during auto detect mode [25]
```

https://github.com/V33RU/baudrate

Understand network level replay/reverse engineering concepts

1. Understand Concepts of port mirroring

2. Capture action request of replay with python socket program

3. Play with tcpdump , taskstat and netstat

Fuzzing will help you to find cool bugs in IoT devices

Use tools like AFL++ and Radamsa and Boofuzz actively help you in IoT Devices Pentesting

1. Radamsa
2. Boofuzz
   a. Network (FTP , HTTP)
   b. BACNET
3. AFL ++

Fuzzing for Fun and Profit

https://www.exploit-db.com/papers/12965

demo

Some Fuzzing sources

Fuzzing Things

- OWASP Fuzzing Info
- Fuzzing_ICS_protocols
- Fuzzowski - the Network Protocol Fuzzer that we will want to use
- Fuzz Testing of Application Reliability
- FIRM-AFL : High-Throughput Greybox Fuzzing of IoT Firmware via Augmented Process Emulation
- Snipuzz : Black-box Fuzzing of IoT Firmware via Message Snippet Inference
- [fuzzing-iot-binaries] - part1 / part2

https://github.com/V33RU/IoTSecurity101#Fuzzing-Things

Work on daemon services inside firmware

- Httpd,lighthttpd,ftpd, and many other daemon services
- Runtime analysis best on these service based binaries

Emulation will help you find crazy bugs

- Qemu deboostrap
- Qiling
- Qemu
- FAT
- Azeria labs VM

Breaking Into hardware

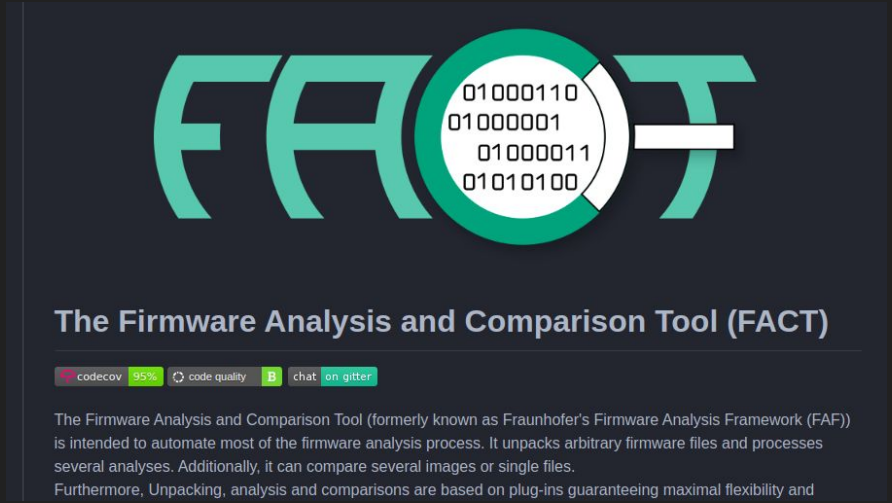
- Analyze the PCB for debug ports , power reboot buttons
- Visual analysis for ROM chips to get datasheets
- Extracting data from EEPROM and EMMC

Mind Mapping your work

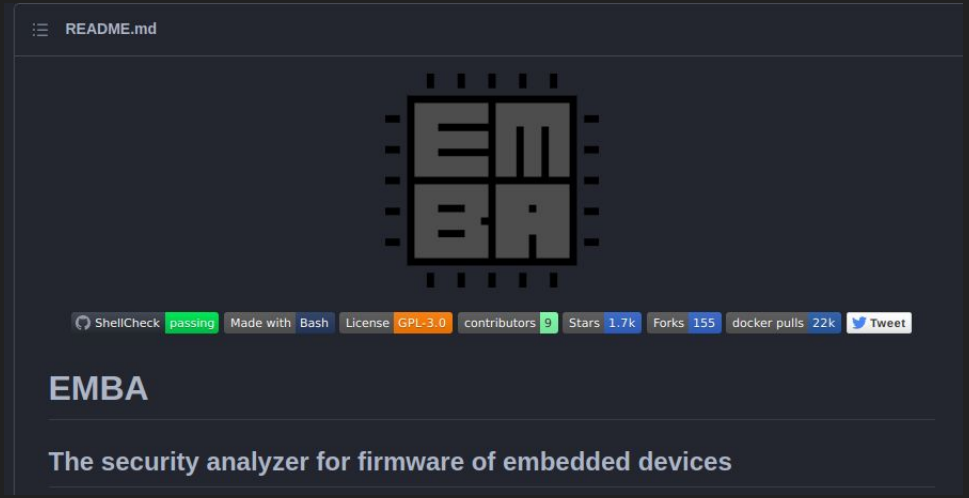
1. MindMaps helps everywhere - choose any software from internet
2. Get all datasheets of device make map each technology
3. Attack vectors always depends version and stack of the protocols and behaviour of it

Automated tools which help us easy tasks

FACT



EMBA



Conclusion

Nothing is secure

And

Learning never ends

Q&A