Binary Exploitation Lab

Michael Schwarz

June 18, 2017

IAIK Spring School 2017

Motivation

IoT Bugs

Top 10 IoT Bugs

Top 10 IoT Bugs

1. Insecure Web Interface

Default usernames and passwords

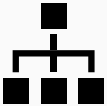
Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication

Weak passwords

Top 10 IoT Bugs

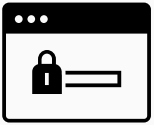
1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services



Unnecessary ports open

Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption



SSL/TLS not available

Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns

Collected information not properly protected

Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns
6. Insecure Cloud Interface

Interfaces with security vulnerabilities

Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns
6. Insecure Cloud Interface
7. Insecure Mobile Interface

No account lockout mechanisms

Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns
6. Insecure Cloud Interface
7. Insecure Mobile Interface
8. Insufficient Security Configurability

Encryption is not available

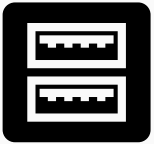
Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns
6. Insecure Cloud Interface
7. Insecure Mobile Interface
8. Insufficient Security Configurability
9. Insecure Software/Firmware

Updates are not signed

Top 10 IoT Bugs

1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns
6. Insecure Cloud Interface
7. Insecure Mobile Interface
8. Insufficient Security Configurability
9. Insecure Software/Firmware
10. Poor Physical Security



Unnecessary external ports like USB

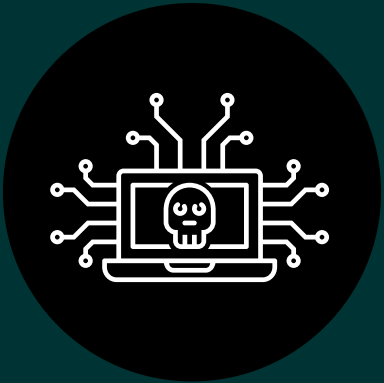
Summary

The 90s called...

Summary

The 90s called...

...they want their bugs back!



Let’s try it!

Infrastructure

There are 6 different hacklets

Name	Difficulty	Type	IP	Access Point
Admin Panel	□□□□□		192.168.3.235	Expl0it

Memory corruption

Puzzling

Binary/Reversing

Python

Infrastructure

There are 6 different hacklets

Name	Difficulty	Type	IP	Access Point
Admin Panel	□□□□□		192.168.3.235	Expl0it
Secure Router	■□□□□		192.168.3.101	M0reExpl0it

Memory corruption

Puzzling

Binary/Reversing

Python

Infrastructure

There are 6 different hacklets

Name	Difficulty	Type	IP	Access Point
Admin Panel	☐☐☐☐☐		192.168.3.235	Expl0it
Secure Router	■☐☐☐☐		192.168.3.101	M0reExpl0it
Debug Shell I	■■☐☐☐		192.168.3.100	M0reExpl0it

Memory corruption

Puzzling

Binary/Reversing

Python

Infrastructure

There are 6 different hacklets

Name	Difficulty	Type	IP	Access Point
Admin Panel	□□□□□		192.168.3.235	Expl0it
Secure Router	■□□□□		192.168.3.101	M0reExpl0it
Debug Shell I	■■□□□		192.168.3.100	M0reExpl0it
Debug Shell II	■■■□□		192.168.3.239	Expl0it

Memory corruption

Puzzling

Binary/Reversing

Python

Infrastructure

There are 6 different hacklets

Name	Difficulty	Type	IP	Access Point
Admin Panel	☐☐☐☐☐		192.168.3.235	Expl0it
Secure Router	■☐☐☐☐		192.168.3.101	M0reExpl0it
Debug Shell I	■■☐☐☐		192.168.3.100	M0reExpl0it
Debug Shell II	■■■☐☐		192.168.3.239	Expl0it
Time Server	■■■■☐		192.168.3.102	M0reExpl0it

Memory corruption

Puzzling

Binary/Reversing

Python

Infrastructure

There are 6 different hacklets

Name	Difficulty	Type	IP	Access Point
Admin Panel	☐☐☐☐☐		192.168.3.235	Expl0it
Secure Router	■☐☐☐☐		192.168.3.101	M0reExpl0it
Debug Shell I	■■☐☐☐		192.168.3.100	M0reExpl0it
Debug Shell II	■■■☐☐		192.168.3.239	Expl0it
Time Server	■■■■☐		192.168.3.102	M0reExpl0it
Power Plant	■■■■■		192.168.3.241	Expl0it

Memory corruption

Puzzling

Binary/Reversing

Python

Infrastructure

- Every hacklet has a hidden flag

Infrastructure

- Every hacklet has a hidden flag
- Flags are usually in a text file `flag.txt` on the device

Infrastructure

- Every hacklet has a hidden flag
- Flags are usually in a text file `flag.txt` on the device
- A flag looks like `{TH1S_IS_A_FL4G!}`

Infrastructure

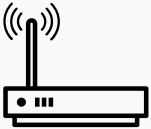
- Every hacklet has a hidden flag
- Flags are usually in a text file `flag.txt` on the device
- A flag looks like `{TH1S_IS_A_FL4G!}`
- Goal is to get the flag and submit it to the highscore list

Infrastructure

- Every hacklet has a hidden flag
- Flags are usually in a text file `flag.txt` on the device
- A flag looks like `{TH1S_IS_A_FL4G!}`
- Goal is to get the flag and submit it to the highscore list
- Highscore can be found here: `http://192.168.3.191/hs` (Expl0it)

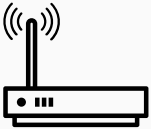
Infrastructure

- All IoT devices/hacklets are in an internal network



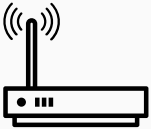
Infrastructure

- All IoT devices/hacklets are in an internal network
- They are not connected to the internet



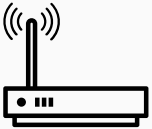
Infrastructure

- All IoT devices/hacklets are in an internal network
- They are not connected to the internet
- Connect to the routers `Expl0it` or `M0reExpl0it` to start hacking



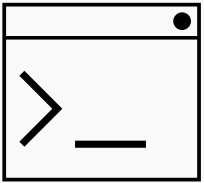
Infrastructure

- All IoT devices/hacklets are in an internal network
- They are not connected to the internet
- Connect to the routers `Expl0it` or `M0reExpl0it` to start hacking
- The password is `iotiotiot` (3x "iot")



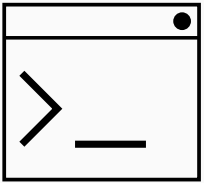
How to Connect

- Hacklets are accessible over the network



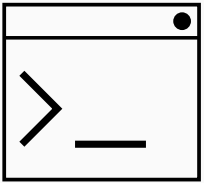
How to Connect

- Hacklets are accessible over the network
- Every hacklet has a text interface on port 8888



How to Connect

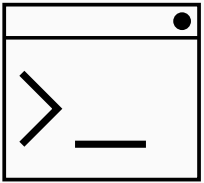
- Hacklets are accessible over the network
- Every hacklet has a text interface on port 8888
- You can connect using any telnet-like program:
  - PuTTY
  - Terminal, `netcat`, `telnet`
  - `netcat`, `telnet`



How to Connect

- Hacklets are accessible over the network
- Every hacklet has a text interface on port 8888
- You can connect using any telnet-like program:
  - PuTTY
  - Terminal, `netcat`, `telnet`
  - `netcat`, `telnet`
- For example on Linux/Mac in the shell:

```
telnet 192.168.3.235 8888
```



How to Start

- Use your own computer or our provided Linux VM (on USB or from `http://192.168.3.191` (Expl0it))

How to Start

- Use your own computer or our provided Linux VM (on USB or from `http://192.168.3.191` (Expl0it))
- Download a hacklet to analyze it: `http://192.168.3.191` (Expl0it)

How to Start

- Use your own computer or our provided Linux VM (on USB or from `http://192.168.3.191` (Expl0it))
- Download a hacklet to analyze it: `http://192.168.3.191` (Expl0it)
- Connect to the hacklet

How to Start

- Use your own computer or our provided Linux VM (on USB or from `http://192.168.3.191` (Expl0it))
- Download a hacklet to analyze it: `http://192.168.3.191` (Expl0it)
- Connect to the hacklet
- Remember today's talk of Ahmad Sadeghi
  - What happens if I enter a lot of text?
  - Does it crash? Can I exploit that?
  - Is there maybe a different interface?

How to Start

- Run `strings` on the binary to extract all texts

How to Start

- Run `strings` on the binary to extract all texts
- Use a port scanner to check if there is an alternative interface (SSH is not exploitable!)

How to Start

- Run `strings` on the binary to extract all texts
- Use a port scanner to check if there is an alternative interface (SSH is not exploitable!)
- `arm-linux-gnueabi-objdump` could be useful
  - `arm-linux-gnueabi-objdump -d <hacklet>` to disassemble
  - `arm-linux-gnueabi-objdump -x <hacklet>` to see headers and symbols

How to Start

- Run `strings` on the binary to extract all texts
- Use a port scanner to check if there is an alternative interface (SSH is not exploitable!)
- `arm-linux-gnueabi-objdump` could be useful
  - `arm-linux-gnueabi-objdump -d <hacklet>` to disassemble
  - `arm-linux-gnueabi-objdump -x <hacklet>` to see headers and symbols
- Watch out for dangerous functions (e.g. `strcpy`)

Questions?

Bonus: Running hacklets locally

It can be useful to run hacklets locally

- Install `qemu`
- Download Raspbian Image + Kernel + Starter from `http://192.168.3.191`
- Execute `chmod +x ./run.sh` and run `./run.sh`
- Remote shell to QEMU: `ssh localhost 2222`
- Connect to hacklet: `netcat localhost 8888`