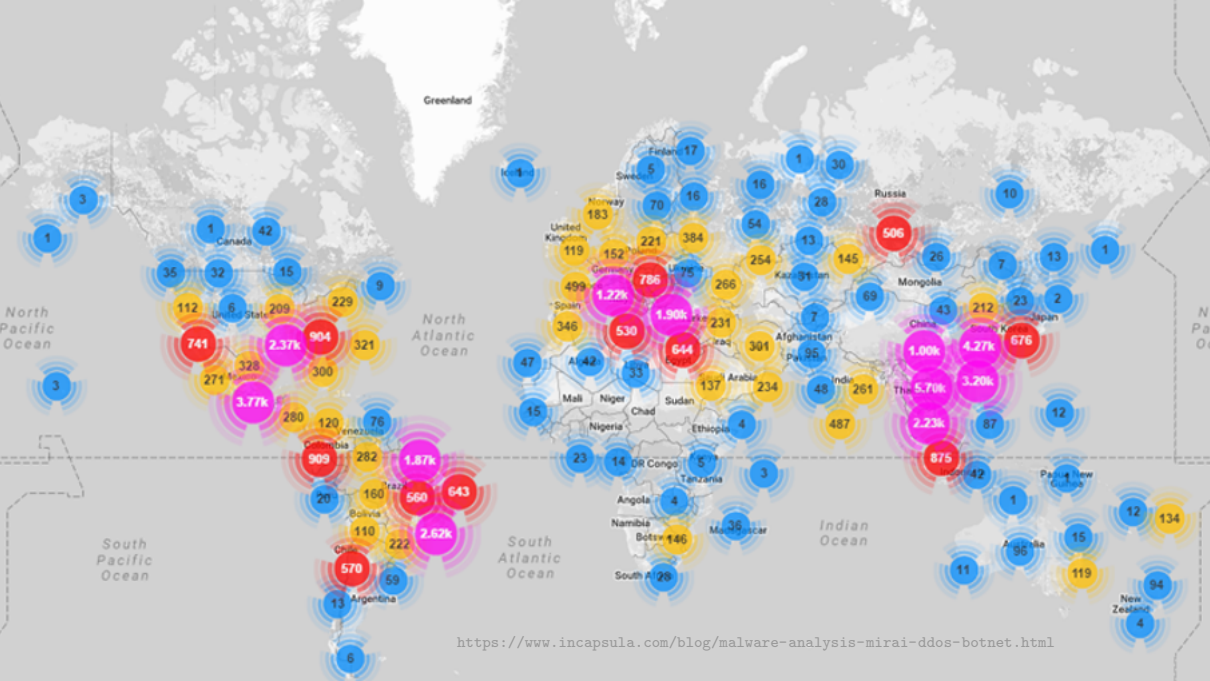
# Runtime Security Lab

**Michael Schwarz**

September 3, 2018

Security Week Graz 2018




https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html

# Large IoT Incidents



- September 21, 2016
  > **600 Gbps** on Brian Krebs (security researcher) website (Mirai botnet)
- September 30, 2016
  Mirai source code published
- October 21, 2016
  ~**1 Tbps** on DNS provider Dyn
- November 26, 2016
  > **900 000** routers of Deutsche Telekom attacked and offline
- February, 2018
  > **1.35 Tbps** attack on GitHub


BUGS
SPACE RANGER
LIGHTYEAR
BUGS EVERYWHERE

# Top 10 IoT Bugs

HELPING SECURE THE INTERNET OF THINGS WITH THE

OWASP

INTERNET OF THINGS TOP 10

VULNERABILITY CATEGORIES

# Top 10 IoT Bugs

1. Insecure Web Interface

Default usernames and passwords

# Top 10 IoT Bugs

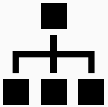


1. Insecure Web Interface
2. Insufficient Authentication

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



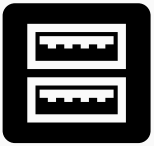
1. Insecure Web Interface
2. Insufficient Authentication
3. Insecure Network Services
4. Lack of Transport Encryption
5. Privacy Concerns
6. Insecure Cloud Interface
7. Insecure Mobile Interface
8. Insufficient Security Configurability
9. Insecure Software/Firmware

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

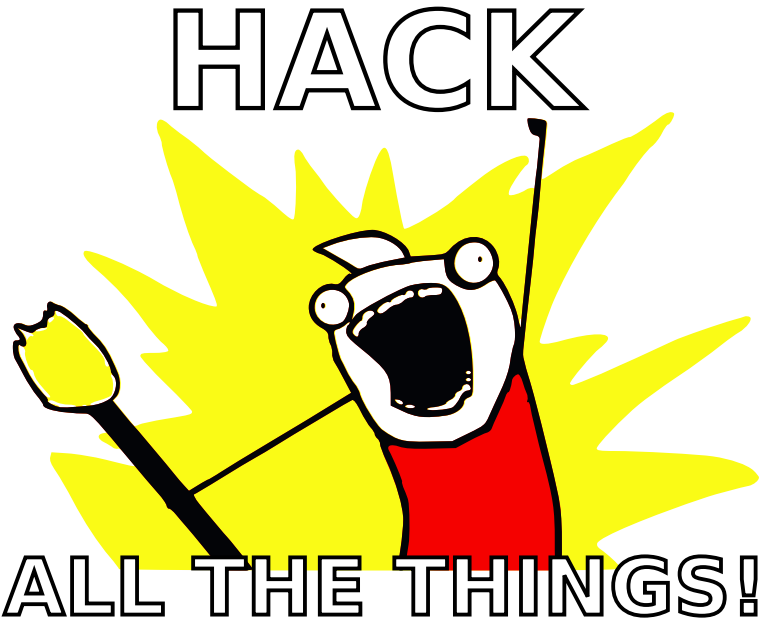
# Summary

The 90s called...

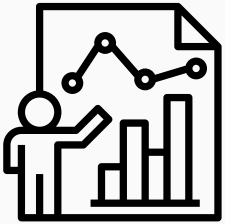
# Summary

The 90s called...

...they want their bugs back!

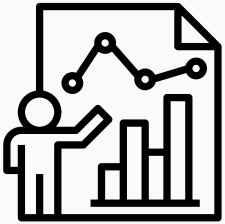

HACK
ALL THE THINGS!

# Infrastructure



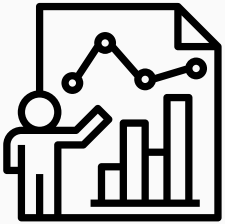
- There are 15 challenges

# Infrastructure



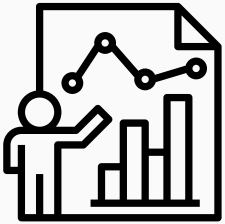
- There are 15 challenges
- Different difficulties (the more points, the harder)

# Infrastructure



- There are 15 challenges
- Different difficulties (the more points, the harder)
- 4 different categories

# Infrastructure



- There are 15 challenges
- Different difficulties (the more points, the harder)
- 4 different categories
- Play on your own or as team

# https://ctf.attacking.systems


## pwn

| Warm Up | Secure Webserver | Debug Shell I | Debug Shell II |
|---|---|---|---|
| 10 | 20 | 50 | 80 |

| diJIT Integrator | diJIT Plot |
|---|---|
| 100 | 150 |

## misc

| RTFM | Weird Architecture | Secure Router | Let's Play a Game |
|---|---|---|---|
| 5 | 20 | 50 | 50 |

## forensics

| Printer Update | Random Blob |
|---|---|
| 50 | 50 |

## crypto

| Flight Radar | Lightweight Crypto | Encrypted Blob |
|---|---|---|
| 30 | 30 | 50 |

# Infrastructure

- Capture-the-flag (CTF) style

# Infrastructure

- Capture-the-flag (CTF) style
- Every challenge has a hidden flag

# Infrastructure

- Capture-the-flag (CTF) style
- Every challenge has a hidden flag
- Flags are usually in a text file `flag.txt` on the device

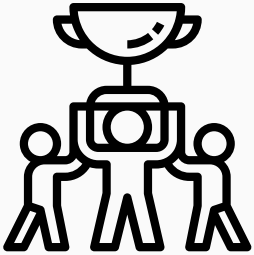
# Infrastructure

- Capture-the-flag (CTF) style
- Every challenge has a hidden flag
- Flags are usually in a text file `flag.txt` on the device
- A flag looks like `{A_S4MPL3_FL4G!}`

# Infrastructure

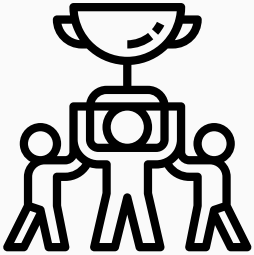
- Capture-the-flag (CTF) style
- Every challenge has a hidden flag
- Flags are usually in a text file `flag.txt` on the device
- A flag looks like `{A_S4MPL3_FL4G!}`
- Goal is to get the flag and submit it to the CTF system

# Timeline



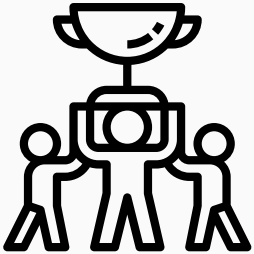
- CTF runs until Friday, 3:00pm

# Timeline



- CTF runs until Friday, 3:00pm
- Last-minute questions from 2:00pm to 3:00pm

# Timeline



- CTF runs until Friday, 3:00pm
- Last-minute questions from 2:00pm to 3:00pm
- Best player/team gets a price

# How to Start

- Use your own computer or our provided Linux VM (on USB or from `https://ctf.attacking.systems/res`)

# How to Start

- Use your own computer or our provided Linux VM (on USB or from `https://ctf.attacking.systems/res`)
- Create or join a team in the CTF system: `https://ctf.attacking.systems`

# How to Start

- Use your own computer or our provided Linux VM (on USB or from `https://ctf.attacking.systems/res`)
- Create or join a team in the CTF system: `https://ctf.attacking.systems`
- Choose a hacklet, read the description, and download it

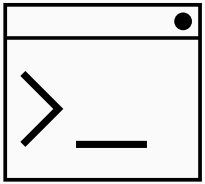
# How to Start

- Use your own computer or our provided Linux VM (on USB or from `https://ctf.attacking.systems/res`)
- Create or join a team in the CTF system: `https://ctf.attacking.systems`
- Choose a hacklet, read the description, and download it
- Solve the hacklet by connecting to the hacklet

# How to Connect



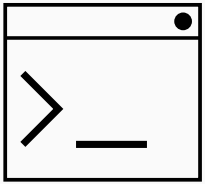
- Hacklets are accessible over the network



# How to Connect

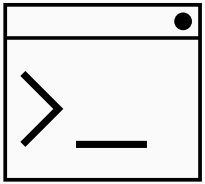


- Hacklets are accessible over the network
- Every hacklet has a text interface on a specific port



# How to Connect

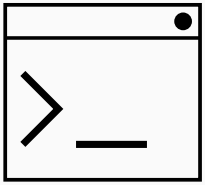
- Hacklets are accessible over the network
- Every hacklet has a text interface on a specific port
- You can connect using any telnet-like program:
  - PuTTY
  - Terminal, `netcat`, `telnet`
  - `netcat`, `telnet`



# How to Connect

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

```
netcat hacklets2.attacking.systems 8000
```



# The Categories



There are 4 categories: pwn (  ), forensics (  ), crypto (  ), misc ()

# The Categories



There are 4 categories: pwn (  ), forensics (  ), crypto (  ), misc ()

-  Vulnerable binaries which you have to exploit

# The Categories



There are 4 categories: pwn (), forensics (), crypto (), misc ()

- Vulnerable binaries which you have to exploit
- Basically finding/reconstructing hidden/deleted stuff

# The Categories



There are 4 categories: pwn (  ), forensics (  ), crypto (  ), misc (  )

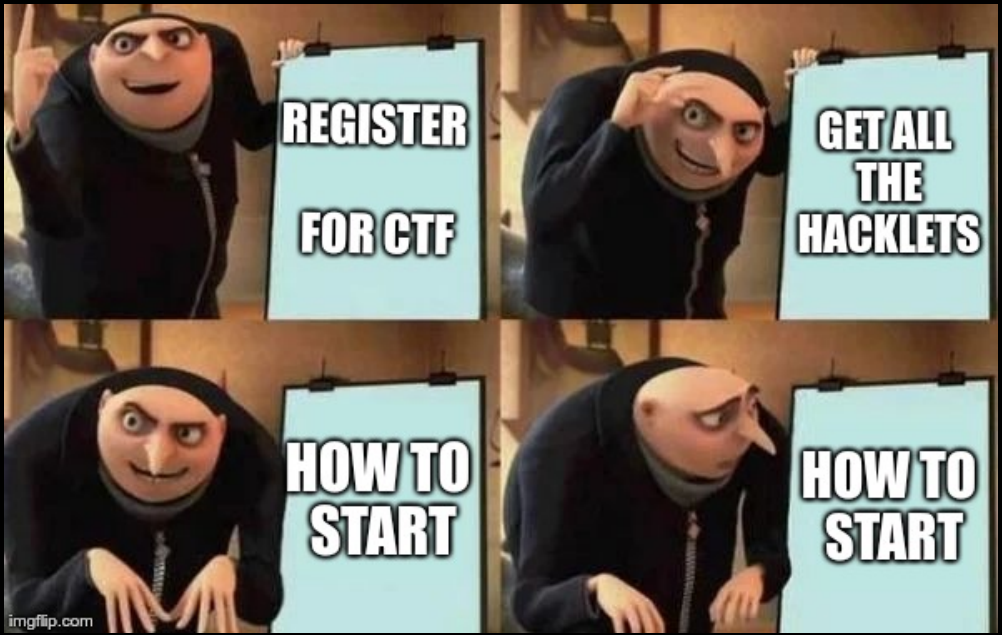
-  Vulnerable binaries which you have to exploit
-  Basically finding/reconstructing hidden/deleted stuff
-  (Bad) Cryptography you have to break

# The Categories



There are 4 categories: pwn (  ), forensics (  ), crypto (  ), misc (  )

-  Vulnerable binaries which you have to exploit
-  Basically finding/reconstructing hidden/deleted stuff
-  (Bad) Cryptography you have to break
-  Random and fun hacklets which do not fit into any category (often no programming required)


REGISTER
FOR CTF
GET ALL
THE
HACKLETS
HOW TO
START
HOW TO
START
imgflip.com

# How to Start

- Download the hacklet

# How to Start

- Download the hacklet
- Identify the type of file
  - Executable? For which platform?
  - Data? Which program can open it?
  - Unknown?

# How to Start

- Download the hacklet
- Identify the type of file
  - Executable? For which platform?
  - Data? Which program can open it?
  - Unknown?
- Useful Linux tool: `file` – determines the file type

# Unknown Files



- Maybe file is some archive...

# Unknown Files



- Maybe file is some archive...
- ...or contains multiple files

# Unknown Files



- Maybe file is some archive...
- ...or contains multiple files
- Binwalk Firmware Analysis Tool
  `https://github.com/ReFirmLabs/binwalk`

# Unknown Files

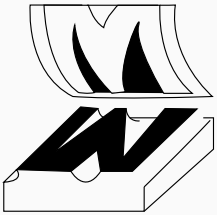


- Maybe file is some archive...
- ...or contains multiple files
- Binwalk Firmware Analysis Tool
  `https://github.com/ReFirmLabs/binwalk`
- Can also extract files

# Readable Information

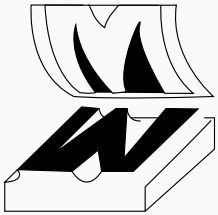


- Run `strings` on the file to extract all texts



# Readable Information

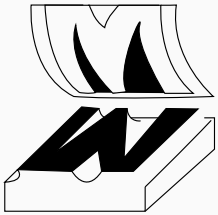
- Run `strings` on the file to extract all texts
- For binaries: see all functions/variables (*i.e.*, symbols)
  - x86: `objdump -x <hacklet>`
  - ARM: `arm-linux-gnueabi-objdump -x <hacklet>`



# Readable Information



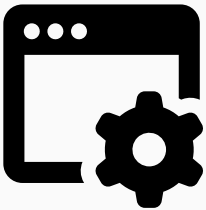
- Run `strings` on the file to extract all texts
- For binaries: see all functions/variables (*i.e.*, symbols)
  - x86: `objdump -x <hacklet>`
  - ARM: `arm-linux-gnueabi-objdump -x <hacklet>`
- Watch out for function names containing `flag`



# Binaries

- Try to run the binary
  - x86: no requirements
  - ARM: requires

```
libc6-dev-armhf-cross qemu-system-arm qemu-user
```



# Binaries



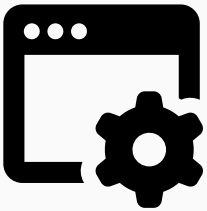
- Try to run the binary
  - x86: no requirements
  - ARM: requires

    ```
    libc6-dev-armhf-cross qemu-system-arm qemu-user
    ```
- Then simply execute

  ```
  qemu-arm -L /usr/arm-linux-gnueabihf ./hacklet
  ```

  or for ARMv8

  ```
  qemu-aarch64 -L /usr/aarch64-linux-gnu ./hacklet
  ```



# Binaries

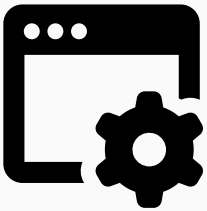
- Try to run the binary
  - x86: no requirements
  - ARM: requires

    ```
    libc6-dev-armhf-cross qemu-system-arm qemu-user
    ```
- Then simply execute

  ```
  qemu-arm -L /usr/arm-linux-gnueabihf ./hacklet
  ```

  or for ARMv8

  ```
  qemu-aarch64 -L /usr/aarch64-linux-gnu ./hacklet
  ```
- More details: `https://ctf.attacking.systems/res`



# Binaries

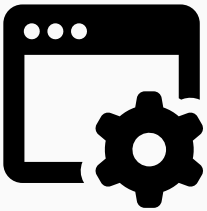
- Try to run the binary
  - x86: no requirements
  - ARM: requires

    ```
    libc6-dev-armhf-cross qemu-system-arm qemu-user
    ```
- Then simply execute

  ```
  qemu-arm -L /usr/arm-linux-gnueabihf ./hacklet
  ```

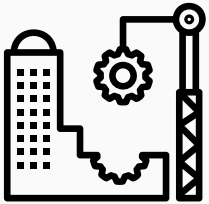
  or for ARMv8

  ```
  qemu-aarch64 -L /usr/aarch64-linux-gnu ./hacklet
  ```
- More details: `https://ctf.attacking.systems/res`
- Use a port scanner to check for alternative interface (SSH is not exploitable!)



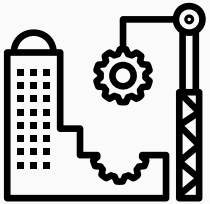
# Reverse Engineering

- Command-line disassembler
  - x86: `objdump -d <hacklet>`
  - ARM: `arm-linux-gnueabi-objdump -d <hacklet>`
  - All platforms: radare2



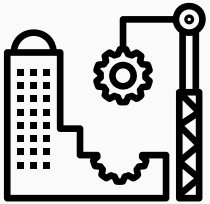
# Reverse Engineering

- Command-line disassembler
  - x86: `objdump -d <hacklet>`
  - ARM: `arm-linux-gnueabi-objdump -d <hacklet>`
  - All platforms: radare2
- Watch out for dangerous functions (e.g. `strcpy`, `gets`)



# Reverse Engineering

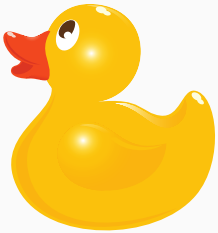
- Command-line disassembler
  - x86: `objdump -d <hacklet>`
  - ARM: `arm-linux-gnueabi-objdump -d <hacklet>`
  - All platforms: radare2
- Watch out for dangerous functions (e.g. `strcpy`, `gets`)
- GUI disassembler: cutter
  `https://github.com/radareorg/cutter`



# Rubberduck Debugging



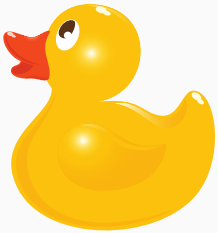
- It helps to explain what you see



# Rubberduck Debugging



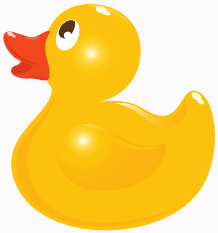
- It helps to explain what you see
- Talking about the problem can be the first step



# Rubberduck Debugging

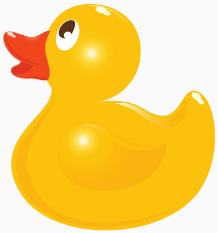


- It helps to explain what you see
- Talking about the problem can be the first step
- Usually we talk to humans



# Rubberduck Debugging

- It helps to explain what you see
- Talking about the problem can be the first step
- Usually we talk to humans
- If none available/interested: use a rubber duck!



# What next?



- Let's start with the challenges!

# What next?



- Let's start with the challenges!
- `https://ctf.attacking.systems`

# What next?

- Let's start with the challenges!
- `https://ctf.attacking.systems`
- If you are unsure, there is a walkthrough of one hacklet:
  `https://ctf.attacking.systems/res`

# What next?

- Let's start with the challenges!
- `https://ctf.attacking.systems`
- If you are unsure, there is a walkthrough of one hacklet:
  `https://ctf.attacking.systems/res`
- Additionally: Slides from our lecture "Security Aspects in Software Development"
  `https://teaching.iaik.tugraz.at/sase/slides`

# A Challenge a Day Keeps the Boredom Away

# Questions?