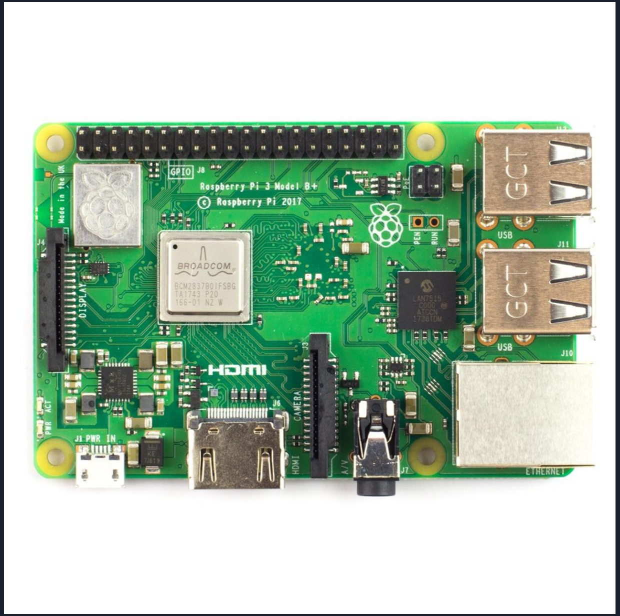
Security vulnerabilities in the Raspberry Pi

Brief overview of what a Raspberry Pi is

- SBC (Single Board Computer)
- Size of a credit card
- Microprocessor
- Can be loaded with OS
- Easy to program



Motivation

- Provide insight on security vulnerabilities
- Teach how to secure Raspberry Pis
- Teach the differences between Windows Machines and Raspberry pi’s

Paper #1: Security Vulnerabilities in Raspberry Pi–Analysis of the System Weaknesses

- Main goal was to outline the components for hardware and software analysis
- After installing an OS leads to default configuration
- Users need to understand the risks of not securing their system

Software analysis

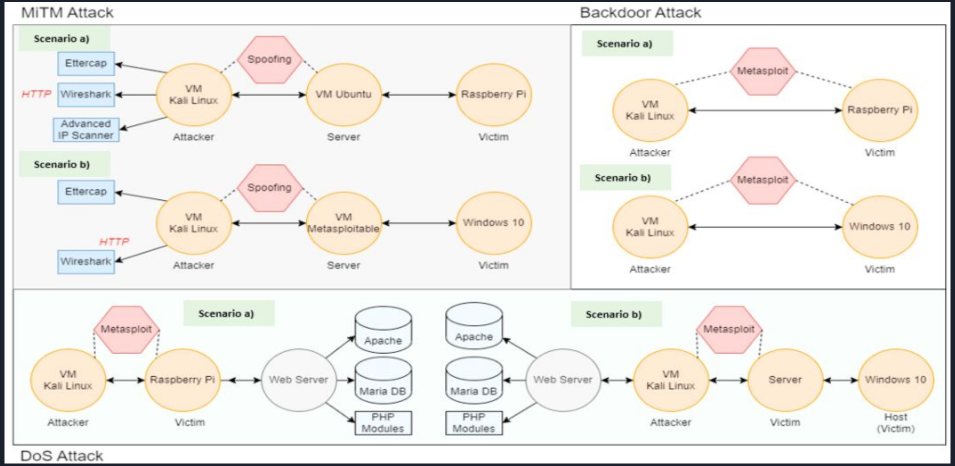
- Raspbian
  - Default username/password can only be changed after installation
  - Debian-4þdeb7u2 1.0.1e 11 Feb 2013
    - Some immunity to UserRoaming and Heartbleed bugs
      - UserRoaming - Stdio buffers are cleaned after usage
      - Heartbleed - Bug fix
- Windows 10 IoT
  - Same user/passwd config as raspbian
  - Windows Device Portal has authentication done through clear text
  - ARP-spoofing and MITM
- Open-ELEC/Libre-ELEC
  - Default user/passwd cannot be changed
  - No 2FA for HTTP and Samba config
- Ubuntu
- RiscOS

Hardware analysis

- USB
  - Powered USB hub required for devices using more than 500mA
  - Backfeeding: drawing power from incoming current from a USB port
  - No USB protection
- Overclocking
  - Can be safely performed
  - Bad config sets a bit inside the SoC (system on chip)
- GPIO
  - SoC can be destroyed if output voltage > 5V or output current > 2.5A
- Real time clock absence
  - Date/time not stored internally after being powered off
  - Critical for certificate validation, cryptography
  - Fetches the date/time from a Network Time Server
- Xenon flash
  - RPI 2 reboots if being pointed at with a laser.

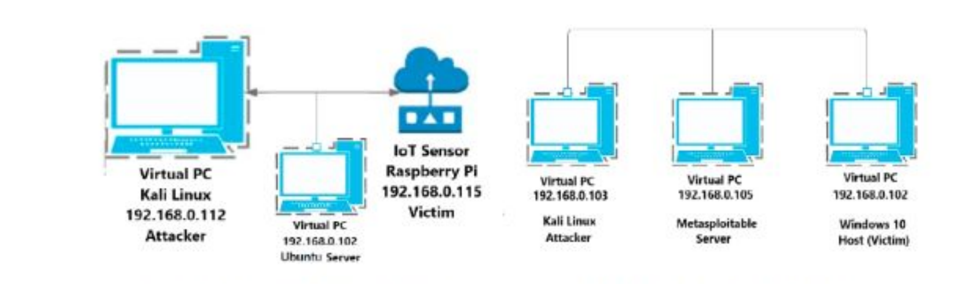
Paper #2: Security Vulnerability Analysis for IoT Devices Raspberry Pi using PENTEST

- Introduction
  - Done in both Windows and Raspberry Pi
  - Used Kali Linux as attacker
  - Same techniques used
- Material Used
  - Oracle VM
  - Raspberry Pi loaded with Raspbian
  - VNC Viewer
- Attacks
  - Man in the middle
  - Backdoor
  - DoS (Denial of services)



Results-Man in the Middle (MITM)

- Common Factors
  - Used HTTP
  - Wireshark
  - ARP Poisoning
  - Ettercap
- Raspberry Pi
  - Advanced IP Scanner
  - Ubuntu Server (Server)
  - Successful
- Windows 10
  - Metasplotiable2 (Server)
  - Successful

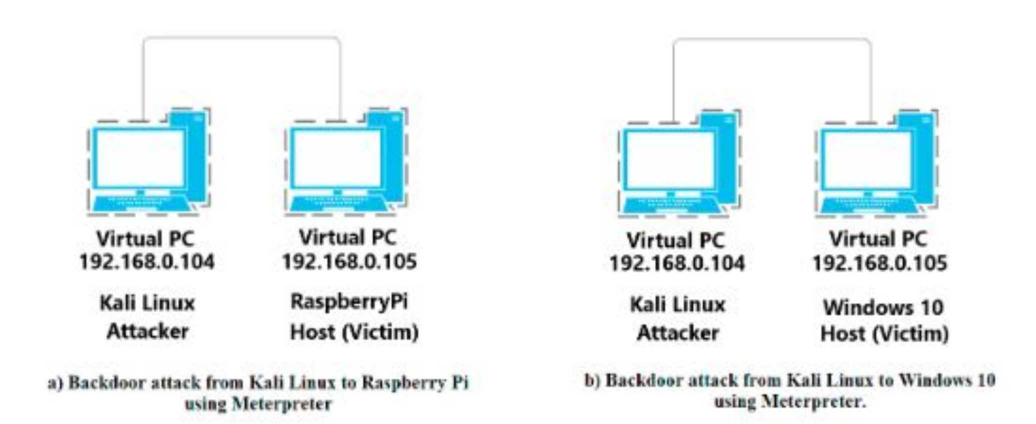


a) MITM attack from Kali Linux to a Raspberry Pi

b) MITM attack from Kali Linux to Windows10

Results - Back Door

- Common Factors
  - Meterpreter
- Raspberry Pi
  - Had to be given execution permissions
  - Successful
- Windows 10
  - Disable firewall
  - Turning off all antivirus
  - Successful

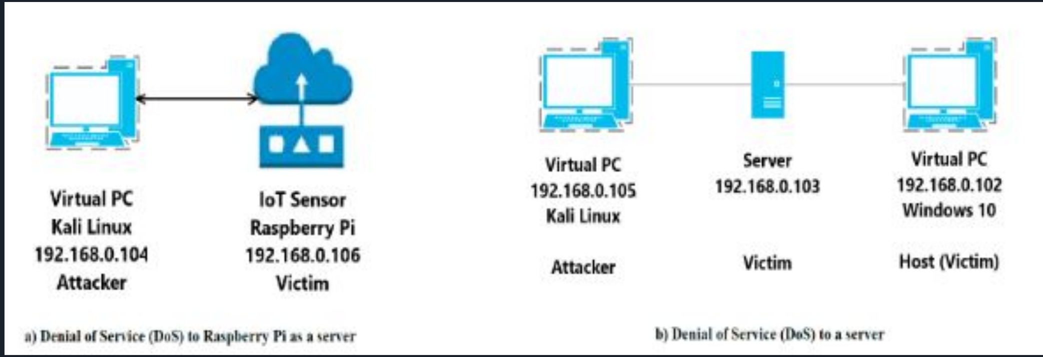


a) Backdoor attack from Kali Linux to Raspberry Pi using Meterpreter

b) Backdoor attack from Kali Linux to Windows 10 using Meterpreter.

Results - Denial of Service (Dos)

- Common Factors
  - Apache
  - MariaDB Database
- Raspberry Pi
  - Delay 4 min 15s
- Windows 10
  - Failed to mention time delay



a) Denial of Service (DoS) to Raspberry Pi as a server

b) Denial of Service (DoS) to a server

Summary of Results

Man in the middle attack (1 second)

- Raspberry Pi : 15 minutes
- Windows 10 : 15 minutes

Backdoor Attack (1 second)

- Raspberry Pi : 13 minutes
- Windows 10 : 15 minutes

Denial of Service (60 seconds)

- Raspberry Pi : 8 minutes
  - Load times 4 minutes 15 seconds
- Windows 10 : 8 minutes

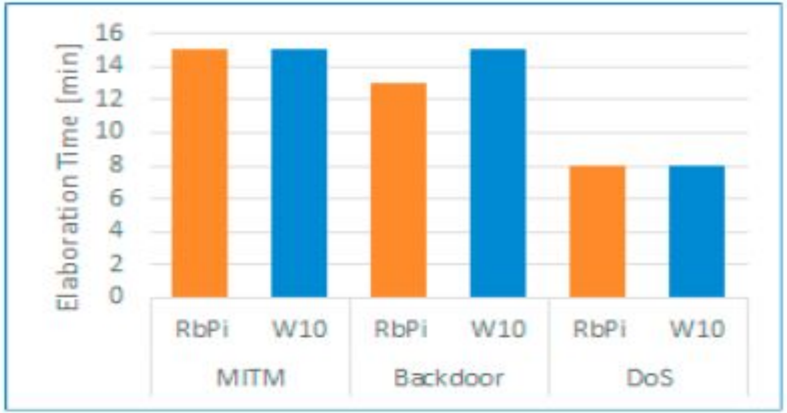
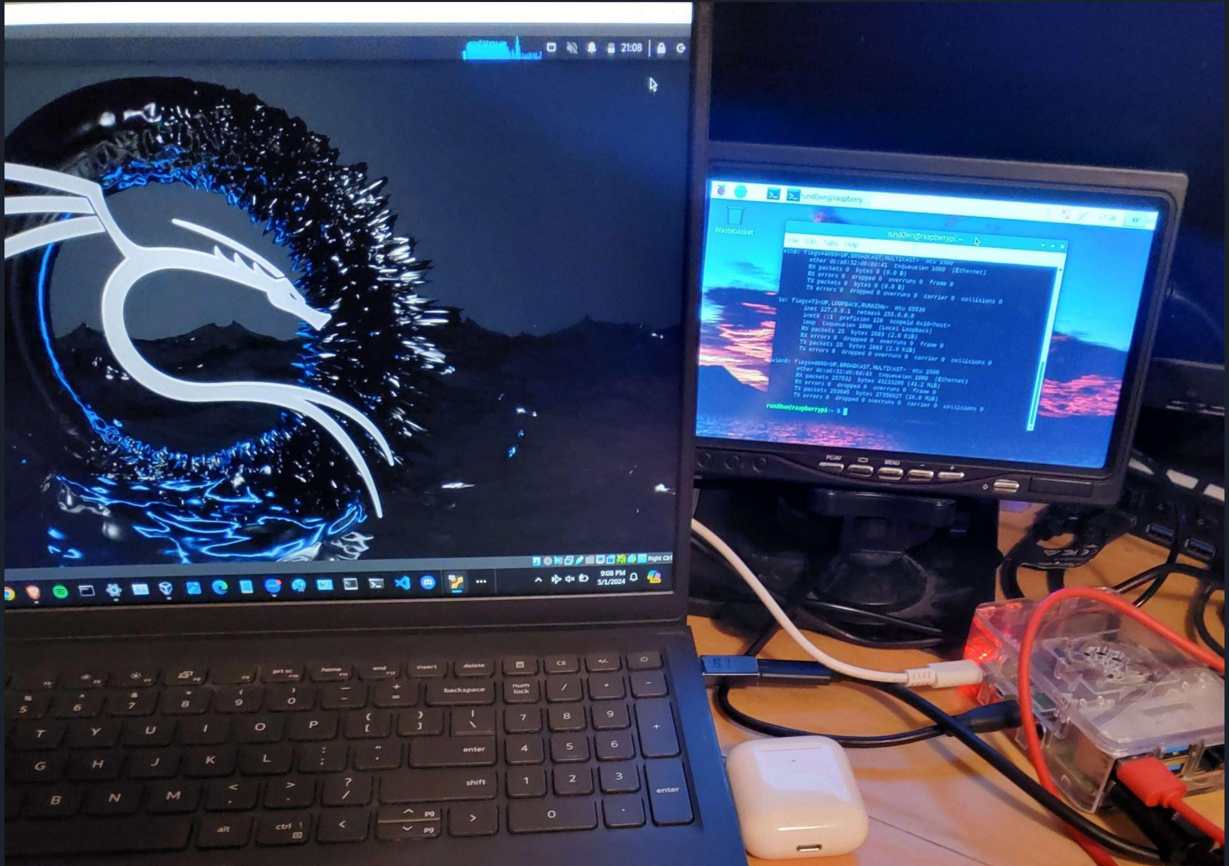


Fig. 5. Graph of attack elaboration times in the tests.

References

- J. Sainz-Raso, S. Martin, G. Diaz and M. Castro, "Security Vulnerabilities in Raspberry Pi–Analysis of the System Weaknesses," in IEEE Consumer Electronics Magazine, vol. 8, no. 6, pp. 47-52, 1 Nov. 2019, doi: 10.1109/MCE.2019.2941347. keywords: {Computer security;Internet of Things;Servers;Password;Universal Serial Bus;Software engineering},
- Nestor X. Arreaga, Genessis M. Enriquez, Sara Blanc, Rebeca Estrada, “Security Vulnerability Analysis for IoT Devices Raspberry Pi using PENTEST” ,Procedia Computer Science,Volume 224,2023,Pages 223-230,ISSN 1877-0509

Demo - MITM (Kali Linux & Pi 4)



```
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether dc:a6:32:d6:8d:41  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 25  bytes 2683 (2.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 25  bytes 2683 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether dc:a6:32:d6:8d:43  txqueuelen 1000  (Ethernet)
        RX packets 257532  bytes 43233205 (41.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 253045  bytes 27356027 (26.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

rund0wn@raspberrypi:~ $
```