IoT Security

Embedded Interface Design

with **Bruce Montgomery**

Learning Objectives

- Students will be able to…
  - Understand the basic security concerns for IoT devices
  - Understand the uses of Kali Linux
  - Consider security issues to address in their designs

My own path to Security

- Security is an important topic, but we cover it in detail in other classes:
  - CU ESE Classes: Embedded IoT Firmware, Developing the Industrial Internet of Things, Introduction to Computer Security
  - Still recommend these classes for detailed coverage
- Recently I attended Black Hat [1] and Def Con [2], leading conferences for security vendors, experts, and issues - ~20,000 attendees
- I took classes there on Hacking IoT Devices and Hardening Linux
- Opened up several things to consider

Def Con Hacker Communities

- Hacker Communities/Villages
  - IoT
  - Industrial Controls
  - Biohacking (Medical Devices)
  - Cryptography
  - Wireless
  - Lock-picking
  - Hardware hacking and Soldering skills
  - Social Engineering
  - Tamper Evident Devices
  - Data Duplication
- Recon and OSINT (Open Source Intelligence)
- Voting Machines
- AI
- Drones
- VX (Vulnerability Execution)/Chip-off
- Ethics
- Laser Cutting
- CAAD – Competition on Adversarial Attack and Defense
- Car/Automotive Hacking
- Blue Team/Defensive Measures

Things I broke into at Black Hat

- IoT Devices:
  - An MQTT broker
  - A BLE smart watch
  - A CoAP server
  - A Zigbee security strobe/speaker
  - A CANbus automobile controller (simulated)
  - An Android smart electric plug
- On an embedded Linux PCBA
  - I2C and SPI interfaces
  - An unmarked UART port
  - An unmarked JTAG port
- Thanks to Payatu for the class [3]

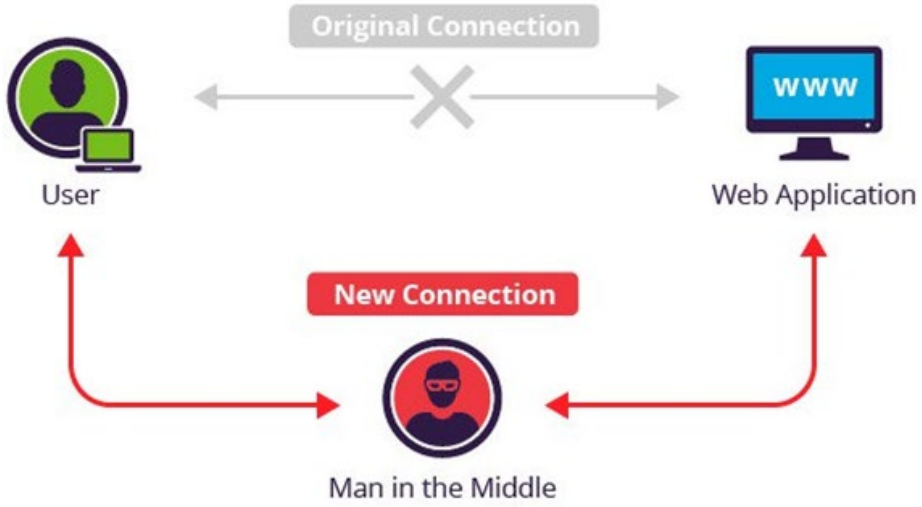
- A Kali Linux laptop I didn't have the password for
- A Wordpress web page
- A Linux system using wfuzz and dirbuster
  - Then hardened it with OSSEC
- A Linux system using nmap, dirbuster, and metasploit to break into a mailer app
  - Hardened it with ModSecurity
- A Linux system using nmap, dirbuster, a fake GIF file(!), and exploitdb
  - Hardened it with AppArmor
- Docker instances and Kubernetes clusters (Virtual Machines and management for Cloud apps)
  - Hardened with SELinux
- Thanks to InGuardians for the class [4]

Observations…

- Doesn't require fancy tools
  - Open source Kali Linux and other open source utilities like Wireshark
  - A multi-meter, breadboards, and a screwdriver
  - Occasionally, some inexpensive security-focused utility boards
- IoT Devices are particularly vulnerable
  - IoT and Industrial Control Device are some of the most vulnerable systems due to age, lack of updates, or lack of security features (tradeoff with performance and power use)
- If it's connected to a network, someone is going to try to talk to it
  - **It's only a matter of time before there's a public vulnerability, a session on hacking your company's devices at an RF or IP level, or a worse security breach and effect – inevitable**
  - You can't win, but… You have to try: potential legal and reputation costs; potential disruptive, destructive effects or information loss

IoT Attack Surfaces

- Mobile and Legacy Apps
  - Insecure storage
  - Insecure communications
  - Hardcoded information
  - Improper encryption
  - Man-in-the-middle attacks [5]



- Network
  - Custom IoT protocols
  - Radio analysis, replay, command injection
  - Improper encryption
  - Insecure protocol implementations – HTTP, SSH, Telnet…
  - Protocol vulnerabilities

Reference [3]

IoT Attack Surfaces

- Cloud/Web
  - XSS – Cross site scripting
  - SQL-i – SQL injection
  - XSRF - Cross-site request forgery
    - aka one-click attack or session riding
  - SSRF – Server side request forgery
  - Remote code execution
  - Remote device control
  - Insecure storage or communications
  - Insecure authentication or authorization
  - Improper encryption

- Devices
  - Hardware debug ports
  - Sensing interfaces
  - Insecure bus communications
  - Insecure firmware storage
  - Insecure firmware updates
  - Hardcoded information
  - Remote code execution
  - Insecure storage or communications
  - Insecure authentication or authorization
  - Improper encryption

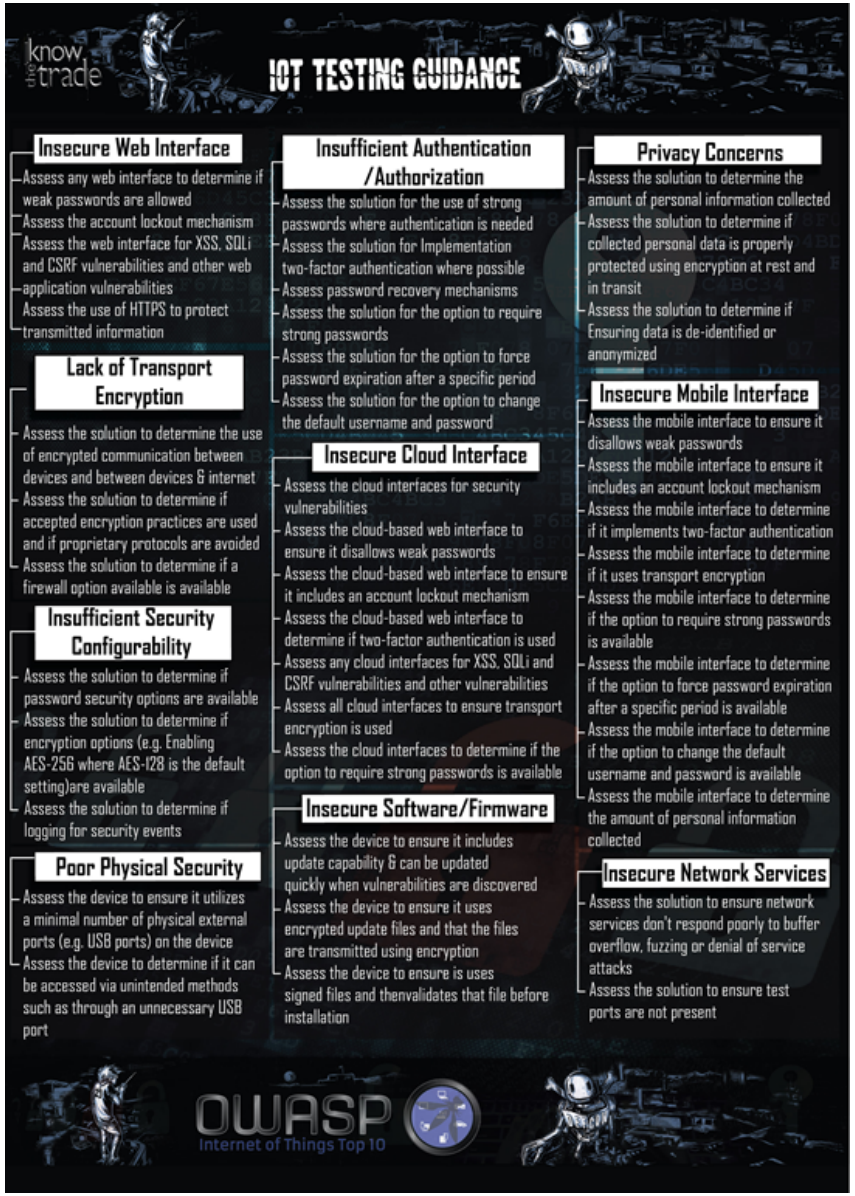
  - Reference [3]

OWASP: Top 10 IoT Security Concerns

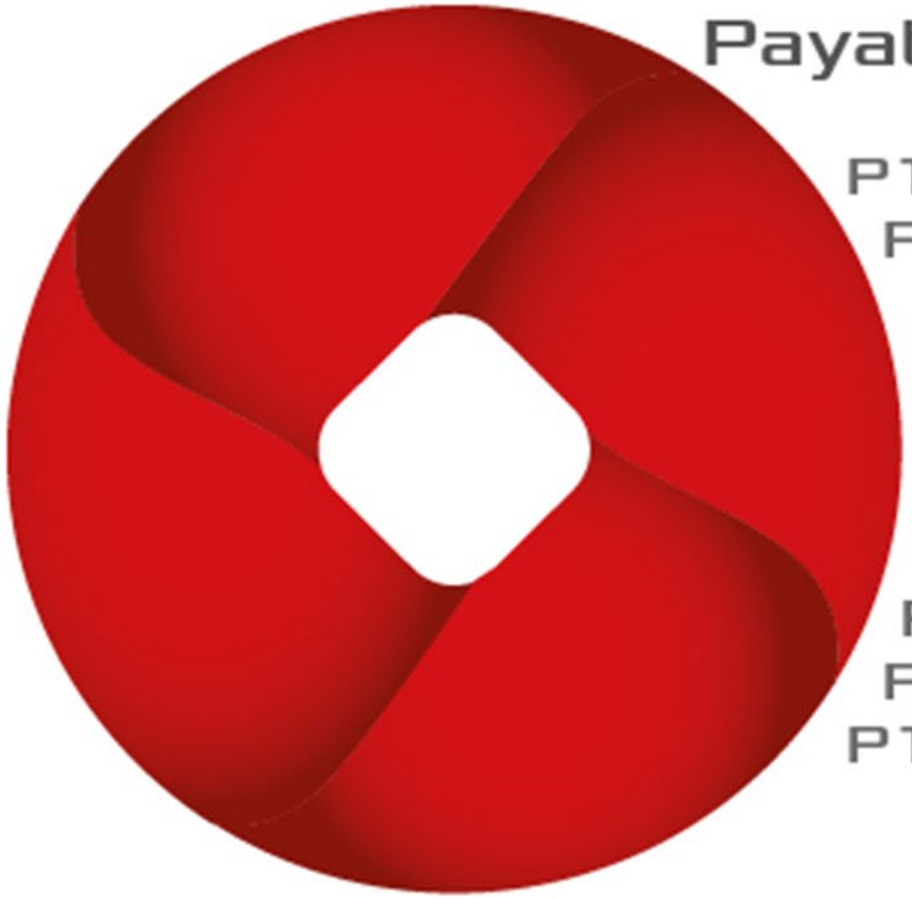
- OWASP – Open Web Application Secuity Project [6]

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

- Start at OWASP for more information on test approaches
- Testing Guidance Poster at [7], More info at [8]



Payatu: Top 10 IoT Security Concerns



Payatu IoT Top Ten Vulnerabilities

P1. Hardcoded Sensitive information
P2. Enabled hardware debug ports
P3. Insecure Firmware
P4. Insecure Data storage
P5. Insufficient Authentication
P6. Insecure Communication
P7. Insecure Configuration
P8. Insufficient data input filtration
P9. Insecure Mobile Interface
P10. Insecure Cloud/Web Interface

Reference [9]

IoT Security Testing Process

- Security Testing Process [1]
  - Requirements gathering
  - Scoping
  - Knowledge Transfer
  - Recon and Penetration Testing
  - Attack Surface/Threat Modeling
  - Firmware and Protocol Analysis for Mobile, Cloud, and Hardware elements for hardening
  - Report with recommendations
  - Retest fixes, cycle testing
- Generally recommended to focus on security of third party elements before your own, if only because those elements are more likely hacker targets (Linux, networking tools, etc.)
- Security regulation may eventually require BOMs for software elements in systems

Kali Linux

Kali Linux [10] for Penetration testing and Ethical hacking

- Features [11]
  - More than 600 penetration testing tools included
  - Free
  - Open source Git tree
  - FHS compliant: Kali adheres to the Filesystem Hierarchy Standard for Linux
  - Wide-ranging wireless device support
  - Custom kernel, patched for injection
  - Developed in a secure environment
  - GPG signed packages and repositories
  - ARMEL and ARMHF support: for SBCs like the Raspberry Pi and BeagleBone Black

Also a good book for Kali – Learning Kali Linux by Messier [13]

Tools included in Kali Linux

Amazing array of tools [12]

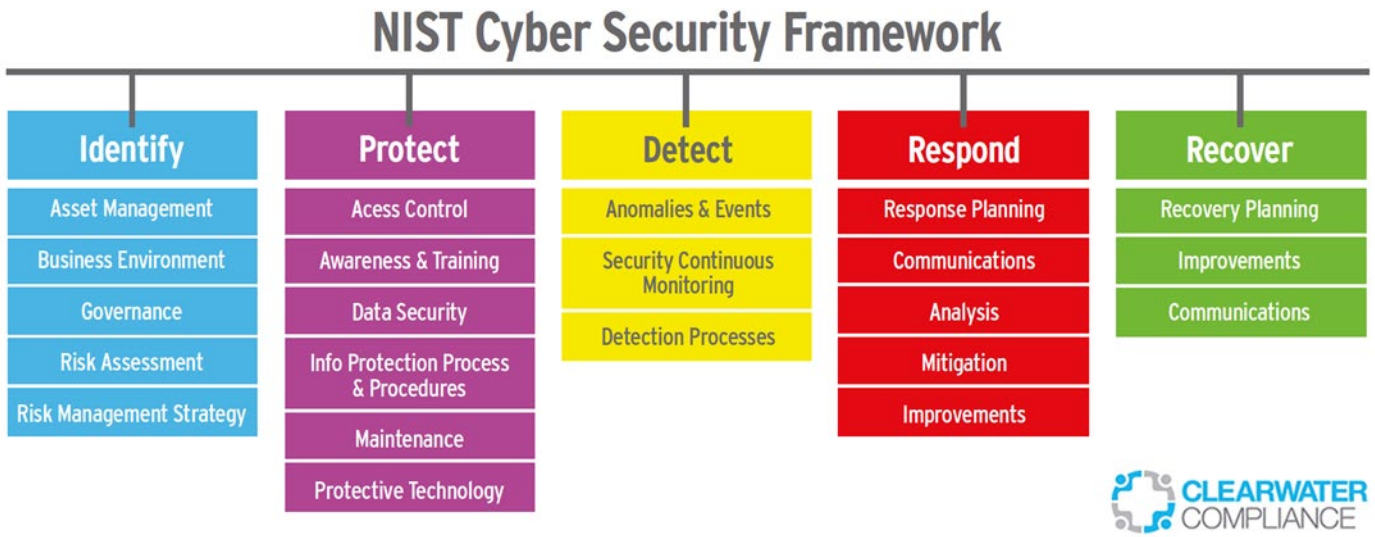
- Information Gathering
- Vulnerability Analysis
- Wireless Attacks
- Web Applications
- Exploitation Tools
- Stress Testing
- Forensic Tools
- Sniffing and Spoofing
- Password Attacks
- Maintaining Access
- Reverse Engineering
- Hardware Hacking
- Reporting Tools

UL 2900 Standards

- New industry standard for cybersecurity for network connected products [16]
- Focus is on evaluation and test for vulnerabilities, weaknesses, and malware issues as well as the presence of security risk controls
- Released
  - 2900-1 is the base standard
  - 2900-2-1 is for healthcare systems
- Under development
  - 2900-2-2 is for industrial control systems
  - 2900-2-3 is for life safety and physical security

NIST Cybersecurity Framework

- Structured approach to putting together a security development plan for your products [17, 18]
- Five core functions: Identify, Protect, Detect, Respond, Recover
- Within each, there are categories of activities to consider, with references to other guiding standards



Be careful out there…

- Only test or hack systems you have permission to
- Very easy to slip into illegal or privacy issues
- Best to set up your own local systems, not connected to the outside to test against
- Hacking ethics [14]:
- Set goals
- Plan work
- Obtain permission
- Work ethically
- Keep records
- Respect other's privacy
- Do no harm
- Use a scientific process
- Focus on core tools
- Report all findings
- Look online for penetration test practice sites to explore

Simple things to do…

- Protect your passwords/use strong passwords
  - Always change the default Pi account password
- Update to latest OS for security patches (Linux apt-get update/upgrade)
- Open access only to services you'll use
- Protect keys and certificates
- Do NOT post anything security related in code or text files that will be in a public tool like GitHub or AWS
- If you have the option, use non-standard IP Ports [15]
  - Ports 0 to 1023 are well-known ports
  - Ports 1024 to 49151 are registered ports
  - Ports 49152 to 65535 are public ports
- Be sure of the origin of your tools and connections

References

[1] https://www.blackhat.com/
[2] https://www.defcon.org/
[3] https://payatu.com/
[4] https://www.inguardians.com/
[5] https://www.incapsula.com/web-application-security/man-in-the-middle-mitm.html
[6] https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project
[7] https://www.owasp.org/images/2/2d/Iot_testing_methodology.JPG
[8] https://www.owasp.org/images/3/36/IoTTestingMethodology.pdf
[9] https://payatu.com/iot-security-part-3-101-iot-top-ten-vulnerabilities/
[10] https://www.kali.org/
[11] https://docs.kali.org/introduction/what-is-kali-linux
[12] https://tools.kali.org/tools-listing
[13] http://shop.oreilly.com/product/0636920130512.do
[14] https://www.dummies.com/programming/networking/obeying-the-ten-commandments-of-ethical-hacking/
[15] http://www.meridianoutpost.com/resources/articles/well-known-tcpip-ports.php
[16] https://industries.ul.com/cybersecurity/ul-2900-standards-process
[17] https://www.nist.gov/cyberframework
[18] https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf