Internet of Things Honeypots as Distributed Network Defenses

By Ed McCorry
What is the Internet of Things?

● The Internet of Things is the idea of connecting any and all devices to the internet.
● Cell phones, coffee makers, washing machines, headphones, lamps, fitbits, pacemakers, and even trees.
● With IPv6 there are 340 undecillion addresses
Why should we care?

- The amount of data that is produced and amalgamated.
  - Who has access to this data?
- How secure are these devices?
- How many of these devices are in our lives?
Limitations of IoT Devices

- Limited processing power
- Limited available power
- Physical exposure
- Remoteness and unmanned operation
- Network connectivity
Characteristics: Limited Processing Power

- IoT devices cannot run complicated programs
  - Virus scanners
  - Intrusion detection systems
- IoT Devices cannot run strong encryption algorithms.
Characteristics: Limited Available Power

- Battery operated
- Increased power consumption = shorter system lifetime
- Increased power consumption = increased maintenance frequency
- Power must be divided among all systems
Characteristics: Physical Exposure

- IoT devices owners rarely have immediate control over the system
- Deployed in public locations or at the customer’s premise
- Physical proximity opens the device up for attack
Characteristics: Remoteness and Unmanned Operation

- IoT devices are often deployed in inaccessible locations
- Harsh environments or remote field locations
- Updates/patches are hard or impossible to push
- If they are possible they must be automated
- Automated mechanisms present a target for attack
Characteristics: Network Connectivity

- Becoming increasingly prevalent
- Can be wired or wireless
- Access is necessary for remote control, data collection, and/or updates
- Communication in and out of the device opens up an avenue of attack, especially if the device is connected to the Internet
IoT Security Issues

- Network Intrusion
- Information Theft
- Introduction of Forged Information
- Reprogramming of Systems for Other Purposes
Software Vulnerabilities: Network Intrusion

- Done using a malware attack
- Vulnerable to same remote exploits as workstation and servers
- Examples:
  - Buffer overflows (Stack and Heap)
  - Integer arithmetic
  - Formatted output
Software Vulnerabilities: Information Theft

- Privacy issue
- Possible due to physical proximity to attacker
- Common types of information theft
  - Sensor data
  - Money off of smart cards
Software Vulnerabilities: Introduction of Forged Information

- Authenticity issue
- Done through either the system’s sensors/peripherals or directly writing to memory
- Examples:
  - Wrong video feeds in security cameras
  - Overwriting measurement data in an electricity meter
Software Vulnerabilities: Reprogramming of Systems for Other Purposes

- This is the same as stealing the system
- Many IoT devices are just general-purpose processing systems
- An example of this is reprogramming a gaming console to run Linux
Possible Solutions

- Safe Languages
- Static Code Analyzers
- Dynamic Code Analyzers
- Sandboxing or Damage Containment Approaches
Software Countermeasures: Safe Languages

- Java is a safe language but has performance ramifications
- Use safe functions in C/C++
- Use std::string. This also has performance issues
Software Countermeasures: Static Code Analyzers

- Analyze software without executing it
- Helps catch bounding errors, potential buffer overflows, arithmetic errors
- Enforces best practices
Software Countermeasures: Dynamic Code Analyzers

- Tests are run as code is compiling and running to detect vulnerabilities.
- More accurate since more info is available at runtime compared to compile time.
- Drawback is that some errors may be missed since some logic may not be tested at runtime.
Dynamic Analysis
• Also known as:
  – Web app scanning
  – Penetration testing
  – Black box testing
• Benefits
  – Quick and easy to get started
  – Simulates a hacker’s point of view
• Drawbacks
  – Difficult to exercise the entire application
  – Lacks code-level details

Static Analysis
• Also known as:
  – Source code analysis
  – Binary or byte-code analysis
• Benefits
  – 100 percent code coverage
  – Early in SDLC
• Drawbacks
  – Results require review
Software Countermeasures: Sandboxing/Damage Containment

- A sandbox is a confined execution environment
- It operates on the concept of least privilege
- Usually coupled with other access-control enforcement mechanisms
My Research

● Goal
  ○ Secure networked IoT devices and turn them against a potential adversary

● Challenges
  ○ IoT device limitations
  ○ Giving the adversary power to work with but limiting it
  ○ Data collection

● Current Status
  ○ Devices are secure
  ○ Honeypots are installed

● Future Work
  ○ Make the project offensive on top of being defensive
Goal

- Take a bunch of insecure IoT devices on a network, secure them, and then turn them into honeypots.
  - Similar to the Mirai botnet
- Learn about an attacker, how their attack propagates across a network, and prevent the attack from spreading
  - This is only the case if an IoT device is the initial vector
Process and Current Status

● Setup Raspberry Pi
  ○ Install brutespray
  ○ Create script to create and store IoT device passwords
  ○ Create configuration for data collection

● Brute force IoT logins, change them, report them back to the Raspberry Pi
  ○ Telnet and SSH

● Transfer and run installation script
  ○ Install dependencies
  ○ Install honeypot
  ○ Rewrite honeypot versions of SSH and Telnet
  ○ Configure honeypot

● Report data back Raspberry Pi or to collection server
Challenges

- Size requirement for honeypot install
  - Specifically python
- Giving the attacker the SSH and Telnet tools while limiting them
  - The adversary shouldn’t learn they are caught in a honeypot
- Data collection
  - When should the honeypots transmit?
  - Where should they transmit to?
Future Work

- Give the setup offensive capabilities as well
- The honeypots will be able to respond to a threat on top of logging it
- The data collected will be able to be funneled into other applications or setups
Questions?
References

- [http://s3.eurecom.fr/docs/secucode09_francillon.pdf](http://s3.eurecom.fr/docs/secucode09_francillon.pdf)
References

- https://en.wikipedia.org/wiki/Buffer_overflow