IoT Lockdown
Adam Englander, LaunchKey
Know This

• You will be attacked

• You will be exposed to a Zero Day vulnerability
Security is like an Ogre...

…it has layers
Layered Security

• Prevents single point of vulnerability
• Increases the cost of penetration by an attacker

“Swiss cheese model of accident causation” by Davidmack — Own work - Licensed under CC BY-SA 3.0 via Commons
Operating System Layer

- Operating System
- File System
- Services
- Application
- Network
Operating System Security

- Randomize user passwords
- Disable unused ports
- Encrypt the file system
File System Layer

Operating System
File System
Services
Application
Network
File System Security

- Named application user
- Remove “everyone” access where possible
- Restrict app user to files necessary to run
- Avoid write access – use pipes
Services Layer

- Operating System
- File System
- Services
- Application
- Network
Service Security

• Use web services for communication

• Remove all non-essential services (SSH, FTP, etc)

• Use authentication on remaining services

• Be as secure as possible with service data
Network Layer

- Operating System
- File System
- Services
- Application
- Network
Network Security

• Devise a system with only outbound IP traffic
• Restrict inbound and outbound IP traffic
• Only allow paired Bluetooth devices to connect
• Pair Bluetooth devices with challenge-response
Application Layer

- Operating System
- File System
- Services
- Application
- Network
What You Are Preventing

- Account Hijacking
- Sensitive Data Exposure
- Escalation of Privilege
- Denial of Service
- Remote Code Execution

- SensiSve Data Exposure
- EscalaSon of Privilege
- Denial	of Service
Sensitive Data Exposure

- Protect data in transit by:
  - utilizing TLS
  - not following redirects
  - pinning SSL certificates
  - using DNSSEC to verify DNS
  - encrypting data

- Protect data at rest with encryption
SSL Pinning

• Certificate verification via fingerprint
  res.socket.getPeerCertificate().fingerprint

• Have backup fingerprints to quickly rotate when primary is compromised.

• Additional certificates must use different private keys to have a different signature.
Encrypting Data

- Data at rest can use symmetric encryption as secret is not shared but local.
- Data in transit should use asymmetric encryption. It’s more complex and slower but does not require transmitting your secret.
- Both are available natively via the “crypto” package.
Symmetric Encryption

• Uses shared “secret” key.
• Uses initialization vector (IV).
• Use crypto.randomBytes() for cryptographically random IV.
• Always use some sort of block chaining or cipher feedback to ensure pseudo-randomness.
• aes-256-cbc is a good standard
Asymmetric Encryption

- Uses public/private key pairs.
- Private and public key can encrypt and verify signature.
- Only private key can decrypt and create a signature.
- Private key should be password protected.
- Key size at least 2048 bytes but 4096 bytes is preferred.
Account Hijacking

• Never use plain text credentials
• Use strong hashing:
  • PBKDF2 is in crypto package
  • 32+ character random SALT
  • 10,000+ iterations
  • sha256 digest
• If you use email for username, hash the value for storage
• Alert for changes to accounts
Encryption vs. Hashing

- Hashing is one way
- Encryption is reversible
- Hashing is more secure than encryption
- If you do not need to decrypt sensitive data, consider hashing it
Escalation of Privilege

• Always use TLS
• Set "secure" and "HttpOnly" flags for session cookies
• Use a CSRF token (nonce)
• Strict-Transport-Security
• Expire your requests
• Sign API requests including credential data and location
Nonces

- Used only once
- Must be cryptographically random
- Provided by crypto package with randomBytes()
- Should expire

"Cutting head of a paper shredder" by wdwd - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/
File:Cunng_head_of_a_paper_shredder.jpg#/media/File:Cunng_head_of_a_paper_shredder.jpg
Digital Signatures

- Verifiable hash of supplied data
- HMAC or RSA is provided by crypto
- RSA is preferred
- JOSE is IETF standard
Denial of Service

- Detection
  - Honey Pot
  - Request frequency
  - Request signatures
- Mitigation
  - Black list IPs
  - Black hole (no response)
- Detect early in process
Remote Code Execution

- Content Security Policy
  - Restrict to local source
  - No inline CSS/JS
- No eval(), ever!
- Prepared statements in SQL
- Code Object with Scope in MongoDB
Further Reading

• https://en.wikipedia.org/wiki/Layered_security
• https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project
• https://en.wikipedia.org/wiki/JSON_Web_Token