HOW TO HARDEN A MEDICAL DEVICE

It’s easier than you think.

Rochester Security Summit
2019

Daniel Megalo
Cybersecurity Engineer
Carestream

Disclaimer

The views and opinions expressed in this presentation and on the following slides are solely my own and do not necessarily represent those of my employer.

Outline

- Problem
- Fundamentals
- Secure Development Life Cycle
- Threat Modeling
- OWASP IoT Top 10
- System Components
- Final Security Review
- Conclusion

Problems Securing Medical Devices

- Physical access
- Insecure environments
- No connectivity
- Weak authentication
- Default passwords
- Limited alerting
- Unencrypted communications
- Patching
- Legacy products

Excuses

- Usability
- Inconvenience
- Too risky
- Budgets
- Timelines
- Untestable
- Works fine the way it is
- Not sure where to start

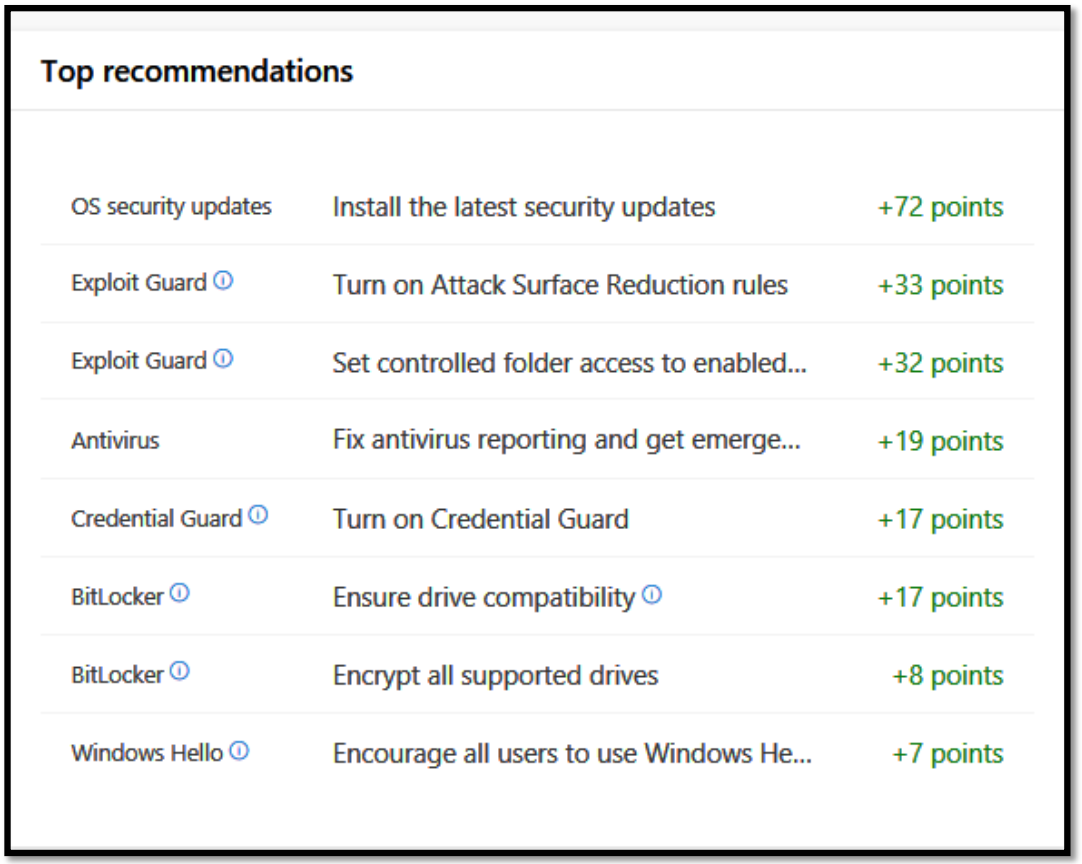
(Some) Regulations

- HHS / FDA
  - Premarket / Postmarket
- HIPAA
- EO 13636
- NIST
- ISO – 27001, 27002, 62304, 80001
- 21 CFR
- EU Medical Device Regulation
- GDPR
- IEC 62443
- FTC
- others…

Motivation

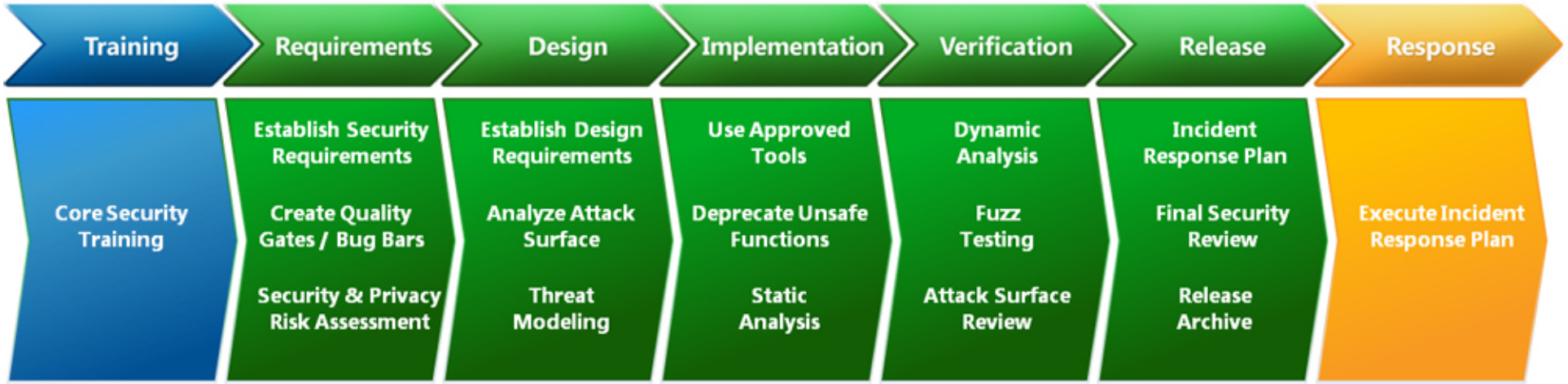
Fundamentals of Hardening

- Patches
- Remove or disable unnecessary
  - Programs
  - Services
  - Ports
  - Protocols
- Protect sensitive files
- Separation of duties
- Least privilege
- Endpoint protection



Source: Jackson, C. *Introducing the security configuration framework: A prioritized guide to hardening Windows 10*. April, 2019.

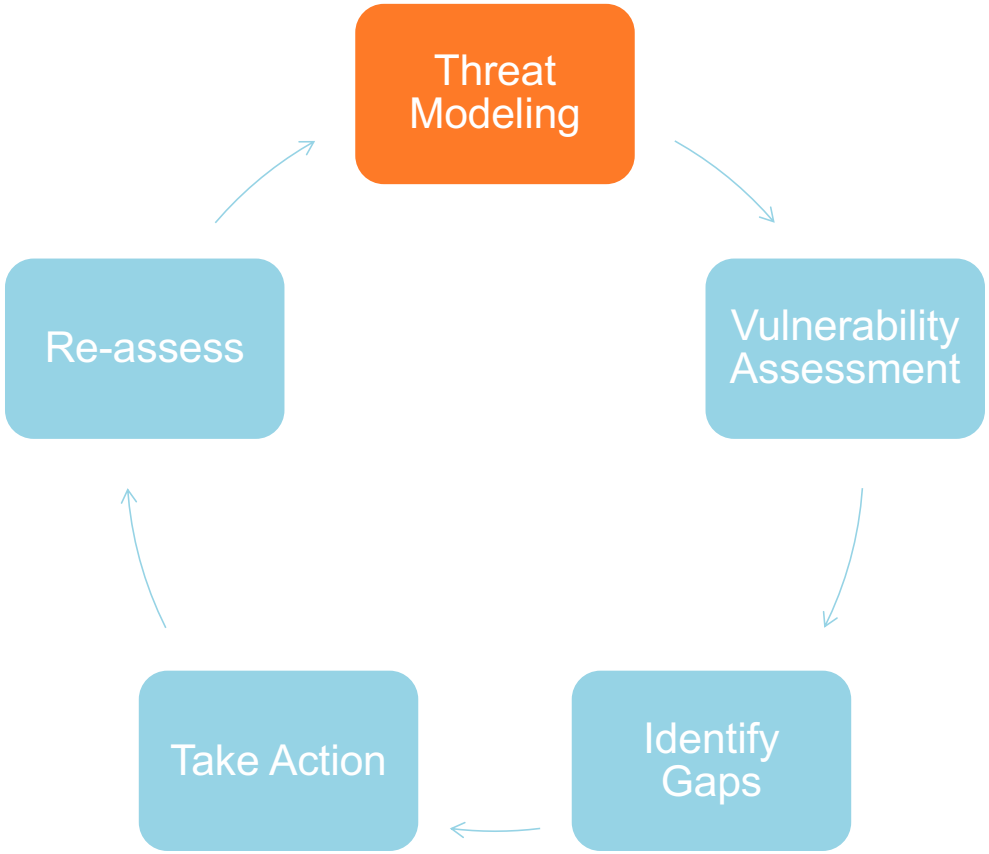
Microsoft Secure Development Lifecycle



Source: *The Security Development LifeCycle*. Microsoft TechNet. 2015

Continuous Monitoring Process

Secure by Design



OWASP IoT – Top 10

Category
1. Weak, Guessable, or Hardcoded Passwords
2. Insecure Network Services
3. Insecure Ecosystem Interfaces
4. Lack of Secure Update Mechanisms
5. Use of Insecure or Outdated Components
6. Insufficient Privacy Protection
7. Insecure Data Transfer and Storage
8. Lack of Device Management
9. Insecure Default Settings
10. Lack of Physical Hardening

Source: *OWASP Internet of Things TOP 10 2018*. OWASP IoT Security Team. 2018

Requirements and Design

- Establish security requirements
- Attack Surface Analysis
- Threat Modeling
  - Identify various types of threats
  - Based on the design of the product
  - 12 different methods

Threat Modeling Methods
STRIDE
PASTA
LINDDUN
CVSS
Attack Trees
Persona non Grata
Security Cards
hTMM
Quantitative TMM
Trike
VAST Modeling
OCTAVE

Source: Shevchenko, N. et al, *THREAT MODELING: A SUMMARY OF AVAILABLE METHODS*. Carnegie Mellon University, Software Engineering Institute. July 2018

S.T.R.I.D.E. – Threat Categories

Threat	Property	Definition	Example
Spoofing	Authentication	Impersonating something or someone else	Pretending to be Microsoft.com
Tampering	Integrity	Modifying data or code	Modifying a DLL or packet
Repudiation	Non-repudiation	Claiming to have not performed the action	“I didn’t modify that file”
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Publishing a list of customers to a web site
Denial of Service	Availability	Deny or degrade service to users	Crashing Windows or a web site
Elevation of Privilege	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands

Source: Shostack, A. *STRIDE chart*. Microsoft Security. September 2007

Know Your Components

- Operating System
- Application
- Database
- File system
- Network
- Firewall
- Endpoint software
- Encryption
- 3rd party components
- Etc.

Really Know Your Components

- systeminfo
- netstat –abno
- tasklist
- wmic qfe
- wmic nic get AdapterType, Name, Installed, MACAddress, Speed
- wmic startup list full
- netsh advfirewall show all
- driverquery /v
- nmap –sT –sU –A –p 1-65535
- …

Deep dive every component.

Implementation and Verification

- Vulnerability scanning
- Static and dynamic code analysis
- Fuzz testing
- Manual testing

Release

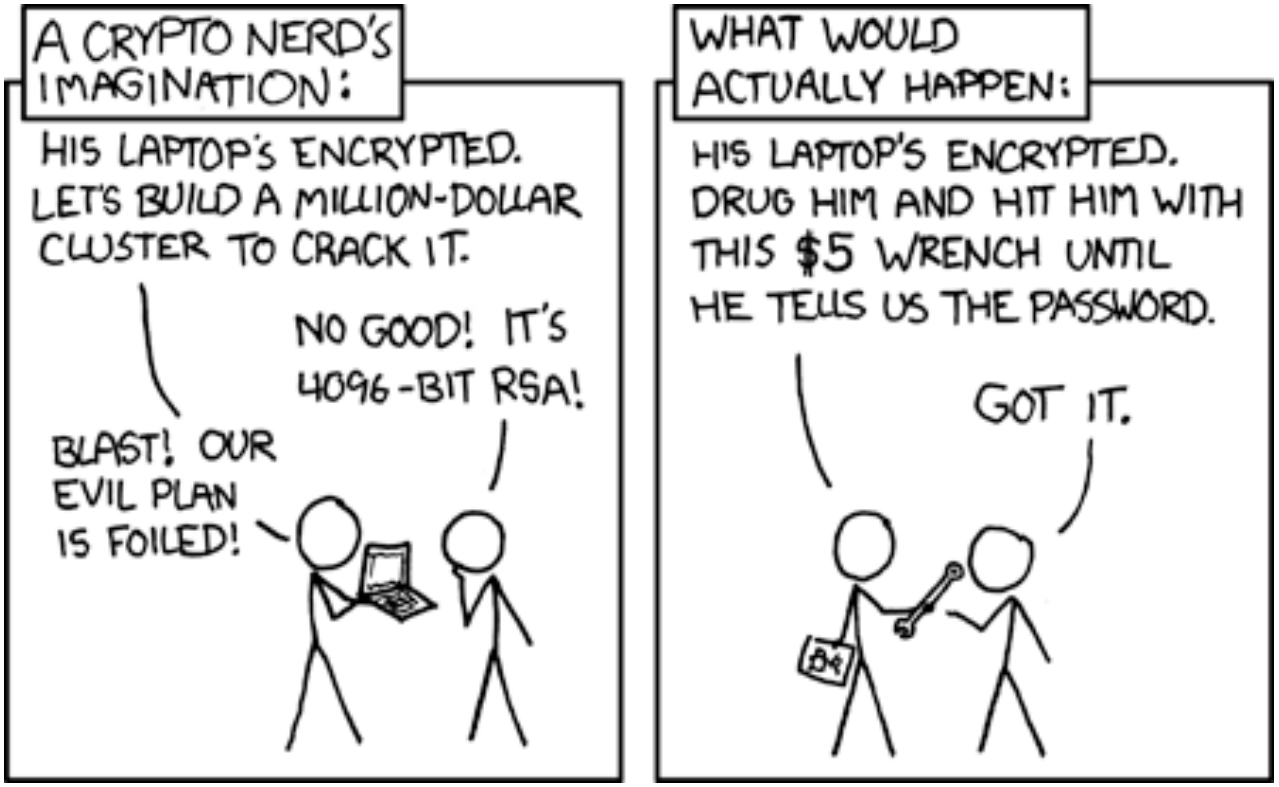
- Final Security Review
  - Fully patched
  - Minimal services
  - Only required ports
  - Physical controls
  - Automated and manual assessment findings
  - Only required software and features
  - 3rd party vulnerabilities
  - Drivers up to date
  - Open source libraries

Conclusion

- Security must begin with design and continue throughout lifecycle
- Pick a framework and use it
- Identify why every component is required; remove the rest
- Securely configure using CIS benchmarks or DISA STIGS
- Know your threats
- Educate those around you
- Continuously monitor for vulnerabilities

It’s easier than you think.
It just takes time.

Questions?



Source: *Security*. XKCD

References

- Jackson, C. *Introducing the security configuration framework: A prioritized guide to hardening Windows 10*. April, 2019. https://www.microsoft.com/security/blog/2019/04/11/introducing-the-security-configuration-framework-a-prioritized-guide-to-hardening-windows-10/
- *OWASP Internet of Things TOP 10 2018*. OWASP IoT Security Team. 2018. https://www.owasp.org/images/1/1c/OWASP-IoT-Top-10-2018-final.pdf
- *Security*. XKCD. https://xkcd.com/538/
- Shevchenko, N. et al, *THREAT MODELING: A SUMMARY OF AVAILABLE METHODS*. Carnegie Mellon University, Software Engineering Institute. July 2018. https://resources.sei.cmu.edu/asset_files/WhitePaper/2018_019_001_524597.pdf
- Shostack, A. *STRIDE chart*. Microsoft Security. September 2007. https://www.microsoft.com/security/blog/2007/09/11/stride-chart/
- *The Security Development LifeCycle*. Microsoft TechNet. 2015. https://social.technet.microsoft.com/wiki/contents/articles/7100.the-security-development-lifecycle.aspx