AppSec Application security Labs

AppSec Labs Ltd.
info@appsec-labs.com
https://appsec-labs.com

Cyber Security Testing of Medical Devices

Erez Metula

Application Security Expert

Founder, AppSec Labs

ErezMetula@AppSec-Labs.com

Agenda

- Changes in cyber security testing of medical device
- Types of security testing you must know about
- Stages of a security test
- Common vulnerabilities often seen in tested products
- Example – cyber security testing report

About Me & AppSec Labs

- Founder of AppSec Labs 
- Application Security Expert
- Book author - Managed Code Rootkits 

- Speaker & trainer

- AppSec Labs focuses on penetration testing, with special expertise in medical devices

Well experienced with medical device security testing

- We do cyber security testing from 2010
- Some of our medical clients

Game changer

- The FDA has become increasingly focused on medical device cybersecurity, as can been seen by the frequency and length of their guidance (2014, 2016, 2018, 2022, and 2023).
- in 2023, congress gave FDA additional authority to regulate the cybersecurity of medical devices
- The threat is real
  - Critical patient data no longer available
  - Sensitive PII/PHI had stolen
  - People are afraid to use connected medical devices
  - People died !

A significant step

- The world moved from data on papers, physical old school connected devices, to smart, cloud connected devices, that can operate remotely
- A bit late, but not too late
- Significant step towards more secure medical devices
- Regardless of the FDA, you also protect your own company and product
- Embrace this opportunity !

Major cyber security testing types you must know about

- Penetration testing
- Static code analysis
- Dynamic code analysis
- Fuzz Testing
- Vulnerability assessment
- SBOM

- Output of those tests are part of your cybersecurity premarket submission!

Penetration Testing

- Penetration testing is an offensive security testing approach, using both **manual** techniques and **automated** tools.
- Penetration Testing involves **simulating** cyber-attacks against the medical system to identify and exploit security vulnerabilities.
- This proactive approach helps in understanding potential attack vectors and the effectiveness of existing security measures.
- It's crucial for ensuring that the system can withstand real-world attack scenarios, safeguarding patient data and system integrity.

Static Code Analysis

- Static code analysis uses both **manual review and automated analysis** using software tools.
- This process involves scrutinizing the system's source code **without executing** it, specifically to identify cybersecurity vulnerabilities such as potential backdoors, buffer overflows, and insecure coding practices.
- By analyzing the code statically, it uncovers security flaws that could be exploited by attackers.
- This method is crucial for ensuring that the software components of the medical system are robust against cyber threats, thereby protecting sensitive patient data and the system's overall integrity from security breaches.

Dynamic Code Analysis

- Dynamic code analysis includes **manual examination** conducted by testers and an **automated analysis** with software tools.
- Dynamic Code Analysis involves testing the system in a **live environment**, under real-time operating conditions.
- This method identifies security and operational issues that **may not be visible** during static analysis, such as runtime errors and memory leaks.
- It is vital for assessing the system's behavior under normal usage and extreme conditions, ensuring reliability and security in practical scenarios.

Malformed Input (Fuzz) Testing

- Fuzz testing uses malformed inputs to identify vulnerabilities.
- Fuzz Testing exposes the system to **unexpected or random inputs** to assess its ability to handle abnormal or unforeseen data.
- This type of testing is key in identifying potential vulnerabilities like buffer overflows, which could lead to system crashes or data corruption.
- It ensures that the medical system remains stable and secure even when faced with irregular or erroneous input.

Vulnerability Scanning

- Vulnerability Scanning is an automated process that identifies **known security weaknesses** in the system, such as outdated components or misconfigurations.
- Regular scanning is critical for the early detection of potential vulnerabilities, allowing for timely remediation.
- This is integral to maintaining the ongoing security and compliance of the medical system, ensuring it remains protected against evolving cybersecurity threats.

SBOM - Software Bill of Materials

- Medical devices incorporate significant amounts of software, both proprietary and open-source
- Vulnerabilities in software are discovered by time to time, the software ages and becomes unsupported
- medical device software supply-chains are documented and shared with regulators, users, and other appropriate parties
- SBOM enables transparency of your SW packages
- Machine readable.
  - Recommended tool: CycloneDX

- **Example** bom-server_example.json

Major steps for conducting a cyber security testing

- Scoping
- Device vendor provides the device (usually with source code)
- Kickoff meeting, meeting with developers, questions, etc
- Actual testing – usually takes 3-4 weeks
  - Realtime vulnerability notification is still not common for most security companies
- first report, outlining all vulnerabilities detected during the various testing stages
- Vendor fixes the vulnerabilities, and ask for a retest
- Security vendor is performing a retest (sometimes a couple of cycles are needed) for full remediation
- Final report is compiled and added to premarket submission documentation

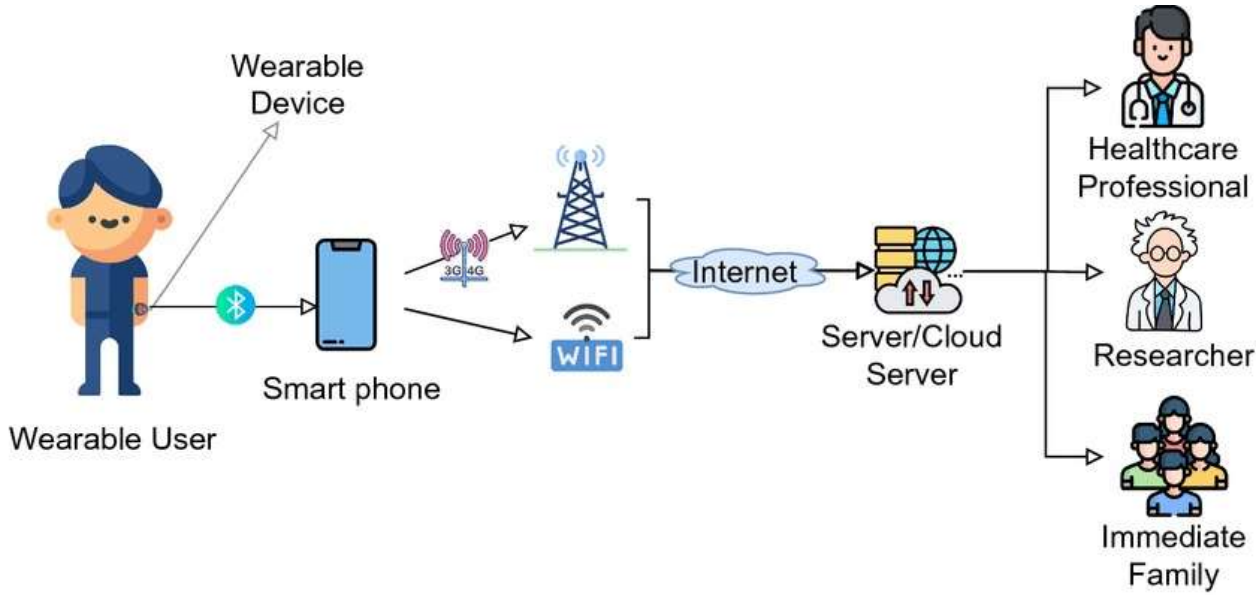
What should be in scope?

- The **medical device** (obviously…)
- **and** its **supporting components**:
  - Web apps
  - Mobile apps
  - REST API
  - Peripherals (sensors, remote controls etc)
  - Communication using BLE, WIFI and such

What actions are usually done in the penetration test lab

Some examples – what is performed for each system component

1. Code Review
2. Analyze all storage – device, PC, mobile apps, etc
3. Sniffing of ALL traffic – wifi, ethernet, BLE
4. Look into configuration files – looking for secrets, sensitive info
5. Port scanning – looking for open ports
6. Assessment of know vulnerabilities – OS & software packages
7. Manipulation of remote API's
8. Penetration testing – authentication, authorization, Denial of Service, etc

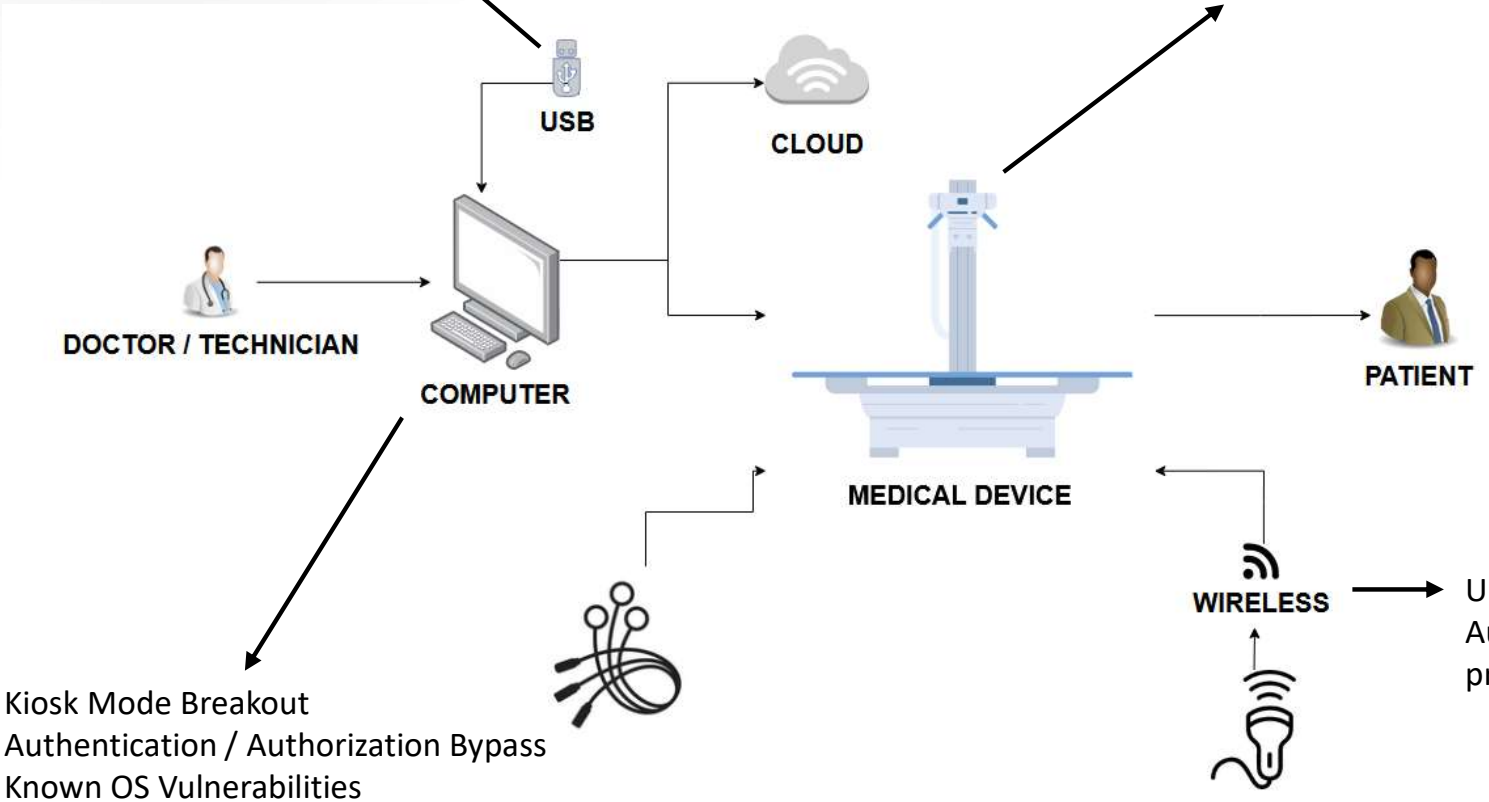


Common problems
case study 1 – treatment machine

Insecure import/export
Denial of Service

1. Authentication / Authorization Bypass
2. Injections

1. Open Ports
2. Insecure Firmware Update

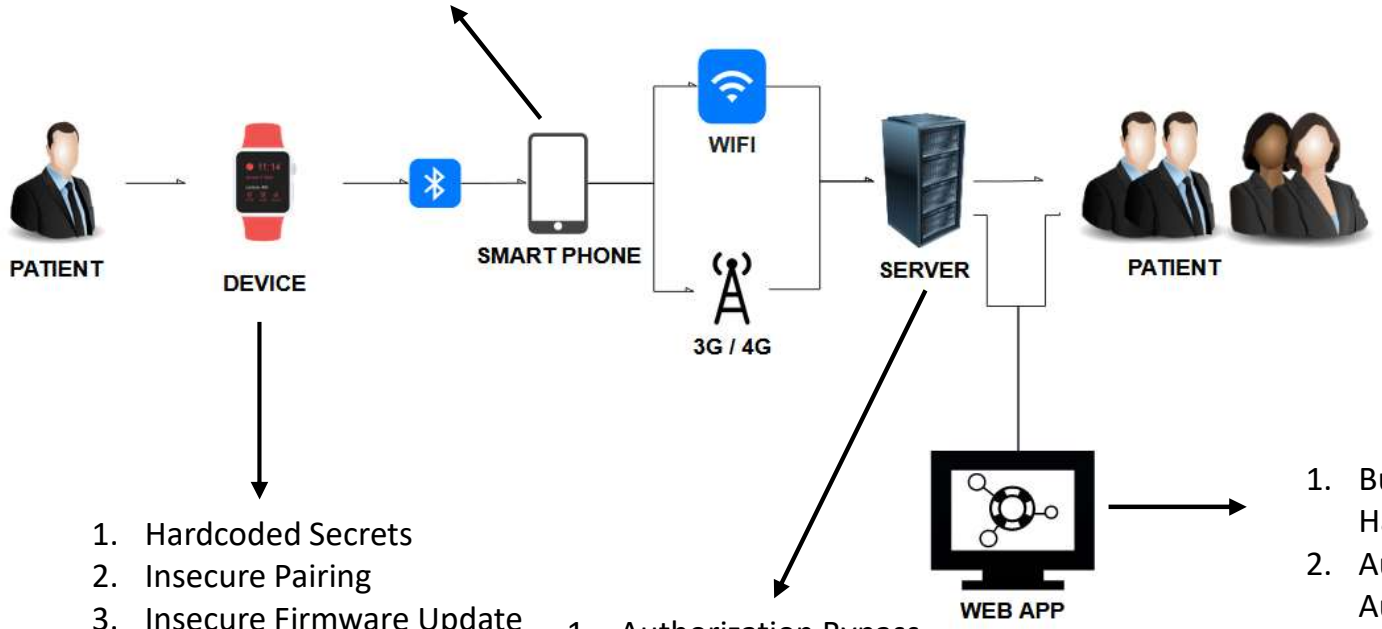


Unencrypted Traffic
Authentication problems

1. Kiosk Mode Breakout
2. Authentication / Authorization Bypass
3. Known OS Vulnerabilities
4. Vulnerable Services
5. Hardcoded Secrets
6. Unprotected patient data

Common problems
case study 2 – wearable/smart device

1. Hardcoded API Keys
2. Authentication Bypass
3. Cleartext Secrets in Local Storage



1. Hardcoded Secrets
2. Insecure Pairing
3. Insecure Firmware Update

1. Authorization Bypass
2. SQL / Commands Injections
3. Denial of Service

1. Business Logic Bypass / Hardcoded Secrets
2. Authentication / Authorization Bypass
3. SQL / Commands Injections

How a cyber security testing document looks like

- General details like system name, date of testing, types of testing, Duration of testing
- Scope
- Methods and tools used during the test
- Name of penetration testers, skills, etc.
- List of findings
  - Finding name
  - Threat level
  - Description
  - Attack demonstration (PoC)
  - Mitigation

- **If time permits: Cyber Security Testing Report.docx**

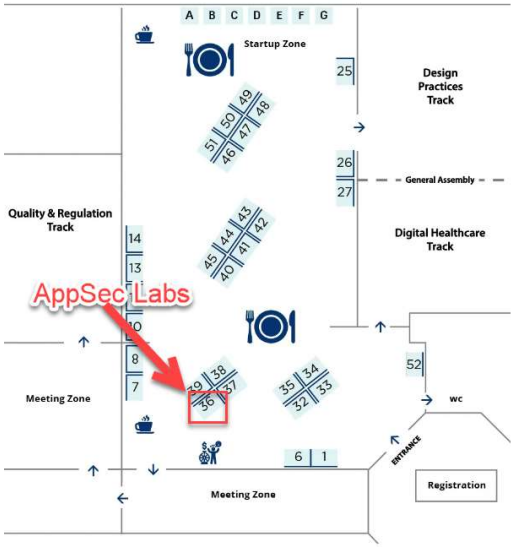
Summary

- Plan for it – don’t wait for the last minute
- Look for components that have built in security (chips with secure boot, OS with built in kiosk mode, network protocols that are secure, etc.)
- Make sure your FOTA is secure
- Encrypt/Sign all data at rest
- Most important - Do not reinvent the wheel. Look at what’s going on at other industries.

QUESTIONS ?

THANK YOU!

Meet me at booth #36



Erez Metula

Application Security Expert

Founder, AppSec Labs

ErezMetula@AppSec-Labs.com