A threat analysis of a networked medical device

Danny Lieberman, Open Solutions
This work is licensed under the Creative Commons Attribution License
www.software.co.il
dl@software.co.il

Abstract
A threat analysis was performed on a networked Windows-based embedded medical device used for patient monitoring. The system helps hospital staff prevent crisis situations through ongoing supervision of patient status, early detection of warning signs, and alert notifications of changes in patient condition. The threat analysis used the PTA (Practical threat analysis) methodology, described in Appendix A.

Our analysis considered threats to three assets: medical device availability, the hospital enterprise network and patient confidentiality. Following the threat analysis, a prioritized plan of security countermeasures is suggested in Section III. We devoted special interest to the issue of propagation of viruses and malware into the hospital network. Our analysis shows that installing anti-virus software on the medical device is less effective than implementing other security countermeasures that mitigate the most severe threats – ePHI leakage, software defects and USB access to the bedside units. A detailed discussion appears in Section IV of this paper. Section V suggests segregating the bio-med functions from the hospital enterprise IT. Section VI provides a summary of the analysis and its findings.

A novel benefit of our approach is derived by providing the analytical results as a standard PTA threat model database, which can be used by the medical device manufacturers and hospital customers to model changes in risk profile as technology and customer environment evolve. The threat model can be downloaded at www.software.co.il/downloads/MedicalDeviceThreatModel.zip and the PTA software that runs the threat model: http://www.software.co.il/pta
I. Introduction

A threat analysis was performed on a networked medical device that performs real
time patient monitoring at the bedside. The analysis considers the security
implications of deploying the devices inside a hospital network. A key concern for
hospital IT management is whether the medical devices create new entry points for
viruses and malware in the enterprise network.

System configuration
The system configuration consists of fan-less, diskless embedded devices, with a
touch screen and an Intel processor running Windows XP. Flash storage is used for
the operating system and application software. The devices have two Ethernet ports,
a USB port and are networked using a proprietary TCP socket protocol to nursing
stations that provide a management console function. The devices are not members
of a Microsoft Active Directory domain and do not have Internet connectivity.

The process
A data collection phase employed face to face interviews with software and
hardware developers and directly examined the medical device software and
hardware. We identified potential attackers, entry points, threats, vulnerabilities and
security countermeasures (those already implemented and those that might be
implemented). Following data collection, we performed a threat analysis using the
PTA (Practical Threat Analysis) methodology summarized in Appendix A and
described at length on the PTA Technologies web site.

Threat model entities
Our threat model uses four base classes; mapping assets to vulnerabilities, threats
that exploit vulnerabilities and countermeasures that mitigate vulnerabilities.

For example:

| Threat T1 – an attacker may obtain monitoring information and impact |
|-------------------------|-------------------------|
| Asset A1 – patient privacy |
| Vulnerability V1 – Nursing station has Internet connectivity |
| Countermeasure C1 – Encrypt ePHI |

The key assets were medical device availability, the hospital enterprise network and
patient confidentiality. We received input from hospital IT management regarding
annual rates of occurrence of virus and malware attacks (rare) and phishing attacks
on hospital employees (the usual email-borne pharmacy scams etc...).
II. Top unmitigated threats

After building the threat model with the four base classes and their relationships, we estimated the probability of threat occurrence, percent damage to assets and risk mitigation effectiveness. Trusted insider information leakage event frequency was estimated as twice/year in the threat model, while virus, denial of service and malware attacks frequency were estimated to be rare (less than once every 3 years). Hospital IT were primarily concerned with the health of their enterprise network (as opposed to the availability of the medical devices) – described in threat T017.

The 5 most severe unmitigated threats in our model (shown below), are derived using the PTA calculative method (http://www.ptatechnologies.com/?action=4pta), which takes into account estimated asset value, threat probability and percent damage due to a threat event.

The TXXX identifiers in the left hand column refer to the entities in our model.

<table>
<thead>
<tr>
<th>Entity Id</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>T002</td>
<td>Trusted insiders may leak ePHI to interested parties (insurance companies etc...)</td>
</tr>
<tr>
<td>T019</td>
<td>Software defects and/or configuration changes may cause the units to become unresponsive and incapable of providing the patient monitoring service</td>
</tr>
<tr>
<td>T017</td>
<td>The Windows-based medical devices may become infected and propagate malware/viruses to the hospital enterprise network</td>
</tr>
<tr>
<td>T001</td>
<td>Malicious agents may access the system from inside the hospital network in order to steal, modify data or disrupt operation.</td>
</tr>
<tr>
<td>T021</td>
<td>Hardware defects may cause the units to become unresponsive and incapable of providing the monitoring service</td>
</tr>
</tbody>
</table>
Removing electronic Protected Health Information (ePHI) from the medical device

Unauthorized disclosure of ePHI (T002) was nominally the most severe threat at the start of the analysis due to compliance (HIPAA) / patient privacy concerns.

Protected health information (PHI) is any information in the medical data set that can be used to identify an individual and that was created, used, or disclosed in the course of providing a health care service such as diagnosis or treatment.

Following the threat analysis, it was decided to remove all personally identifying information and use an alphanumeric designator, displayed on the bedside unit screen and at the nursing station. Nurses can respond quickly to alarms of changes in patient signs (heart/respiratory rate) reported by a particular bedside unit without being exposed to personally identifiable information.

After removing ePHI, the risk assessment changed and the threat of the medical device infecting the hospital enterprise network (T017) then became our primary concern.
III. Recommended countermeasure plan

Using the PTA quantitative threat model, we then calculated a prioritized plan of security countermeasures as shown in the following table. The below table is sorted according to recommended priority of implementation in terms of risk mitigation effectiveness. After implementing the below countermeasures, the calculative model estimates a residual risk of less than 3% to system assets.

Security countermeasures plan

The TXXX and CXXX numeric identifiers refer to threat and countermeasure entities in our threat model.

<table>
<thead>
<tr>
<th>Threats/Security countermeasures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>T002</strong> - Trusted insiders may leak ePHI to interested parties (insurance companies etc...).</td>
</tr>
<tr>
<td><strong>C061</strong> - Remove ePHI (protected health information) from the system</td>
</tr>
<tr>
<td><strong>T019</strong> - Software defects and/or configuration changes may cause the units to become unresponsive and incapable of providing the patient monitoring service</td>
</tr>
<tr>
<td><strong>C014</strong> - Perform software security assessment of relevant module/component/functions and QA review</td>
</tr>
<tr>
<td><strong>C041</strong> - Set write permissions at start of upgrade procedure</td>
</tr>
<tr>
<td><strong>C055</strong> - Perform post-install, post-software update validation check</td>
</tr>
<tr>
<td><strong>C057</strong> - Use updated .NET framework from Microsoft and upgrade the report writer application at the nursing station that uses .NET to produce PDF reports</td>
</tr>
<tr>
<td><strong>T017</strong> - The Windows-based medical devices may be infected by a USB device and propagate malware/viruses back to the hospital enterprise network</td>
</tr>
<tr>
<td><strong>C048</strong> - Implement an IO-board hardware toggle for disabling USB ports</td>
</tr>
<tr>
<td><strong>C039</strong> - Implement a procedure for ensuring clean device version update media</td>
</tr>
<tr>
<td><strong>T001</strong> - Malicious agents may access the system from inside the hospital network in order to steal, modify data or disrupt operation.</td>
</tr>
<tr>
<td><strong>C001</strong> - Block enterprise network access to bedside monitoring units</td>
</tr>
<tr>
<td><strong>C047</strong> - Configure communications software to validate and discard invalid messages</td>
</tr>
<tr>
<td><strong>C052</strong> - Implement system tcp/ip data messages in binary format that are relatively difficult to decode via sniffing</td>
</tr>
<tr>
<td><strong>C061</strong> - Remove all ePHI (protected health information) from the system</td>
</tr>
<tr>
<td><strong>T021</strong> - Hardware defects may cause the units to become unresponsive and incapable of providing the monitoring service</td>
</tr>
<tr>
<td><strong>C058</strong> - Provide system health check and expose alert to nursing station</td>
</tr>
</tbody>
</table>
Patch management

The question of patch / update management arose during the study – the results are perhaps counterintuitive for typical IT managers:

- IT policy of running automated Windows Update on Windows PCs in the office is not necessarily a relevant countermeasure for embedded medical devices.

- Although FDA 510(K) recertification of the medical device may not be necessary when applying security patches – running Windows Update is practically impossible in an embedded device that does not have Internet access. The medical device vendor would typically apply patches to the embedded image as part of ongoing device field maintenance.

We note that the bedside monitoring system is a specialized (not COTS) embedded device that does not have Internet or removable device connectivity. The device does not run MS Office, does not run IE and is not connected to the Internet and therefore has a much smaller threat surface than a typical Windows PC installed on the hospital network.

For these reasons, we focused our efforts on security countermeasures that would reduce the most severe threats – ePHI leakage, software defects and USB access to the bedside units.
IV. Propagation of viruses/malware in the enterprise network

One of the key security concerns when operating networked, Windows-based embedded medical devices is whether new entry points for viruses and malware are created in the enterprise network.

We sub-divided this concern into 3 separate threat scenarios:

1. Can the medical devices be infected from the enterprise network?
2. Can the medical devices be infected via USB devices?
3. Can infected medical devices propagate malicious software back into the enterprise network?

1. Can the medical devices be infected from the enterprise network?

The short answer is no.

The medical device analyzed in the study uses Windows XP/SP3 and run a proprietary TCP/IP messaging protocol in order to communicate with the nursing station. The data flow using TCP/IP sockets follows the following high-level message flow:

A bed pressure sensor sends a signal via RS232 to a thread that acquires data and sends the data to a DSP sub-system via an exported function interface that returns heart rate (HR) and respiratory rate (RR) metrics based on patient movement. The bedside unit operator can set thresholds for alerts.

A local TCP/IP socket that binds to localhost is used to send trend data and alerts to the local bedside unit GUI. Heart rate and respiratory rate data is averaged and accumulated into a local binary event log at a sampling frequency of 2hz. A remote socket is used by the bedside unit to send alert data to the remote GUI at the nursing station. The nursing station uses the remote socket to request reports from the bedside unit, which responds by sending a report file in PDF format.

The TCP/IP port numbers are defined in a configuration file included in the factory installation package. As noted above, there are internal and external sockets, two external sockets are used, one for transferring binary data and the other for transferring XML text; these sockets are in the range 3030-3035. The TCP/IP message format contains a 2 byte guard prefix, a size field, a 2 byte guard suffix and a checksum. The TCP/IP listeners validate the message using the guard fields, the checksum and length, enforcing a maximum message length of 2K although most messages are about 200 bytes.

The operating system itself is hardened, does not run Windows shell, does not run IE and shuts down all unneeded services such as SMB and RPC. In addition, it runs a Windows Firewall instance that blocks all ports except the TCP/IP listener ports.
Although a dedicated attacker with the right skill set might be able to sniff traffic, reverse engineer the protocol and fuzz there is always the question of whether or not the value of the asset justifies the attack. In this configuration – where the bedside units use guard fields, validate message lengths and contain numeric heart rate and respiratory rate data we felt that the medical device software is not particularly vulnerable to such attacks.

2. *Can the medical devices be infected via USB devices?*

The short answer is yes – potentially by anyone who inserts an infected USB removable storage device.

We have addressed this vulnerability in countermeasure C048 – “Implement an IO-board hardware toggle for enabling/disabling USB ports”. We also strongly recommend migration to Linux – with no USB auto-run functions (see next section).

The bedside unit uses an IO board with a hardware toggle on the USB lines. The USB lines are normally disabled (i.e. the gate is normally open). When a device technician needs to perform an update, he enters a maintenance password and the medical device software enables USB access temporarily.

3. *Can USB-infected medical devices propagate malicious software back into the enterprise network?*

The short answer is yes.

Although the proprietary request/response communications protocol used by the units cannot be used to transport files to other Windows PCS, a worm such as Conficker can exploit vulnerabilities in Windows services, disable the Windows firewall and propagate from the source computer to the hospital network on an arbitrary port between 1024 and 10000.
V. Future: segregation of bio-med and IT domains

While hospital IT typically uses Microsoft Windows; for embedded networked medical devices, we suggest using Linux due to its ease of maintenance and resistance to USB exploits and worms such as Conficker that exploit Windows software. The suggested configuration would consist of an up-to-date Linux kernel, touch-screen support and a main loop to run the medical device application. A more detailed discussion of the proposed Linux implementation is beyond the scope of this article.

VI. Summary

A threat analysis of a networked medical device was performed and a mitigation plan of countermeasures was produced, including recommended priority of implementation.

As a result of the analysis, it was decided to modify the medical device design and not to store ePHI. This is obviously the most effective countermeasure possible for HIPAA compliance and protecting patient privacy.

Attack vectors using infected USB storage devices remain a concern and need to be addressed by migrating from Windows XP to Linux.

The methodology itself proved to be highly effective in identifying the key issues and most cost-effective security countermeasures in a demanding schedule dictated by an end-user hospital customer. Data collection started on June 1, 2010 and by June 4, 2010, a preliminary threat model and draft report was delivered to the customer. The final report was produced on June 15, 2010 following regulatory review. A novel benefit of our approach is providing the output of threat analysis as a database, which can be used by device manufacturer and hospital customer to model changes in risk profile as technology and customer environment evolve.
Appendix A – The Practical Threat Analysis methodology (PTA)

Introduction
The objective of any threat analysis process is to understand where are the security issues, how much they can impact the organization - and determine the most cost-effective way of mitigating risk; whether from viruses and phishing attacks, disclosure of patient health information or theft of sensitive information that impacts national security.

A threat analysis process is required for complex systems that integrate diverse infrastructures and technologies; an integrated system of custom hardware, Windows operating systems, and Java and third-party libraries is not an unusual example. From a software security perspective, threat analysis is best used throughout the entire SDLC (Software development life cycle) starting with the design - which typically accounts for 50% of the security vulnerabilities.

Building blocks
The Practical Threat Analysis process rests on 4 intuitive building blocks:

Vulnerability - a weakness, limitation or a defect in one or more of the system's elements that can be exploited to disrupt the normal functionality of the system. The weakness or defect may be either in specific areas of the system, its layout, its users, operators, and/or in its associated regulations, operational and business procedures.

Countermeasure - a process, procedure or technology for mitigating a specific vulnerability. A specific countermeasure may mitigate several different vulnerabilities. Countermeasures are often called "controls" or "safeguards".

Asset - information, capability, an advantage, a feature, a financial or a technical resource that may be damaged lost or disrupted. The damage to an asset may affect the normal functionality of the system as well as of the individuals and/or organizations involved with the system.

Threat - a specific scenario of a sequence of actions that exploits a set of vulnerabilities and may cause damage to one or more of the system’s assets.

For more information – see: http://www.software.co.il/application-security/26-practical-threat-analysis-of-complex-systems.html
For a case study and sample reports – see data security case study and download the data security report to the management.
Medical Device threat analysis – rev. 1.7

Practical Threat Analysis Data Model

Asset has **value**, fixed over time or recurring.

Customer product pricing

Threat

**exploits** vulnerabilities & **damages** assets.

An insider sees **special pricing and blogs about** a deal. Customers discover pricing on Google blog alerts and demand same prices. Company **loses face** and **revenue**.

Vulnerability is a state of weakness **mitigated** by a countermeasure.

Any user in the office **can view customer pricing** in office Intranet.

Countermeasure has **cost**, fixed over time or recurring

**Hide pricing** in sales report screens on office Intranet.