A Comprehensive Security Assessment Toolkit for HealthCare Systems

Subrata Acharya Dr.
*Towson University*, sachaarya@towson.edu

Michael Terry
*Towson University*, mterry1@students.towson.edu

Ogbeide Derrick Oigiagbe
*Towson University*, ooigia1@students.towson.edu

Follow this and additional works at: [https://scholarworks.wm.edu/caaurj](https://scholarworks.wm.edu/caaurj)

**Recommended Citation**
Available at: [https://scholarworks.wm.edu/caaurj/vol4/iss1/6](https://scholarworks.wm.edu/caaurj/vol4/iss1/6)

This Article is brought to you for free and open access by the Journals at W&M ScholarWorks. It has been accepted for inclusion in Colonial Academic Alliance Undergraduate Research Journal by an authorized editor of W&M ScholarWorks. For more information, please contact scholarworks@wm.edu.
1. Introduction

In the year 1996, The Health Insurance Portability and Accountability Act (HIPAA) established standards for how individually identifiable health information is received, maintained, and stored in electronic form for any given healthcare organization. In the year 2009, the Health Information Technology for Economic and Clinical Health (HITECH) Act also set standards, implementation specifications and other criteria for the maintenance of Electronic Health Records (EHR) [2]. At the time of HIPAA’s creation, the healthcare industry was transitioning from a paper based information system to an electronic information system. Health records were being transformed into a digital format, and the industry began to rely heavily on the use of electronic systems to conduct business [3]. Prior to HIPAA, there were no security or privacy rules defined for the protection of a patient’s electronic healthcare information. HIPAA created two important guidelines. The first rule, ‘the Security Rule’, warranted that entities (health plans, healthcare clearinghouses, healthcare providers who transmit any health information in electronic form) must ensure the confidentiality, integrity, and availability of all electronic protected health information (e-PHI) they create, receive, maintain or transmit. This includes identifying and protecting against reasonably anticipated threats to the security (or integrity) of the information, and protecting against reasonably anticipated, impermissible uses or disclosure while ensuring compliance by their workforce [4]. The second element created, ‘the Privacy Rule,’ assuring that the information maintained within one’s electronic health records are kept secret while flowing from one healthcare entity to another. More specifically, if a patient’s electronic health records are disclosed, the individually identifiable health information (i.e. a patient’s past, present or future physical or mental health) should not be associated with the individual [5].

Despite these efforts by the federal government, a survey by the Ponemon Institute in 2010-2011 found that there has been a severe lack of security countermeasures [6]. The Ponemon Institute found 60% of healthcare providers had severe critical security breaches in the past two years. Moreover, the average breach cost healthcare entities over $2 million each. The United States Department of Health and Human Services [7, 8] also discusses and identifies the increased trend of security breaches in this area. Furthermore, half of the healthcare entities that were interviewed revealed they had little faith in their information technology (IT) personnel to protect patient’s data. This report suggests an alarming situation and needs immediate attention by the industry. Efforts to learn from other IT fields on how to provide security and privacy measures have proven that there is a lack of completeness in the tool sets available to the healthcare industry.
Therefore, we aim to achieve four goals while developing our tool set:

- **Comprehensiveness**: The tool set needs to be comprehensive enough to scan the entire domain of an information system. It not only needs to detect misconfiguration on servers (i.e. database servers, web servers, email servers, etc.) and network devices, and find missing patches on hosts, but it must also have the capability to detect vulnerabilities within new and upcoming services such as VOIP (e.g. eavesdropping) and virtual infrastructures (e.g. misconfigured virtual networks and virtual hosts).

- **Automation**: Any comprehensive assessments take time to complete, and if an IT professional has to do one manually, it can take substantial time to complete the task. Moreover, a manual, comprehensive assessment of a Healthcare Information System (HIS) can be long and tedious. IT professionals performing the assessment may skip steps in the assessment to save time and resources. By automating a comprehensive assessment, healthcare IT professionals will save time and resources. Likewise, they will be confident in their assessment results.

- **Health IT Compliance**: Our third objective is ensuring our derived assessment tool set will provide Health IT compliance in regards to ‘The Security’ and ‘The Privacy’ rule.

- **Mitigation Strategies**: Finally, we aim to achieve mitigation strategies with complete assessments. Our mitigation technique will differ from current techniques in that our strategy will be real-time and automated rather than offline and delayed.

By integrating the above goals into our derived tool set, we aim to address the security and privacy concerns of healthcare entities and to restore faith in their electronic information systems. Hence to achieve the goals of this research we will do the following:

- We will survey commercial and open source tools in the areas of networks, databases, applications, and infrastructure security. We will then identify the best-fit open source tools to be integrated into a tool set.

- When the best-fit open source tools have been identified, we will then build the comprehensive, assessment tool set and modify it to meet federal compliance of security and privacy regulations in the healthcare industry.
After all the applications have been installed, we will write scripts to automate the vulnerability assessment and penetration test for a typical HIS.

Finally, after the vulnerability and assessments have been completed, we will provide a real-time, automated analysis of the logs, and provide guidance on how to mitigate risk within the HIS.

As the healthcare industry is rapidly moving towards electronic based information exchange, it has become mandatory for it to be compliant with the HIPAA, the HITECH Act, and other federal regulations and standards. This change towards electronic based information systems also requires the industry to maintain the privacy of patient information. Today’s healthcare industry has experienced numerous instances of breach of information and the loss or compromise of critical patient data. According to the Ponemon Research Study, 92% of all healthcare institutions report they have experienced data breaches in recent years. If entities within this industry do not address this problem sufficiently, then it might lead to severe federal penalties along with patient privacy compromises and patient dissatisfaction.

The remaining parts of this paper are split into eight more sections. First, we will discuss the research that is already being done in this field. Then we present our survey results from commercial vendors and open source organizations. In section four we will detail a risk assessment framework for the healthcare industry. Then we outline our evaluation environment and how we plan to evaluate our open source solution against a competitive commercial product. We will then present how our analysis and detection engine operates and how mitigation strategies are recommended. Finally, we will present the conclusion and future directions for this research.

2. Background

Over the years there has been a considerable growth in the availability of automated vulnerability assessment solutions to assess an organization’s information system. In fact, there are several solutions from commercial vendors that provide automated vulnerability and penetration testing software. Yet, none of these solutions offer industry specific compliance testing right out of the box. Most vendors require organizations to buy plug-ins in addition to their product to automate industry specific compliance testing.
Even though the area of automated assessment is quite mature, there is still a dearth of automated approached for assessment. However, two most current open source vulnerability assessments do meet some of the goals of our research. Those tools are OpenVAS 4 [10] from the OpenVAS organization and Fast-Track [15] from Offensive Security Ltd. Relating OpenVAS 4 to our research; we see that it has a comprehensive scanner used to inspect remote hosts and attempts to list all the vulnerabilities and common misconfigurations that affect the host. This tool can be used to comprehensively scan a network as well as server configurations. Yet OpenVAS 4 is not fully automated and does not scan a HIS right out of the box. A healthcare IT professional must configure the scanner to scan a HIS for HIPAA compliance. If the scanner is configured correctly (to scan for HIPAA compliance), then the OpenVAS 4 tool will ensure a HIS is acting in accordance with the Privacy and Security Rules set forth by HIPAA. Additionally, it can be configured to list the vulnerabilities associated with each device/service that it scans, but the tool cannot provide any mitigation strategies for the vulnerabilities found. Likewise, the tool Fast-Track is an automated penetration suite designed to scan and penetrate databases, networks, infrastructures, and applications on an information system's domain. However, this tool's scanning capabilities is not as comprehensive as OpenVAS 4—it cannot scan wireless networks or VOIP infrastructures. Similar to OpenVas 4, the tool does not satisfy the Privacy and Security Rules defined by HIPAA. Additionally, Fast-Track does not list vulnerabilities detected or provide mitigation strategies after it has finished its assessment.

There are numerous policy guidelines on how to keep healthcare entities HIPAA compliant, but there is a lack of implementation solutions (tool sets) on how to provide compliance in practice. Additionally, there is little work, if any, in using an automated assessment tool set for finding vulnerabilities in a typical HIS and provide mitigation strategies.

3. Survey of Tools

In this section, we identify the tools and their capabilities in regards to assessing the four security areas of a HIS. These areas include Database Security, Network Security, Infrastructure Security, and Application Security. First, Database Security ensures that the designed tool set will be able to crawl, and/or use an SQL Injection attack on an EHR database to obtain information on patients (i.e. billing information, social security numbers, type of Healthcare insurance the patient has, etc.). The second area, Network Security means the tool set must be able to scan IP ranges of devices on a network and try to identify the operating system, manufacturer and model. Furthermore, it needs to provide the results from
the scanner(s) with minimal false positives, and offer exploits available from the scanner results. The third security area, *Infrastructure Security* implies that the tool set needs to deliver client side endpoint attacks to test the infrastructure (operating systems security and services reliability). Finally, for the fourth area, *Application Security*, the tool set must offer application testing in regards to endpoint attacks on applications (buffer overflows, cross site scripting attacks, etc.). Additionally, the derived tool set needs to be comprehensive enough to provide the capabilities for various types of assessments.

The first assessment, *External Network Vulnerability Assessment Testing*, involves finding unknown vulnerabilities from outside a HIS’s network through poor network design and backdoors. One should not be able to access private areas of the HIS from outside the network. If there is a vulnerability detected, the tools set will try to penetrate the network in order to prove data can be accessed. The second assessment, *Internal Network Vulnerability: Assessment Testing* deals with finding unknown vulnerabilities from inside a HIS’s network. The tool set will assess the electronic assets (EHR database, application servers, file servers, web server, etc.) of the HIS, and if there is a vulnerability detected, the tool set will try to exploit the vulnerability in order to prove data can be accessed or a service can be degraded.

The next assessment, *Web Application Assessment Testing*, ensures the tool set will test web applications by simulating attacks to gather information on their flaws and vulnerabilities. More specifically, the tool set will need to test for the following types of vulnerabilities (but not limited to): cross-site scripting, SQL injection, input validation, and buffer overflows. In the fourth assessment, *Dial-In/RAS Security Testing*, the task includes testing dial-in/remote access entry point connections that employees or healthcare partners use and identifying exploits that can be used against the system. In the following assessment, *DMZ or Network Architecture Designs/Reviews*, the tool set will verify data stores are not located publicly or in the DMZ. Furthermore, an adversary should not be able to bypass the firewall by piggybacking off a connection from a mail server located outside the internal network.

*Wireless Network Assessment Testing* incorporates the capability for the tool set to be comprehensive enough to detect misconfigurations in wireless access points and exploit them if vulnerabilities arise. Additionally, if enough packets are analyzed, the tool set will try to crack the WEP and WPA-PSK keys. In *Virtual Infrastructure Security Assessments* we see the tool set identifying and mitigating virtual infrastructure risk by checking the configurations of virtual machines, networks, and storage mediums. For the next capability, *Server Configuration*
Reviews, the tool set will review common network service misconfigurations, local password policies, file shares, and file share permissions. **Firewall and Router Configuration Reviews** ensures the tool set will check a HIS’s network perimeter to ensure each firewall is properly configured in order to allow verified network traffic into the HIS's network. In addition, **VPN Configuration Reviews** will verify a VPN is configured correctly and there are no vulnerable entries into the network. A hacker should not be able to view sensitive information flowing from location to another. Finally, for **Voice over IP Assessments**, the tool set will be analyzing a HIS's network to see if it is vulnerable to the following vulnerabilities: SIP-based phone call on eavesdropping, and SIP-based phone call hijacking.

The results from our commercial tools surveys are detailed in the table 1-1 and 1-2. Included in the table is how each vendor’s solution’s capabilities contribute to the goal of our research. In table 1-1, the commercial tools survey, we could infer that eEye Digital Security’s Retina Enterprise Edition is the best choice. We found Core Impact Pro was not a vulnerability scanner, but rather an automated penetration tester solution that records successful and unsuccessful attacks and generates vulnerability reports based on those penetration tests. Therefore, Core Impact Pro may miss a vulnerability that a vulnerability scanner may be able to detect. In addition, we discovered Retina Enterprise Edition could also perform penetration attacks using a built in hacker module [9]. Moreover, Retina and Core Impact Pro could be configured to assess for HIPAA compliance (Security & Privacy Rules); however, each vendor charges extra fees to provide this functionality.

As far as the other tools in this survey, we found that they were not as comprehensive as Retina or Core Impact Pro. WebInspect is geared strictly to finding vulnerabilities in web applications; Foreground Security did not offer any products, but rather vulnerability assessment and penetration testing services. As far as SAINT, information was rather scarce. We did not find as much information on SAINT as with Retina. Comparing the two solutions we were more confident with Retina's Enterprise solution. Additionally, Retina's solution was the highest rated network vulnerability assessment scanners in the industry. Furthermore, our results from the open source survey are listed in table 1-2.
<table>
<thead>
<tr>
<th>Tool</th>
<th>Database</th>
<th>Network</th>
<th>Infrastructure</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>eEye Digital Network Scanner</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retina Enterprise Edition</td>
<td></td>
<td>Database scanning</td>
<td></td>
<td>Web application</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External network vulnerability</td>
<td></td>
<td>scanning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assessment</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wireless network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOIP Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Firewall</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reviews</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.eeye.com/home">http://www.eeye.com/home</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td><strong>HP WebInspect</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Database penetration testing</td>
<td></td>
<td>Web application</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>scanning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>penetration testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>penetration testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wireless network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOIP Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designs</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www8.hp.com/">http://www8.hp.com/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td><strong>Core Impact Enterprise Edition</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Database penetration testing</td>
<td></td>
<td>Web application</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>scanning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>penetration testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>penetration testing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wireless network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOIP Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designs</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.coresecurity.com/">http://www.coresecurity.com/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
</tr>
<tr>
<td><strong>Nessus</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Database scanning</td>
<td></td>
<td>Web Application</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scanning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>External network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>vulnerability</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assessment</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wireless network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOIP Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Designs</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Firewall</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reviews</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.tenable.com/">http://www.tenable.com/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Acharya et al.: A Comprehensive Security Assessment Toolkit for HealthCare Systems

Published by W&M ScholarWorks, 2015
## Tool Database Network Infrastructure Applications

<table>
<thead>
<tr>
<th>Tool</th>
<th>Database</th>
<th>Network</th>
<th>Infrastructure</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saint Enterprise Edition</td>
<td>Database penetration testing</td>
<td>External network vulnerability Assessment</td>
<td>Server configuration reviews</td>
<td>Web application penetration testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internal network Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wireless network assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>VOIP Assessments</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network architecture Design</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1-1 Commercial Tools Survey

<table>
<thead>
<tr>
<th>Database</th>
<th>Network</th>
<th>Infrastructure</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wapiti</td>
<td>OpenVAS 4</td>
<td>OpenVAS 4</td>
<td>OpenVAS 4</td>
</tr>
</tbody>
</table>

Wapiti is an open source and web-based tool that scans the web pages of the deployed web applications, looking for scripts and forms where it can inject data. It is built with Python and can detect:

1) File handling errors
2) Database, XSS, LDAP and CRLF injections.
3) Command execution detection.

This tool can be used to scan the databases as well scan web applications for vulnerabilities.

http://wapiti.sourceforge.net/README

OpenVAS 4

OpenVAS Security Scanner is a security auditing tool made up of two parts: a scanner and a client. The scanner, openvassd is in charge of the attacks, while the client OpenVAS interfaces with the user. The scanner, the most critical part, inspects remote hosts and attempts to list all the vulnerabilities and common misconfigurations that affect the host. This tool can be used to scan the network as well as the server configurations (infrastructure) of an organizations health information system.

http://www.openvas.org/src-doc.html

SQLMap

SQLMap is an open source penetration testing tool that automates the process of detecting and exploiting SQL

Kismet

Kismet is an 802.11 layer2 wireless network detector, sniffer, and intrusion detection system. Kismet will work with any wireless card which supports raw

×

Wapiti
injection flaws and taking over of back-end database servers. It comes with a broad range of features, from database fingerprinting to fetching data from the DB and even accessing the underlying file system and executing OS commands via out-of-band connections. Needless to say this will aid in fingerprinting and exploiting a database.

http://svn.sqlmap.org/sqlmap/trunk/sqlmap/

**Wfuzz** [19]

Wfuzz is a tool for brute forcing Web Applications, it can be used for finding resources not linked (directories, servlets, scripts, etc), bruteforcing GET and POST parameters for different kinds of injections (SQL, XSS, LDAP, etc.), bruteforcing form parameters (user/password), fuzzing, and more.

http://www.edge-security.com/wfuzz.php
http://code.google.com/p/wfuzz/downloads/list

**Aircrack**

Aircrack is a suite of tools for 802.11a/b/g WEP and WPA cracking. It implements the best known cracking algorithms to recover wireless keys once enough encrypted packets have been gathered. The suite comprises over a dozen discrete tools, including airodump (an 802.11 packet capture program), aireplay (an 802.11 packet injection program), aircrack (static WEP and WPA-PSK cracking), and airdecap (decrypts WEP/WPA capture files).

http://www.aircrack-ng.org/

<table>
<thead>
<tr>
<th>Wfuzz</th>
<th>Aircrack</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="https://example.com/xmarkbld" alt="x" /></td>
<td><img src="https://example.com/xmarkbld" alt="x" /></td>
</tr>
</tbody>
</table>

Wfuzz

---

Acharya et al.: A Comprehensive Security Assessment Toolkit for HealthCare Systems

Published by W&M ScholarWorks, 2015
<table>
<thead>
<tr>
<th>Tool (Fast-Track)</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSSQL Brutter</td>
<td>MSSQL Bruter is a Fast-track tool that tries to identify SQL servers with weak &quot;sa&quot; passwords in order to inject payloads into the system. The tool implements this task by brute forcing the SQL server password. <a href="http://www.offensive-security.com/metasploit-unleashed/MSSQL_Bruter">http://www.offensive-security.com/metasploit-unleashed/MSSQL_Bruter</a></td>
<td></td>
</tr>
<tr>
<td>UCSniff</td>
<td>UCSniff is a VoIP &amp; IP Video Security Assessment tool that integrates existing open source software into several useful features, allowing VoIP and IP Video owners and security professionals to rapidly test for the threat of unauthorized VoIP and Video Eavesdropping. UCSniff supports Arp poisoning, VLAN Hopping, VLAN Discovery via CDP, it has a sniffer capabilities and more. UCSniff can operate in 2 modes: 1) Monitor mode – Should be used on a shared media where the IP phones connected to i.e. a HUB, wireless access point, it can be also be used in a switched environment by setting up a SPAN sessions on a Cisco switch. 2) Man in the middle mode – This mode has 2 additional modes which are Learning Mode Targeted Mode <a href="http://sourceforge.net/projects/ucsniff/files/">http://sourceforge.net/projects/ucsniff/files/</a></td>
<td></td>
</tr>
<tr>
<td>Skipfish [20]</td>
<td>Skipfish is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The resulting map is then annotated with the output from a number of active (but hopefully non-disruptive) security checks. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments. Source: <a href="http://code.google.com/p/skipfish/">http://code.google.com/p/skipfish/</a></td>
<td></td>
</tr>
<tr>
<td>MSSQL Injection</td>
<td>A tool that uses SQL techniques in order to ultimately gain full unrestricted access to the underlying system. <a href="http://www.offensivesecurity.com/metasploit-unleashed/">http://www.offensivesecurity.com/metasploit-unleashed/</a></td>
<td></td>
</tr>
</tbody>
</table>

Table 1-2 Open Source Tools Survey
From the open source survey table above, we can infer the best choice tools for the open source toolkit would be:

- **OpenVas 4**: A vulnerability analysis tool that can scan multiple targets concurrently with its supported 20,000 vulnerability test. In addition, the tool consolidates many tools into its scanner (i.e. Nikto, Nmap, and w3af), expanding its reporting capabilities. [10]

- **Kismet**: A wireless network detector, sniffer, and intrusion detection system that can sniff 802.11b, 802.11a, 802.11g, and 802.11n traffic. Kismet works by passively collecting packets; which can lead to detecting hidden networks over time. [11]

- **Aircrack**: A tool designed to assess the security of a wireless network using various WEP and WPA cracking algorithms. Once this tool has captured enough packets from a wireless network, it can begin analyzing those packets and try to break the wireless network’s encryption methods [12].

- **UCSniff**: An application to rapidly test for the threat of unauthorized VoIP and Video Eavesdropping. This tool was developed for current and next generation VOIP infrastructures, so it will aid healthcare entities keep their HIS HIPAA compliant. [13]

- **SQLMAP**: A tool designed to automate the process of detecting and exploiting SQL injection flaws and taking over database servers. Not only can the tool fingerprint databases and fetch data from them, but it can also be used to execute command on the operating system. [14]

- **Fast-Track**: An automated penetration suite designed to scan and penetrate databases, networks, infrastructure, and applications of an information system's network. Some of its various tools include the Nmap scripting engine, SQL Pawnage, and MSSQL injection. [15]

- **Wapiti**: A tool designed to audit the security of web applications through 'black-box' scans looking for scripts and forms where it can inject data. Some of its capabilities include Cross Site Scripting injection, LDAP injections, and file handling errors. [16]
These tools working cohesively, in a toolkit, will address the concerns of developing a toolkit that will be comprehensive enough to ensure a Health Information System (HIS) stays HIPAA compliant.

4. Risk Assessment Framework

The purpose of this section is to offer a framework for describing how the assessment engine (the derived tool set) interacts with our data set (a replica healthcare information system).

![Risk Assessment Framework for a HIS](image)

Figure 1: Risk Assessment Framework for a HIS

The framework is illustrated in figure 1. After the assessment engines have received data from the HIS components, it pipes their data to the master assessment engine. Here at the master assessment engine, an analysis and detection engine analyzes the data to detect anomalies and vulnerabilities within the HIS. Once the analysis is complete and there are vulnerabilities detected, it will send its analysis to the mitigation engine. The mitigation engine then develops mitigation strategies from the analysis and sends its recommendations.
back to the assessment engine. Finally, the assessment engine presents assessments to the user in the form of reports.

5. Experimental Setup

The environment we chose to implement our open source tool set was Backtrack 5 configured as a Virtual Machine (VM). We chose this environment simply because six of the seven tools were already installed in this Linux operating system. Once we configured the VM, we installed Nessus HomeFeed (version 5). In this way any custom scan polices will serve as the baseline for testing our open source solution. Afterwards, when we completed and tested our open source tool set, we could then compare the results with the Nessus Homefeed results. Additionally, to develop the open source tool set, we subdivided the HIS domain into 4 areas. The tool set focuses on the 1) network, 2) databases, 3) applications, and 4) infrastructure of a HIS.

In the following 4 subsections, the paper discusses how each open source tool surveyed will enable the derived toolkit to achieve its objective. Together, these tools integrated into a toolkit have the potential to provide comprehensive, automated assessments for any healthcare organization.

- **Network**: In order to ensure the toolkit was comprehensive enough to provide detailed network assessments, four tools were selected to determine the
associated risk of exploitation to a HIS’s network. The first tool, OpenVAS 4, is a network scanner that audits network host and list the vulnerabilities and common misconfigurations that affect the host. The second tool Kismet, is a tool that was selected based on its capabilities to detect and sniff out wireless networks. However, the tool did not provide Wireless network penetration testing; Therefore, Aircrack was needed to deliver WEP and WPA penetration testing. Lastly, the toolkit needed to test for unauthorized VOIP eavesdropping. To provide this type of assessment, UCSniff was preferred as it can quickly test for unauthorized VOIP eavesdropping.

- **Database:** To assess the security of a HIS’s database, two tools were chosen, SQLMap and Fast-Track. Both tools provide penetration testing against databases. However, each tool excelled at one area of database scanning and penetration testing whereas the other tool excelled in another area of database scanning and penetration testing. First, SQLMAP excels at fingerprinting a database, a feature that Fast-Track lacks. The tool can also be a means of taking over back-end database servers and even access the underlying OS. On the other hand, Fast-track is an automated penetration suite that uses Metasploit to enhance its library of attacks against a given database. By pipelining the fingerprinting results of SQLMap into Fast-Track, we can fully assess the security of a database.

- **Applications:** The tool chosen to scan for web applications within a HIS was Wapiti. This tool was chosen because it is able to detect the most vulnerability (i.e. Cross-Site Scripting, LDAP injection) and act like a fuzzer to inject payloads to see if scripts within web applications are vulnerable.

- **Infrastructure:** To assess the configuration reviews of the servers and desktop host on a HIS, there was only one open source option available - OpenVAS 4. Using server configuration reviews from the scan, we can pipeline the results into Fast-Track and create automated penetration test against a HIS.

Figure 2 illustrates the synergy of the tools chosen to be integrated into the tool set. Together, these tools make up the framework for the assessment engine. As one can see, the tools within the tool set are in constant communication with the apparatuses of the HIS. If the assessment engine detects malicious activity within the HIS, it will signal the apparatus of the HIS to send data back to it at shorter time intervals.
6. Evaluation and Inferences

Dataset: As hinted throughout this paper, our data set will consist of electronic health data. The data is both real world and emulated data set from Google health data.

![Figure 3: Google Health Record](image-url)
Generally speaking, a personal health record contains physical and/or mental information about a patient. To illustrate an electronic personal health record, figure 3 is a sample electronic Google personal health record from Google’s healthcare service [17]. This service allowed individuals to create a profile derived from their individual health records so that they can see the risk and benefits of the current treatment they are receiving. In this record, we see individual identifiable information such as the patient’s name, date of birth, ethnicity, and blood type (information that needs to be protected in order to prevent unlawful disclosures). In addition, there are five additional fields (along with their attributes) within the record. The first field, the Wellness field, contains the patient’s height and weight. The second field, the Problems field, holds information related to illnesses the patient has been diagnosed with to date. The third field, the Allergies field, lists the patient’s allergies. The next field, the Procedures field, details the procedures the patient has undergone. Finally, the last field, the Immunizations field, lists the immunizations the patient has received. By the same token, we will test our data on real world electronic health record data from a HIMMS healthcare organization.

In addition to EHRs, the other major data set used in our research will be the firewall. More specifically, the configuration of the firewall is important because it is the first line of defense against hackers. To clarify, the firewall sits between the Internet and internal healthcare information system, and its main job is to filter connections based on policies set by the administrators. If a connection is allowed by the policy, the firewall will allow the connection into the internal HIS. If not (because of security reasons), then the firewall will drop the connection. Figure 4 illustrates a firewall policy created by an IT professional. In general, a firewall policy rule works by analyzing the source of an incoming connection, the destination of the incoming connection, and the type of service running on that connection. Once the firewall has this information, it checks the information against the rules set forth by the administrator. If the connection meets the requirements, then it is allowed into the firewall. If not, then the connection is denied. As an example, we can take a look at the first row of the firewall configuration policy illustration. The host 104.4.51.22 is allowed to make a samba connection to host 10.0.3.78. Likewise, the firewall uses this same process to deny connections.
<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>104.4.51.22</td>
<td>10.0.3.78</td>
<td>Samba</td>
<td>Allow</td>
</tr>
<tr>
<td>206.8.7.88</td>
<td>10.0.3.33</td>
<td>Ssh</td>
<td>Allow</td>
</tr>
<tr>
<td>194.3.6.2</td>
<td>10.0.3.45</td>
<td>ldap</td>
<td>Deny</td>
</tr>
<tr>
<td>98.5.7.1</td>
<td>10.0.3.91</td>
<td>dns</td>
<td>Allow</td>
</tr>
<tr>
<td>23.1.5.87</td>
<td>10.0.3.178</td>
<td>ntp</td>
<td>Deny</td>
</tr>
</tbody>
</table>

Figure 4: Firewall Configuration Policy

**Results: Real world HIMSS 6 Healthcare Dataset:**

Our results are based on emulated assessments of a HIS network and an EHR database. More specifically, The EHR database we are assessing contains 20,000 electronic medical records (EMR) from the HIMMS 6 Healthcare practice.

Once we finished assessing the network and EMR database, we divided our results into two categories: privacy vulnerabilities and security vulnerabilities. Then we compared the vulnerabilities detected by the derived tool set against the actual number of vulnerabilities in the HIS. For example, there are five actual vulnerabilities in a HIS we are assessing. Our tool set only detected four vulnerabilities; therefore, our tool set was 80% effective. If the tool set detected all five then it is 100% effective. If the tool set detected all five vulnerabilities plus one, then the tool set detected a false positive. In this case, we will disregard all false positives since our results only reflect true positives. In Figure 5, we see
the HIMMS 6 EMR assessment. There were a total of four assessments to detect vulnerabilities against the Privacy Rule, and four assessments to assess the vulnerabilities against the Security Rule. Once we completed the four assessments within each category, we calculated the averages to find the success rate of the tool set. The average success rate for detecting security vulnerabilities against the EMR database was 96%. The average success rate for detecting privacy vulnerabilities against the EMR database was 98%. In Figure 6, we present our HIMMS 6 network assessment results. The average success rate of security vulnerabilities detected by the tool set was 98.75%. In addition, the average percentage of privacy vulnerabilities detected by the tool set was 98.50%.

![HIMMS 6 Network Assessment Results](image)

**Figure 6: HIMMS 7 Network Assessment**

**Results: Emulated Google Health Dataset:**

Our results are based on assessments of a D-Link DIR-655 Xtreme N Gigabit Router (IP address 10.0.0.1), and an ASP.NET web application database (IP address 10.0.0.60). More specifically, the database records we are assessing contain sample information about movies. Additionally, once we finished assessing the network and database, we compared our results to the Nessus baseline scans. We found the baseline scans for the D-link router (IP address 10.0.0.1) found 2 vulnerabilities rated high risk on UDP port 2003, one vulnerability rated medium risk on UDP port 53, and 23 vulnerabilities rated as low risk. The derived assessment scan detected one high risk vulnerability, one medium risk, and 12 vulnerabilities rated low risk. Analyzing and comparing these results, we see the toolkit did not detect the two high risk vulnerabilities.
found in the baseline scans, but it did detect a high risk vulnerability related to
SMNP Agent responding to community names, and one medium risk
vulnerability related to denial of service attacks against TCP services. However,
the medium risk vulnerability detected by the derived toolkit was not the same
medium risk vulnerability detected by the baseline assessment.

For the host (IP 10.0.0.60) running the ASP.NET application, we found that
the results were very interesting. The baseline assessment detected three medium
risk vulnerabilities. When we examine the medium risk vulnerabilities further, we
find two medium risk vulnerabilities are associated with the web server certificate
being a self-signed certificate. The X.509 certificate should be signed from a
known trusted public authority. This becomes a concern if the web server was
actually placed on the Internet, then an attacker could initiate a man in the middle
attack by breaking the chain of certificates on the server. The other medium risk
vulnerability found was due to SMB signaling being disabled. This vulnerability
could be exploited through a man-in-the-middle attack against the SMB server. In
comparison, the derived assessment toolkit did not detect the three vulnerabilities
the baseline assessment scan detected. The toolkit found two high risk
vulnerabilities in the ASP.NET web application. One high risk vulnerability
detected was because of a SQL 5.x Unspecified Buffer overflow vulnerability.
This was due to the fact that the ASP.NET application did not perform boundary
checks on user supplied data. According to the data logs, failed exploits can cause
a denial of service on the database. The second high risk vulnerability found was
for a MySQL Multiple Vulnerabilities vulnerability. This could be the direct result
of the buffer overflow detected by the other high risk. As one can see the two
assessments contrast greatly. Upon further research, we found that the baseline
assessment software needs a plugin license to detect vulnerabilities within
databases. The detailed screenshots are in Appendix A.

7. Log Analysis and Mitigating Strategies

Log Analysis Detection:

Because of the ever changing environment electronic information systems are
being utilized in, continuous assessments and adjustments are needed to keep
those systems secured and compliant with regulatory laws. This is very true in
terms of healthcare information systems. In order to provide mitigation to the
vulnerabilities found during scans, we will conduct a review of the logs recorded
during assessments. For this we will use four techniques to review the logs.
Knowledge based: A technique where the log assessment engine will be conducting an analysis on the tool set assessment data by comparing it to a stored knowledge base (a file containing the baseline assessment scans of a HIS undergoing HIPAA compliance testing) within its engine. It will then send its analysis to the mitigation engine for recommendations.

Anomaly based: A technique that involves comparing the results from the tool set assessment data with previously recorded assessment data of the HIS. From this comparison, this analysis will detect anomalies (e.g. a host is scanning the network searching for vulnerable host, a host is scanning ports of other host to find what services are running on them, host attacking another host) found within the HIS and send its analysis to the mitigation engine. From these analyses the mitigation engine will provide a means to mitigate the vulnerabilities found.

Reputation based: This technique involves analyzing the behavior of each component of an HIS to determine the trustworthiness within the domain of the HIS. To determine the trustworthiness of a host, the analysis and detection engine will form its own opinion about the host in question, and incorporate how other host’s view it’s trustworthiness to form a selected host’s reputation.

Hybrid Based The hybrid technique embraces the previous three techniques processes for forming analysis on a host. Once it has finished its analysis, it sends the analysis to the mitigation engine. Because of this technique’s complexity, it requires more time and resources then the previous techniques.

Mitigating Strategies

To illustrate how mitigation strategies work, we will use the example of an unencrypted database found within a HIS by our tool set. In this scenario, the analysis and detection engine has reviewed the logs and discovered the tool set 1) successfully exploited a database containing EHRs, and 2) the toolkit can search through the database without limitations. Since the EHRs on the database are unencrypted, any hacker would be able to exploit the data. The analysis and detection assessment engine would identify this vulnerability and send its analysis to the mitigation engine. The mitigation engine would then recommend using a data encryption solution such that even though the hacker is able to gain access to the data, it is not able to view the encrypted information without possessing the correct decryption keys. One such encryption solution is the open source framework Charm-Crypto [18]. Charm is a framework for rapidly prototyping advanced cryptosystems. Its library includes public key encryption schemes,
identity-based encryption schemes, attribute-based encryption schemes, digital signatures, privacy-preserving signatures, commitment schemes, zero-knowledge proofs, and interactive protocols such as anonymous credential and oblivious transfer schemes. To conclude, once the tool set has assessed a HIS, we can begin to provide solutions to mitigate the vulnerabilities.

We reviewed the logs, and provided mitigation strategies to address the high and medium risk vulnerabilities that we detected. Our mitigation strategies were based on recommendation from online sources such as us-cert.gov, and owasp.org. First, examining the router of the network (IP address 10.0.0.1), the 2 high risk vulnerabilities associated with the D-link daemon, it is recommended that one implement authentication methods to mitigate the Click ‘n Connect daemon to disallow attackers from gaining control of server functions via the D-link Daemon. Secondly, for the 1 medium risk vulnerability found on the router from DNS snooping attacks on the router, we recommend re-configuring of the DNS to stop this kind of snooping activity. When we took a look at the 1 medium risk from the derived assessment toolkit results, the TCP Sequence Number Approximation Reset Denial of Service Vulnerability, we found that it is recommended that one either implement IPSec (IP Security) to encrypt traffic and obscure TCP information available to the attacker, implement ingress and egress filtering to expected addresses, or implement TCP MD5 signature option to verify and checksum TCP packet carrying BGP data. Finally, regarding the host hosting the ASP.NET application, the 2 high risk vulnerabilities (buffer overflow), we recommend implementing Integrating boundary checking in the test application.

8. Conclusions and Future Research Directions

In conclusion, our research achieves the task of fully assessing a healthcare information system domain. We accomplished this by first comprehensively assessing the databases, networks, applications, and infrastructure within the HIS domain, and then automating the comprehensiveness assessments to ensure time and resource efficiency. Furthermore, automating the comprehensive assessment ensures no steps would be skipped and that the assessment of the HIS is HIPAA compliant. Finally, in order for a healthcare entity to maintain its HIPAA compliance, we see the mitigation engine recommending solutions based on the data it receives from the analysis and detection engine. Our research was useful in that it exposed the issues facing the healthcare industry not widely publicized and makes us think about the entirety of the information healthcare entities possess that are not entirely secure within their domain. As future research we would like to test our proposed toolkit on different real world data sets. Additionally, one can use different EHR formats to test the assessment engine against. Also, we
would like to develop additional mitigation strategies to ensure security of EHRs in storage, access and transmission. These efforts will in turn enable the maintenance of compliance in the HIS.

References


Appendix A

Appendix Figure 1: Configuring the Database Assessment Scanner

Appendix Figure 2: Configuring the Infrastructure Scans
Appendix Figure 3: Configuring the Network Scans

Appendix Figure 4: Configuring the Web Application Scans
Appendix Figure 5: Baseline Assessment of 10.0.0.0

Appendix Figure 6: Overview Baseline Assessment of 10.0.0.1

Appendix Figure 7: Toolkit Assessment Results of 10.0.0.1
Appendix Figure 8: Overview Baseline Assessment of 10.0.0.60

Appendix Figure 9: Toolkit Assessment Results of 10.0.0.1
(High Risk Vulnerabilities)