Edith Cowan University
Research Online

Australian Information Security Management Conference

Security Research Institute Conferences

2007

The Need for an Investigation into Possible Security Threats Associated with SQL Based

Lee Heinke
Edith Cowan University

Originally published in the Proceedings of 5th Australian Information Security Management Conference, Edith Cowan University, Perth Western Australia, December 4th 2007
This Conference Proceeding is posted at Research Online.
http://ro.ecu.edu.au/ism/29

The Need for an Investigation into Possible Security Threats Associated with SQL Based EMR Software

Lee Heinke
School of Computer and Information Science
Edith Cowan University
Mount Lawley WA, Australia

Abstract

An increasing amount of E-health software packages are being bundled with Standard Query Language (SQL) databases as a means of storing Electronic Medical Records (EMR's). These databases allow medical practitioners to store, change and maintain large volumes of patient information. The software that utilizes these databases pulls data directly from fields within the database based on standardized query statements. These query statements use the same methods as web-based applications to dynamically pull data from the database so it can be manipulated by the Graphical User Interface (GUI). This paper proposes a study for an investigation into the susceptibility of popular E-health software packages to code injection attacks that are prevalent on web based applications. The proposed research also aims to examine the vulnerability of popular Australian E-Health software to network based attack methods in a test environment. Attacks of this nature on medical information systems have the potential to alter or destroy patient data, hold medical information services ransom or even disclose sensitive patient information.

Keywords

SQL, Injection, Exploit, Medical, Security, Threat

INTRODUCTION

The risks associated with storing medical records electronically lies in the fact that there accessibility has been increased. This increase in accessibility for legitimate users has provided opportunity for attackers to compromise the integrity of these once tangible records. Like all data stored within computer based information systems, Electronic Medical Records (EMR's) can be attacked using the same platform/service specific methodologies and techniques used to compromise, modify or destroy other data.

These platforms that are used by medical record systems are typically a database or data store utilising Standard Query Language (SQL) technologies. These SQL technologies are susceptible to injection based vulnerabilities where malicious code is "injected" into legitimate channels. SQL injection attacks being plainly the input of arbitrary SQL code into an application with a malevolent goal. The vulnerability does not exist in the underlying SQL architecture and in fact most injection based attacks rely on valid functioning of the SQL syntax. Many of the vulnerabilities that these attacks target exist within the front-end/user interface of these systems.

The front end of a computer based information system typically refers to the Graphical User Interface (GUI) which is what the user inputs commands and receives output via. However the front-end is often only a single part and often small part of the system. In the case of records management software the backend SQL database engine is the device that stores the data. In terms of EMR software this front-end typically consists of input fields that directly interact with the SQL database via web based or web enabled interface. If inputs are not properly sanitised into these input fields they have the potential to interact with the database to a level that is only limited by the attacker's knowledge of SQL injection attacks (Halfond, Viegas and Orso, 2006).

These attacks can enable insertion, modification or deletion of SQL structures and artefacts. Due to the widespread use of SQL for database systems across a wide range of applications and industries there exists a variety of readily available tools to exploit them. The article "SQL injection tools for automated testing" (Beaver, 2006) demonstrates how freely available tools can scan for and exploit vulnerabilities on susceptible target by a user with basic Windows skills. On a vulnerable system the tools in question have the potential to automate an entire SQL injection attack in the hope of achieving absolute control over the database culminating in its corruption or ultimate destruction and possibly even full administrative control of the underlying server.

There has been extensive research and investigation conducted in the field of SQL injection. There is even material available on the platforms of database this research intends to experiment with, Microsoft SQL Server 2000 and 2005. However the majority of the current research is focused on penetration testing in web based

applications using SQL systems. Within the field of research in medical information security there is certainly an awareness of the risks associated with storing medical records electronically (Williams and Mahncke, 2005; Valli 2006). It should be noted that the area of application package penetration testing within this field is largely unexplored.

The two EMR software packages that have been chosen for the proposed study are the Health Communications Network's (HCN) Medical Director and Best Practice by Best Practice Software Ltd. Medical Director has been chosen for the study due to the fact it has a market share of over 80% in the General Practitioner (GP) patient management software market (Department of Communications, Information Technology and the Arts, 2006). Best Practice is however a relative newcomer to the market as it was only released in 2004. It should be noted Best Practice was developed by a team under the guidance of Dr Frank Pyefinch who co-developed the original Medical Director software. Best Practice does not dominate the market like Medical Director however it does claim to be an advertisement free alternative to HCN's popular product (Helman, 2006).

ELECTRONIC MEDICAL RECORD SECURITY

Adopting computer based medical information systems to store the once paper based medical record has ignited discussion into whether or not our medical information is safe. This discussion has lead to research into what risks are associated with storing Electronic Medical Records (EMR's) and also the awareness of these risks amongst the medical fraternity.

The paper "The Underestimation of Threats to Patient Data in Clinical Practice" (Williams, 2005) investigates the security risks associated with storing medical information electronically and how they these risks are underestimated by medical practices. This brings to attention the need for security safeguards in medical practices in Australia and how education on the topic of EMR security is a worthy cause. The concerns presented in the above paper can be justified by anecdotal evidence released from the General Practice Computing Group in 2005 on the types of attacks medical practices have been subject to.

Similarly the feasibility and likelihood of common network based attacks occurring in a medical practice are discussed in the paper "The Insider Threat to Medical Records: Has the Network Age Changed Anything" (Valli, 2006). This paper makes light of the likelihood of network based attack to medical information systems due to the fact that the amount of online communication by medical practices is increasing.

Also there has been research conducted in South Australia by Holzer and Herrman in 2002 which revealed alarming results about the security of EMR's. The research found that 37 of the 73 practices surveyed did not protect patient data with even trivial precautionary measures as passwords. These findings which can be seen in Figure 1 below are only part of a survey on IT infrastructure in general practices.

Do you have written policies or procedures for:	Yes	No
Using passwords for electronic patient data security	46	37
Ensuring unauthorised persons cannot access confidential patient data when computers are left unattended	47	36
Maintaining a computer hardware and software register/ inventory	31	52
Routine maintenance and checking of the computer system	52	31
Implementing software upgrades	55	28
Electronic patient practice data backup	57	26
Staff access to the Internet	39	44
Staff use of e-mail	35	48
Virus protection	45	38

Figure 1 – Partial results from a survey of Practice Managers

E-HEALTH SOFTWARE

EMR's are stored, accessed, retrieved and modified by users via Electronic Medical Record Management Software (EMRS). Gradually E-health software is replacing paper based medical record systems. A review of popular E-Health software in Australia was conducted in 2006 by the health industry magazine Australian

Doctor. The review highlighted the fact that the majority of E-Health software in Australia is now bundled with commercial database software to store EMR's. Typically these databases are in the form of SQL or a similar variant. This software is not just an 'Ad-hoc' mix of databases and makeshift interfaces. There has been a standard developed to regulate the E-health software industry by the International Standards Organisation (ISO). The standard ISO/TC 215 to regulate the Health Information and Communications Technology market to enforce interoperability and compatibility of computer based medical information systems.

It has been cited in two articles, (Helman, 2006) and (Manktelow, 2004) saying that the E-Health software Medical Director 3 by the Health Care Network has a greater than 80% share of the clinical practice software market. Controversy initially arose when there was talk about Medical Director 3 using a Microsoft SQL database. Specifically this was in regards to the Microsoft licensing costs. However the current version of Medical Director 3 is packaged with Microsoft SQL Server Express which is the free version of the licensed software. It should be noted that because it is free its performance is limited. Specifically this has got to do with the size, performance and administration of the database.

MICROSOFT SQL SERVER

Microsoft SQL Server began as an effort between Sybase and Microsoft to create SQL Server 1.0 in 1989. This was Microsoft's first venture into the corporate database arena. In 1993 Microsoft and Sybase parted ways and then in 1999 Microsoft completely re-wrote the code to SQL Server and released Microsoft SQL Server 7.0. The current release by Microsoft is SQL Server 2005 (version 9.0). Aside from the proprietary version used large scale corporate environments there is also Microsoft SQL Server 2005 Express Edition. The express edition is free to download and deploy however it is only suitable in databases that are less than 4GB. Since it can be used in smaller environments it is often packaged with record management software so users of this software can avoid Microsoft licensing fees if they operate on a smaller scale.

Securing any SQL Server has always been an issue even from a simple user rights perspective on Local Area Network (LAN) based systems. A LAN being a network of interconnected computers within close proximity of one another. The propagation of the Internet and widespread adoption of it as a valid communication conduit for business data further exacerbates the problem. Specifically there have been publicized exploits of the software such as the SQL slammer worm. Robert Beverly of the Massachusetts Institute of Technology conducted an analysis of the SQL Slammer worm in 2003 on how it was able to exploit a buffer overflow vulnerability and autonomously replicate itself on Microsoft SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000. The result of this allowed an attacker to execute their own code and elevate privileges within the system so they could potentially 'own' the database. The solution from the vendor for this vulnerability was the application of a security patch by Microsoft.

SQL INJECTION

Structured Query Language (SQL), pronounced 'sequel' is a standardized language which is used to query relational databases. SQL was developed in the 1970's by IBM and is now both an International Standards Organisation (ISO) and American National Standards Institute (ANSI) standard. Most 'industry strength' database systems will meet or exceed these standards.

SQL injection attacks being simply the input of arbitrary SQL code with malicious intent. These attacks can only occur when the target application is vulnerable. Specifically these vulnerabilities are caused by input validation and error handling deficiencies (O'Leary-Steele, 2007).

The syntax of SQL injection attacks is dependent on the platform of the targeted application. This is due to the fact that different SQL database platforms have different query structures and rules. Currently the material that is readily available on SQL injection focuses on Microsoft SQL Server and My SQL. The paper by Cerrudo, 2002 "Manipulating Microsoft SQL Server Using SQL Injection" gives platform specific examples on the detection and attack methods of SQL injection. Even though the paper was published 2002 the syntax of the methods discussed in the paper are still applicable with current versions of Microsoft SQL Server. Examples of SQL injection on 'POST' and 'GET' variables within the HTTP protocol that drives web applications on Microsoft SQL Server can be seen in figure 2 below. Web Applications use the 'POST' and 'GET' arrays to store variable values between pages. SQL injection works by inserting SQL statements into input fields and the applications Uniform Resource Locator (URL) which is from where these variables receive their values.

Function	Example
Check for a vulnerability	http://site.com/start.asp?id=' OR 1=1--
Gathering information	http://site.com/start.asp?page=1&L=A'(select%20@@version)--
Adding a login	Exec sp_addlogin 'name', 'password'
Finding a field value	'x' OR full_name LIKE '%Bob%';

Figure 2 – Examples of SQL injection

SQL injection is classified into two main categories, Error Based SQL injection and Blind SQL injection. Error based injection techniques are reliant on the application outputting informational error messages. Figure 3 shows error based SQL injection being used to exploit a 'GET' variable which will output software information of the host.

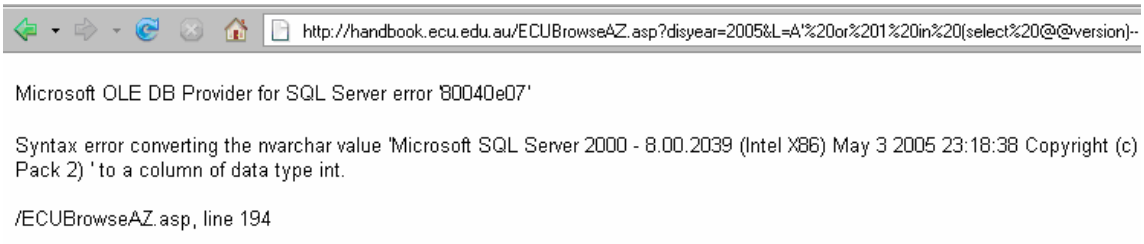

Figure 3 – An example of error based SQL injection

Blind SQL injection however has no reliance on error messages. A paper was published by the Imperva security group in 2003 titled 'Blindfoled SQL injection'. The authors, Maor and Shulman, expressed their concern that many applications are reliant on the 'Security by obscurity' when it comes to security. This was implying that just because an error message was not produced when processing input that the application is immune to SQL injection. However, this is simply not the case as currently there exists an abundance of research and tools which are based on blind SQL injection methods.

Classic examples of SQL injection include writing information to the database, reading hidden information from the database and avoiding authentication mechanisms. Recent examples of attacks using SQL injection include "Hacker Defaces Microsoft U.K. Web Page" (Ward, 2007) and "Hackers' deface U.N. site" (Keizer, 2007). These two examples demonstrate that the threat of SQL injection is very real and is not limited to whitepapers that simply analyse the topic such as "A Classification of SQL Injection Attacks and Countermeasures" (Halfond, Viegas and Orso, 2006). There is readily available material that goes beyond the simple modification of web-site material such as the paper "Advanced SQL Injection In SQL Server Applications" (Anley, 2002). Published by a security consulting group in the United Kingdom where the research methods describe how to access the internal network on which the database server is stored on.

NETWORK SECURITY TOOLS

If a vulnerability exists in service which is the result of a hack or bug, then there is a possibility it can be exploited. The use of platform specific exploits encompasses the areas of gaining unauthorized access, denial of service and code injection. Exploits can be downloaded, compiled and run individually or there are packages available of pre-compiled exploits. One of the more popular packages is Metasploit. The article "Powerful payloads: The evolution of exploit frameworks" (Skoudis, 2005) describes how Metasploit eliminates the need for an attacker to write their own exploits for a specific system. As well as having an encyclopaedia type resource of common exploits the framework also allows the user to customize the payload used in the attack. A payload is the code that is triggered by the exploit where as the exploit is the software that capitalizes on the flaw.

Conventional security measures such as firewalls and Intrusion Detection Systems (IDS) fail to negate the possible vectors for these types of attacks to be successful. A firewall is a hardware or software device that

protects a network from unauthorized access and an IDS is a system that detects unauthorized access, i.e detects intruders. However SQL injection and exploit based attacks utilize flaws in the targeted platform to gain legitimate access so it fails to raise any "red flags" within the system. SQL injection based attacks rely on the SQL functionality within the application working properly. If one was to limit the SQL functionality for security reasons then the application would not be able to take advantage of the SQL database either.

RISKS TO MEDICAL SYSTEMS

The use of the SQL injection and network based attacks on medical based information systems can lead to the deletion and modification of integral patient data. The possible consequences of this could be catastrophic. If a whole suburb's medical records are deleted how will doctors know what medications patients have had adverse reactions to in the past? If a just one part of a record is modified, such as a diabetic's insulin dosage, the ramifications could be fatal. There are also the legal implications of this data being compromised to be considered when looking at the importance of securing these systems. If a public figure had an embarrassing condition such as an STD on their EMR and this was leaked, then the organisation that was storing this data could potentially be liable.

Attacks of this nature are not limited to the destruction or falsification of electronic data. There is the potential for a 'terrorist' to hold electronic information at ransom. This is collectively known as a Denial of Service (DoS) attack. This type of attack involves flooding a host with requests in the aim of preventing legitimate users from accessing a system. Examples high profile of DoS attacks are discussed in an IEEE journal article "Denial-of-Service Rip the Internet" (Garber, 2000). Hypothetically a method like this could be used during a conflict so a hospital could not effectively treat its patients or even manage patient data such as a medication schedule. The effects of SQL injection and network based attacks are not limited to the welfare of patients. Insurance companies could potentially be targeted. A patient could alter a record about an injury that occurred in the workplace just to falsify a compensation claim.

DoS attacks do not discriminate their victims based on background or cause. Typically victims are selected using a 'shotgun' like approach. There is not a person behind a computer wanting to attack a certain organisation but rather a network of computers known as a 'botnet' scanning Internet Protocol (IP) ranges for vulnerable hosts. As soon as one is found an attempt to exploit it or even propagate a virus or a worm is made. A 'botnet' does not have sympathy to patients. If an SQL database that stores vital patient data is vulnerable it is still considered fair game by a 'botnet'.

PROPOSAL FOR FURTHER STUDY

The findings of the proposed research will clarify whether or not injection based vulnerabilities exist in popular Australian E-Health software packages. It is hoped that through performing penetration testing on these popular packages that the results will contribute to the body of research in medical information security. Since there is no readily available material on the penetration testing of EMR software, this research will endeavour to raise awareness on the possible vulnerabilities that not only could exist in the tested packages but also in other brands of EMR software.

The tests that will be performed in the proposed study will include but are not limited to the list below.

- Error based SQL injection testing.
- Blind SQL injection testing.
- Platform specific exploit testing.

The proposed research will not explore the consequences of any successful attacks but whether or not attacks using select SQL injection techniques and network attack tools are in fact feasible on popular E-Health software. It is hoped that this paper will provide future researchers with information concerning SQL injection based attacks in a non web environment. This is due to the fact that a large portion of SQL injection research is specific to the exploitation of 'POST' and 'GET' variables in browser based applications. To complement the proposed experiments which focus on attacks through the input fields of the EMR software, this research also aims to investigate the feasibility of using readily available network based attack tools to target the database behind these applications.

The software companies that produce EMR software do integrate a certain level of security features into the application package. In the case of Medical Director 3 and Best Practice this is in the form of password protection and privilege restriction options of user accounts. It should be noted that there have been claims of Systems Administrator (SA) password cracking being a trivial exercise (Litchfield, 2002). However those

countermeasures fail to take into account the possibility of malicious attacks occurring outside the inbuilt functionality of the interface.

CONCLUSION

The proposed research aims to find out if the confidential medical records in the Microsoft SQL databases packaged with Medical Director 3 and Best Practice are vulnerable to SQL injection and network based attacks. In the case of the proposed experiments returning results that would indicate vulnerabilities within the software then it is hoped that this paper could serve as a warning about the possible risks associated with storing electronic medical records. The possible targets for these types of attacks range from small scale medical practices to large scale hospitals housing thousands of patient records. If the information on one of these targets is destroyed, modified or held ransom then not only can the host machine be affected but the patients who have their vital information stored on these machines could face disaster.

REFERENCES

Anley, C. (2002). Advanced SQL Injection In SQL Server Applications. NGSSoftware Insight Security Research, URL http://www.nextgenss.com/papers/advanced_sql_injection.pdf, Accessed 1 Oct 2007

Beaver, K. (2006). SQL injection tools for automated testing. SearchSQLServer.com, URL http://searchsqlserver.techtarget.com/tip/1,289483,sid87_gci1159434,00.html, Accessed 5 Oct 2007

Beverly, R. (2003). MS-SQL Slammer/Sapphire Traffic Analysis. Massachusetts Institute of Technology, URL http://momo.lcs.mit.edu/slammer/, Accessed 5 Oct 2007

Cerrudo, C. (2002). Manipulating Microsoft SQL Server Using SQL Injection. Application Security Inc, URL http://www.appsecinc.com/presentations/Manipulating_SQL_Server_Using_SQL_Injection.pdf, Accessed 7 Oct 2007

Department of Communications, Information Technology and the Arts. (2006). The Australian software industry and vertical applications markets: Globally competitive and domestically undervalued. Centre for Innovative Industry Economic Research Consortium, URL http://www.dcita.gov.au/__data/assets/pdf_file/36474/Part_C_Vertical_markets_6-8.pdf, Accessed 5 Oct 2007

Garber, L. (2000). Denial-of-Service Attacks Rip the Internet. Computer, 33(4), 12-17.

General Practice Computing Group. (2005). Medical practice network security: Firewall tutorial. South Melbourne, Victoria, Australia: Royal Australian College of General Practitioners.

Halfond, W., Viegas, J., & Orso, A. (2006). A Classification of SQL Injection Attacks and Countermeasures. Georgia Institute of Technology, URL http://www.cc.gatech.edu/~orso/papers/halfond.viegas.orso.ISSSE06.pdf, Accessed 2 Oct 2007

Helman, T. (2006). 8 Medical Software Packages Put to the Test. Australian Doctor, URL http://www.zedmed.com.au/pdfs/SoftwareReview_JUN2306.pdf, Accessed 5 Oct 2007

Holzer, G., & Herrmann, N. (2002). Informatics Survey for Practice Managers. SA Divisons of General Practices Inc, URL http://www.sadi.org.au/survey/Practice_Managers_Survey_2002.pdf, Accessed 7 Oct 2007

International Standards Organisation. (2002). TC 215: Health informatics. International Standards Organisation, URL http://www.iso.org/iso/standards_development/technical_committees/list_of_iso_technical_committees/iso_technical_committee.htm?commid=54960, Accessed 7 Oct 2007

Jajodia, S., Noel, S., & O'Berry, B. (2007). Topological Analysis of Network Attack Vulnerability. George Mason University, URL http://www.karlin.mff.cuni.cz/urad/diplomky/soubory/1155201869-an.pdf, Accessed 3 Oct 2007

Keizer, G. (2007). 'Hackers' deface UN site. Computerworld, URL http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9030318, Accessed 1 Oct 2007

Litchfield, D. (2002). Microsoft SQL Server Passwords (Cracking the password hashes). NGSSoftware Insight Security Research, URL http://www.ngssoftware.com/papers/cracking-sql-passwords.pdf, Accessed 1 Oct 2007

Manktelow, N. (2004). Vendors vie for the GP software pie. Medical Observer, URL http://www.medicalobserver.com.au/displayarticle/index.asp?articleID=2572&templateID=105&sectionID=0&sectionName, Accessed 11 Oct 2007

Maor, O., & Shulman, A. (2003). Blindfolded SQL Injection. Imperva, URL http://www.imperva.com/download.asp?id=4, Accessed 1 Oct 2007

O'Leary-Steele, G. (2007). Buffer Truncation Abuse in Microsoft SQL Server Based Applications, URL SEC-1, URL http://www.sec-1labs.co.uk/advisories/BTA_Full.pdf Accessed 1 Oct 2007, Accessed 5 Oct 2007

Skoudis, E. (2005). Powerful payloads: The Evolution of exploit frameworks. SearchSecurity.com, URL http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci1135581,00.html, Accessed 10 Oct 2007

Valli, C. (2006). The insider Threat to Medical Records: Has the Network Age Changed Anything?. Perth, Western Australia: Edith Cowan University, School of Computer and Information Science.

Ward, K. (2007). Hacker Defaces Microsoft U.K. Web Page. Red Channel Partner Online, URL http://rcpmag.com/news/article.aspx?editorialsid=8762, Accessed 1 Oct 2007

Williams, P. (2005). The Underestimation of Threats to Patient Data in Clinical Practice. Perth, Western Australia: Edith Cowan University, School of Computer and Information Science.

Williams, P., & Mahncke, R. (2005). A New Breed of Risk: Electronic Medical Records Security. Perth, Western Australia: Edith Cowan University, School of Computer and Information Science.

Wolfgang, M. (2002). Host Discovery with Nmap. Insecure.org, URL http://insecure.org/nmap/docs/discovery.pdf, Accessed 1 Oct 2007

COPYRIGHT

Lee Heinke ©2007. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.