National Cybersecurity and Communications Integration Center

22 August 2013

SQL Injection Scanning Tools

Summary:

According to The Open Web Application Security Project (OWASP), injection attacks have been ranked as the number one threat to cyber security in 2013.[1] Injection attacks occur when un-trusted data is sent to an interpreter which relays the data as part of a command or query. This data can deceive the interpreter into executing arbitrary commands or accessing data without proper authorization.

One prominent form of injection attack called structured query language injection (SQLi) is a method which allows actors to communicate directly with a database by submitting SQL queries through vulnerable web applications.[2] SQL is a computer language that provides the ability to store, manipulate, and retrieve data from a database and is necessary for web applications to communicate with databases. This language communicates with databases including Oracle, Microsoft Access, MS SQL Server, and MySQL by sending commands through web application user input fields like login pages, support and product request forms, feedback forms, search pages, and shopping carts. For example:

> *A bank customer enters their username and password to access their account information. When the user clicks enter, a SQL query is generated and sent to the banking site's database for verification. If that SQL query identifies a matching username and password record in the database the user is granted access to the requested page.*

If commands sent to the database are not properly sanitized, actors can bypass login screens and access databases directly. Actors who exploit these vulnerabilities can gain access to sensitive information including user credentials, financial and payment information or proprietary company information. An SQL injection attack can also be used to alter information within databases or eliminate them all together. For example, an actor may add information to a database such as illegitimate user accounts which would then permit them to access a company's internal network with seemingly authentic credentials.

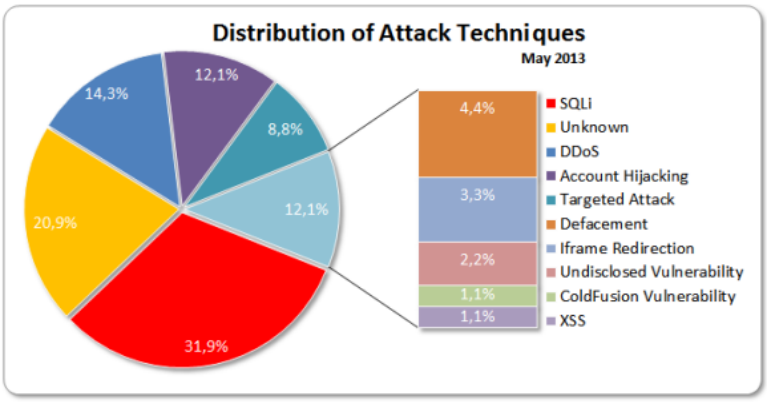
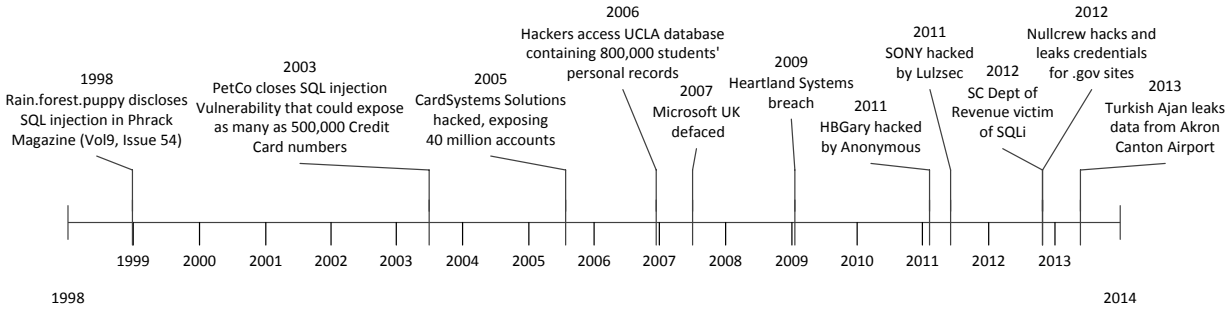


Figure 1: Attack Type Prevalence

SQLi attacks are difficult to defend against because they often communicate through web applications on port 80/443. These ports are configured to pass through firewalls, operating systems (OS) and network level security mechanisms.[3] According to Acunetix, a company that specializes in SQL scanning, nearly

70% of websites contain exploitable SQL vulnerabilities. This may be why SQLi attacks have been the method used in some of the more recently publicized cyber intrusions. The number of SQL vulnerabilities and their varying level of potential impacts are the likely reasons this method is so attractive to an array of malicious actors.



SQLi has been used in various high profile attacks against a wide range of targets. Some of the more recent attacks include that which occurred in October 2012 when Nullcrew used SQLi to target multiple US government domains and subsequently leaked several thousand administrator credentials. In more recent attacks, it is believed that the Turkish Ajan group used SQLi to deface and leak database information from US targets including the City of Lansing, Michigan and the Akron-Canton Airport in Akron, Ohio.[4]

A multitude of tools exist for locating SQL vulnerabilities including Acunetix, Havij and SQLmap. Acunetix and Havij are offered for sale by their creators, while SQLmap is a free download and part of an open source project. Acunetix is considered a vulnerability scanner as opposed to Havij and SQLmap which both have the capability of scanning as well as actually exploiting discovered vulnerabilities. Acunetix was ranked one of the best performing vulnerability scanners in a comparison conducted by security researcher Shay Chen. [5] SQLmap was also very highly ranked by this study in terms of effectiveness and according to Imperva joins Havij as one of the most prominently used SQLi exploitation tools.

Havij:

Havij is a database scanning tool which runs on the Microsoft Windows OS. Havij has the capability to find and exploit SQL vulnerabilities in targeted websites. The combination of the tool's ease of use and potential to cause severe impacts to targeted sites make it valuable to actors of all backgrounds and skill ranges. The tool, which was first released in July 2009, was developed by the Iranian security firm, ITSecTeam, and has been marketed as a penetration testing tool.[6] The tool is kept up-to-date by developers with add-on features and bug fixes. According to ITSecTeam's website, the most current version of the tool is version 1.17 and was released in December 2012. According to Josh Shaul of Application Security Inc, Havij has been adopted heavily by the black hat community who use it for malicious purposes even though it is marketed as a white hat product. The commercial version of the software retails on ITSecTeam's website for $650 and includes a one year

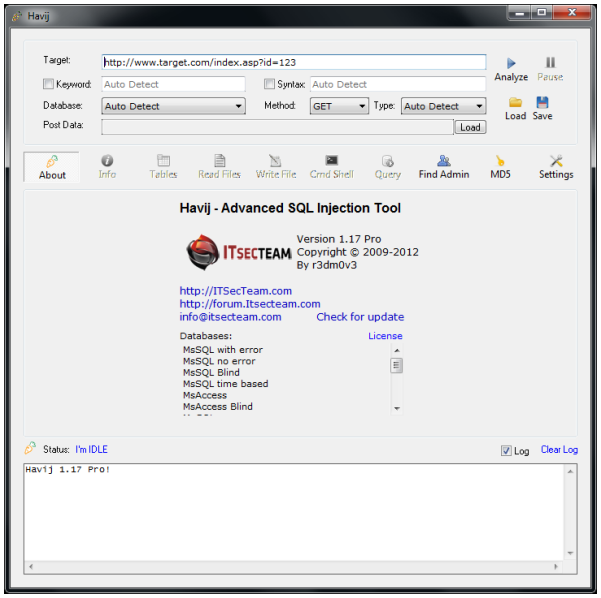


Figure 2: Havij GUI

subscription to use the tool as well as technical support if needed. Unlike previous versions of the software, the latest version does not come with a free evaluation edition which provided limited functionality. However, users are able to download previous versions of the software from the company's website including free evaluation editions. Payment is accepted via WebMoney, Liberty Reserve (now defunct) or Western Union.[7] Although Havij is sold by the ITSecTeam directly, cracked copies of version 1.17 can be found and downloaded through various web sources.

Havij's prevalence is likely tied to its relative ease of use. Actors looking to compromise a website need only enter the site's uniform resource locator (URL) into Havij's user friendly graphical user interface (GUI) and click analyze. Havij can perform back-end database fingerprinting, retrieve database management system (DBMS) user names and password hashes, dump tables and columns, fetch data from the database, run SQL statements, and even access the underlying file system and executing commands on the OS.[8] ITSecTeam marketing claims that against vulnerable targets, Havij has a success rate of 95%. These features make the tool useful to cyber criminals looking to steal user credentials, hacktivists looking to leak database data as a form of public embarrassment or actors looking to gain login credentials to maintain unauthorized access to a network.

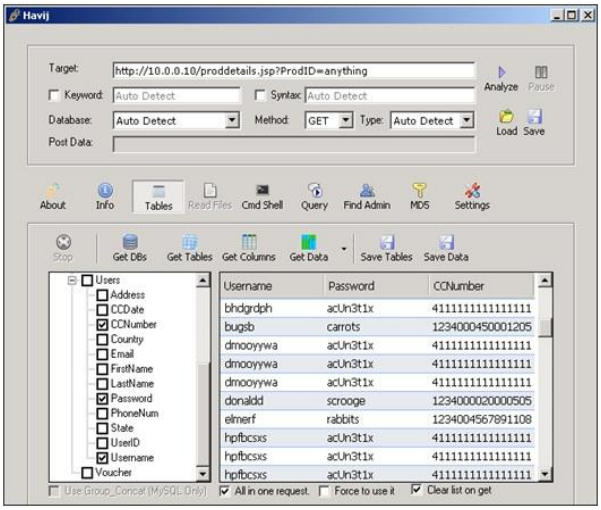


Figure 3: Havij Results

Havij is believed to be used by various groups including the hacktivist collectives of Anonymous and Lulzsec.[9] Additionally, the Tunisian Cyber Army is believed to use Havij for their SQLi attacks.[10] A trusted third party indentified an actor who used Havij to locate SQL vulnerabilities which could allow attackers administrative privileges on a website with over 4.8 million users. The actor then attempted to sell that information on the cyber underground.[11] These examples show Havij's use by both cyber criminals and hacktivists alike.

In April 2012, Imperva, a US security company, posted a blog article comparing Havij with another common black hat injection tool called SQLmap. In their research they noticed that Havij appeared to be more widely used than SQLmap. Imperva was able to identify Havij attacks originating from 48 countries as compared to only 9 for SQLmap. According to Imperva, more advanced actors may favor SQLmap because it can be extended and modified by its users.[12,13]

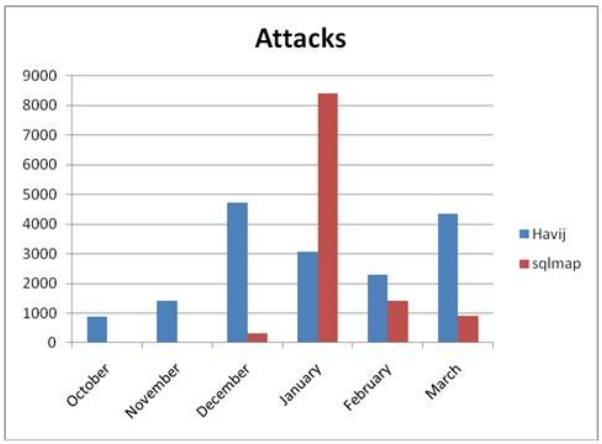


Figure 4: Havij SQLmap Activity

Imperva's blog revealed forum chatter in which one actor said, "Don't be a skid using Havij." The term "skid," short for script kiddy, is a derogatory slang term used to characterize low level/low skill actors who utilize automated tools and scripts to conduct cyber attacks. Further evidence of Havij's popularity among younger, inexperienced actors can be seen by querying Havij "how to" videos on YouTube. After performing this query, one researcher noticed many of these videos are posted by individuals who appear to be young adolescents.[14]

SQLmap:

SQLmap is a freely available open source penetration testing tool written in python which, like Havij, automates the detection and exploitation of flaws vulnerable to SQL injection. According to its website, SQLmap is equipped with a powerful detection engine and broad range of features from database fingerprinting, data fetching from databases, accessing underlying file systems and executing commands on the OS via out-of-band connections.[15] SQLmap is continually updated by its developers: Bernardo Damele A. G. and Miroslav Stampar; both of whom encourage SQLmap users to contribute code to the project. This community input keeps the software up to date and allows its functionality to continually expand.

Some of the key differences between Havij and SQLmap include:

- SQLmap's use of the command line as a user interface as opposed to Havij's GUI.
- SQLmap can be used against any OS running Python whereas Havij runs exclusively on Microsoft OS. Python is a free programming language compatible with Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines.[16]

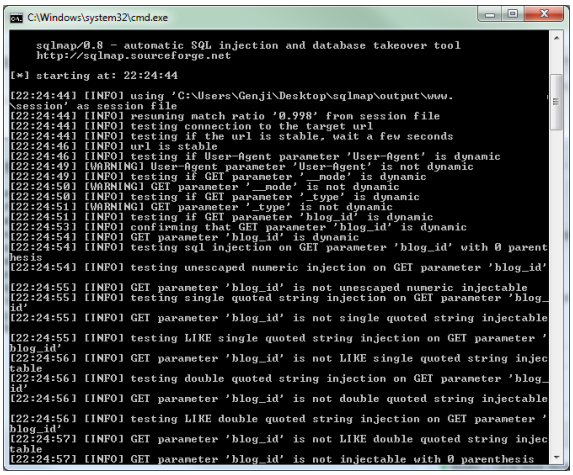


Figure 5: SQLmap Command Line

As opposed to Havij, which is a "point and shoot" application requiring only the URL of the site to be targeted, the use of SQLmap requires the user to input specific commands required by the tool. However, as with Havij, numerous how-to tutorials and videos exists which walk users through the various commands. According to forum discussions, SQLmap has the ability to use a multitude of obfuscation methods including the use of The Onion Router (TOR). TOR is a free tool which facilitates online anonymity by relaying internet traffic through various nodes to obfuscate its origination.[17] The use of TOR-infused anonymity is likely a main point of attraction for malicious actors interested in using SQLmap for nefarious purposes. Additionally, SQLmap is compatible with Metasploit framework, which is an open source sub-project of the Metasploit Project. Metasploit Project is a software platform dedicated to developing, testing and executing exploits.[18] In a comparison conducted between open source and commercially available SQLi tools, SQLmap actually tied a popular commercially available SQL scanning tool called Acunetix based on vulnerability identification success rate.[19]

Acunetix:

The Acunetix Web Vulnerability Scanner was released in July 2005 and is designed to locate security holes in web applications that can be exploited by an attacker. Acunetix looks for vulnerabilities that can be exploited through SQLi, cross site scripting (XSS), and weak passwords.[20]

> *Cross site scripting* (XSS) *occurs when attackers embed malicious code on a vulnerable website, often through web applications, or message boards. This code then runs on a visiting victim's browser unbeknownst to them, and can capture information including stored cookies and private information.*[21]

The software is available for purchase on the Acunetix website and is available in different suites which vary in price from $1,445 for use on a single website, to $12,995 for use on an unlimited number of sites. The company also provides edition upgrades and add-on options for maintenance agreements for additional costs.[22] Acuntetix often appears in the top 10 list of web application vulnerability scanners and in one detailed evaluation, is considered to be tied with Burp Suite as the top scanner for identifying SQLi vulnerabilities.[23] Of note, in this same evaluation, SQLmap tied Acunetix in overall ranking amongst open source and commercial scanning tools for SQLi vulnerability identification. Even though Acunetix may be one of the top tools used to identify SQLi vulnerabilities, it does not have the capability to exploit discovered vulnerabilities; that combined with the price of the tool make it less likely to be used by nefarious actors than tools such as Havij and SQLmap. However, actors who may be using the Acunetix tool for malicious purposes have a plethora of open source websites which keep repositories of exploits that correlate to the flaws discovered by Acunetix.[24]

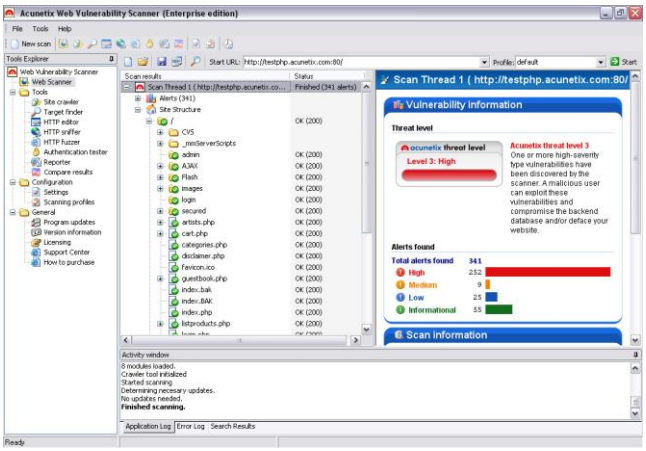


Figure 6: Acunetix GUI

When a vulnerability is discovered, Acunetix provides suggestions to best to defend against and mitigate those vulnerabilities. When scanning with Acunetix, the program first looks at the server to determine what technologies are used and then checks for vulnerabilities in the technologies it discovered. However, users may also manually select the technologies present. Acunetix also allows users to view, edit, or create HTTP requests. This allows users to dig deeply into applications where this type of activity would be difficult to automate. These features can prove just as valuable to penetration testers as potential attackers.

Acunetix provides easy to read findings reports and can import these results directly into supported web application firewalls. Web application firewalls (WAF) provide defense against HTTP (application) conversations by applying rules that can identify and block potential attacks. WAFs are a key factor in defending against SQLi attacks.

Mitigation:

Automated attack tools like SQLmap and Havij often produce user agent strings containing their name in the requests they send. Therefore, one method of defending against attacks from these tools is to blacklist certain user agent header values associated with the tools. Another method of defense includes implementing rate-based detection mechanisms. These tools look for interactions that occur at speeds not possible for human beings which is indicative of an automated tool.[25] The simplicity and automation of modern hacker tools makes it extremely important to understand how to properly defend against attacks that employ those tools.

To maintain a heightened security posture and better defend against injection, researchers suggests that security experts use Google dorking against their own website as this is often how actors locate sites they wish to target.[26]

Google Dorking is the act of using Google's advanced search functions to locate data that is vulnerable or unprotected. The search terms used to find these vulnerabilities are referred to as dorks.[27]

In layman's terms, malicious actors can conduct Google searches for specific vulnerabilities that will effectively return thousands of websites with those desired vulnerabilities. Once those vulnerable websites are collected, malicious actors can copy and paste the URL into a program like Havij and watch as the vulnerability as it is identified and exploited. Dorking your own website can allow security experts to uncover possible vulnerabilities and rectify them.

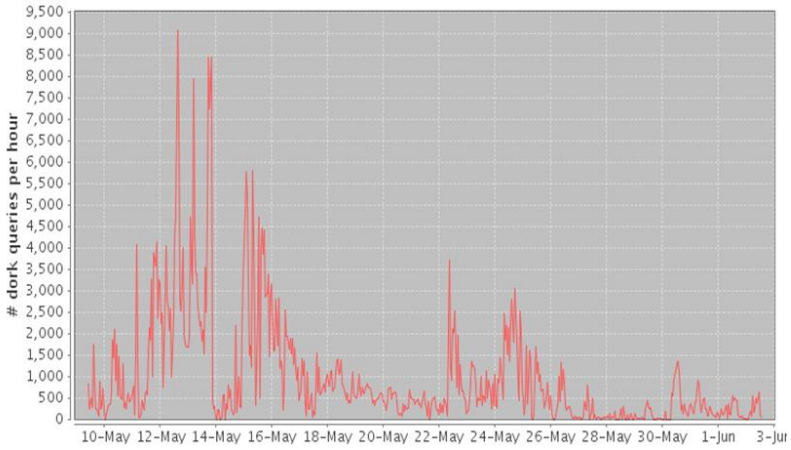


Figure 7: Google Dork Query Rate

It's also recommended to blacklist known attack source IPs. According to researchers, as of 5 January 2012, 40% of SQL injection attacks came from 10 IP addresses. This number has likely grown since then, but the premise of blacklisting known malicious source IPs remains pertinent as a valuable method of defense.[28] The use of WAFs, as discussed in the Acunetix section, is recommended to defend the applications as they can detect patterns and apply rules to help identify automated activity from scanning tools. Lastly, although time consuming and costly, reviewing and correcting errors discovered in code is recommended.

In 2009, US-CERT released a publication on SQLi which provides additional mitigation strategies.[29] US-CERT reiterates the fact the SQLi attacks are difficult to identify but suggests auditing network intrusion detection system (IDS) logs. Tools exists which allow server administrators to search for commands and characters contained in these logs that do not belong there. Examples of such include, "EXEC", "POST", "UNION", "CAST", or a single quotation mark. However, comprehensive detection can still be difficult even with logging enabled as actors can manipulate the whitespace between commands, encode using decimal, HEX, BASE64 and even inject characters that the webserver / database will ignore in order to evade detection by IDS or log-based analysis.

US-CERT provides the following best practices to minimize risks associate with SQLi: [30]

Network Level Recommendations

- Deny access to the internet except through proxies for Store and Enterprise servers and workstations.
- Implement firewall rules to block or restrict internet and intranet access for database systems.
- Implement firewall rules to block known malicious IP addresses.
- Harden internal systems against the potential threat posed by a compromised system on the local network. (Do not rely on firewalls to prevent access to insecure systems; secure them.)

System / Application Level Recommendations

- Secure both the OS and the application.
- Update and patch production servers regularly.

- Disable potentially harmful SQL stored procedure calls.
- Deny extended URLs.
- Sanitize/validate input.
- Ensure error messages are generic and do not expose too much information.
- Use principles of least privilege.
- Enforce best practice password and account policies.
- Document all database accounts, stored procedures, and prepared statements along with their uses.
- Perform regular audits and penetration testing.

Points of Contact

For all inquiries pertaining to this product, please contact the NCCIC Duty Officer at NCCIC@hq.dhs.gov or 1(800) 282-0870. NCCIC Watch & Warning and Analysis can be contacted at NCCIC_WatchandWarning@hq.dhs.gov.

Can I share this product?

Recipients may share TLP: GREEN information with peers and partner organizations within their sector or community, but not via publicly accessible channels.

References

[1] https://www.owasp.org/index.php/Top_10_2013-Top_10
[2] http://www.acunetix.com/websitesecurity/sql-injection/
[3] http://www.acunetix.com/vulnerability-scanner/
[4] http://hackmageddon.com/2013/06/03/15-31-may-2013-cyber-attacks-timeline/
[5] http://sectooladdict.blogspot.com/2012/07/2012-web-application-scanner-benchmark.html
[6] https://isc.sans.edu/diary/The+Havij+SQL+Injection+Tool/11011
[7] hxxp://itsecteam[.]com/products/havij-advanced-sql-injection/#tabset-tab-4
[8] http://www.darkreading.com/database/cybercrimes-love-affair-with-havij-spell/232700449
[9] http://www.danbuzzard.net/journal/lulzsec-and-anonymous-script-kiddie-sql-injection.html
[10] 20130422_TunisianCyberArmy.pdf
[11] Trusted third party
[12] http://blog.imperva.com/2012/04/dissecting-the-sql-injection-tools-used-by-hackers.html
[13] http://www.troyhunt.com/2012/10/hacking-is-childs-play-sql-injection.html
[14] http://www.troyhunt.com/2012/10/hacking-is-childs-play-sql-injection.html
[15] http://sqlmap.org/
[16] http://www.python.org/
[17] https://www.eff.org/torchallenge/what-is-tor
[18] http://searchsecurity.techtarget.in/definition/Metasploit-Project-Metasploit-Framework
[19] http://sectooladdict.blogspot.com/2012/07/2012-web-application-scanner-benchmark.html
[20] https://www.cccure.org/Documents/acunetix/acunetix.pdf
[21] http://www.acunetix.com/websitesecurity/cross-site-scripting/
[22] http://www.acunetix.com/ordering/
[23] http://sectooladdict.blogspot.com/2012/07/2012-web-application-scanner-benchmark.html
[24] http://webcache.googleusercontent.com/search?q=cache:F7EX1nhAWdEJ:www.hackforums.net/showthread.php%3Ftid%3D281202+&cd=7&hl=en&ct=clnk&gl=us
[25] http://www.imperva.com/docs/HII_Automation_of_Attacks.pdf
[26] http://blog.imperva.com/2012/01/sql-injection.html
[27] http://content.usatoday.com/communities/technologylive/post/2011/08/google-hacking-exposes-large-caches-of-personal-data/1
[28] http://blog.imperva.com/2012/01/sql-injection.html
[29] http://www.us-cert.gov/sites/default/files/publications/sql200901.pdf
[30] http://www.us-cert.gov/sites/default/files/publications/sql200901.pdf