Web Application Vulnerabilities Detection Techniques Survey

Nilesh Khochare, Satish Chalurkar, B.B.Meshram

Computer Department VJTI, Matunga, Mumbai

Summary

There are many commercial software security assurance tools that claim to detect and prevent vulnerabilities in application software. However, a closer look at the tools often leaves one wondering which tools find what vulnerabilities. This paper identifies taxonomy of software security assurance tools and defines one type of tool: web application scanner, i.e., an automated program that examines web applications for security vulnerabilities. We describe the types of functions that are generally found in a web application scanner and how to test it.

Key words:

Software assurance; software security; software security assurance tool; web application; vulnerability.

1. Introduction

[15]Firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be installed in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria. Usually, the firewall will only allow port 80 for internet connection and blocks other ports. To a certain extent, it is known that web applications are insecure. As port 80 is the only port available for Internet connection, the hackers will intrude the application layer by using Buffer Overflow, Structured Query Language (SQL) injection, Cross Site Scripting (XSS), Command Injection, and Session Manipulation. Generally, companies always have secured networks with insecure applications where this will possibly jeopardize all the companies system. Firewall is considered to be secured. It is the best tool for both Intrusion Detection and Intrusion Prevention. Figure 1 shows the percentages of the total vulnerabilities reported in the NVD (National Vulnerability Database) [17] represented by cross-site scripting and SQL injection vulnerabilities. The NVD contains no reports for XSS and SQL Web application security is difficult because these applications are, by definition, exposed to the general public, including malicious users. Additionally, input to web applications comes from within HTTP requests. Correctly processing this input is difficult. The incorrect or missing input validation causes most vulnerabilities in web applications.

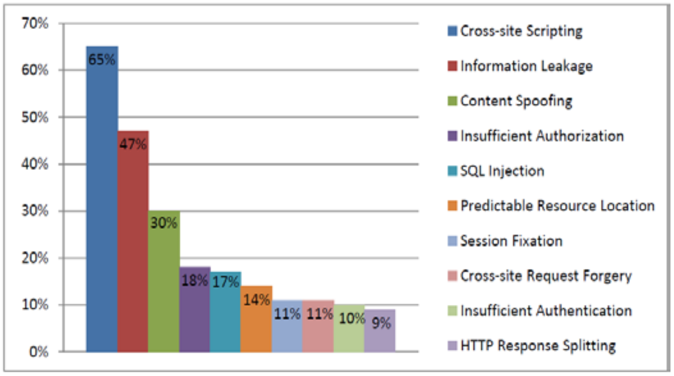

Figure 1 : Web Application Vulnerabilities

2. The Semate project

[16]The Software Assurance Metrics and Tool Evaluation (SAMATE) project intends to provide a measure of confidence in the software tools used for software assurance. Part of the SAMATE project is the identification and measurement of software security assurance tools, including web application scanners. When we have chosen a particular class of tools to work on, we begin by writing a specification. The specification typically consists of an informal list of features, and then more formally worded requirements for features, both mandatory and optional. For each tool class, we recruit a focus group to review and advice on specifications. We also develop a test plan and test sets to check that the tool is indeed capable of satisfying a set of mandatory requirements. Currently, we are developing a specification and test plan for source code analyzers. We also plan to develop a specification for web application scanners.

3. What is web application?

[18]In the early days of the Internet, the World Wide Web consisted only of web sites. These were essentially information repositories containing static documents, and

Manuscript received June 5, 2013
Manuscript revised June 20, 2013

web browsers were invented as a means of retrieving and displaying those documents. The flow of interesting information was oneway, from server to browser. Most sites did not authenticate users, because there was no need to—each user was treated in the same way and presented with the same information. Any security threats arising from hosting a web site related largely to vulnerabilities in web server software (of which there were many). If an attacker compromised a web server, he would not normally gain access to any sensitive information, because the information held on the server was already open to public view.[1]

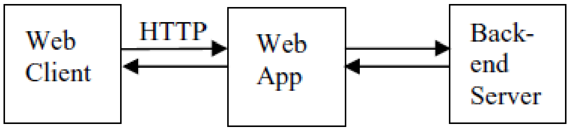

Figure 2 : Environment of Web Application

The technologies used to build web applications include PHP, Active Server Pages (ASP), Perl, Common Gateway Interface (CGI), Java Server Pages (JSP), JavaScript, VBScript, etc. Some of the broad categories of web application technologies are communication protocols, formats, server-side and client-side scripting languages, browser plug-ins, and web server API. A web application has a distributed n-tiered architecture. Typically, there is a client (web browser), a web server, an application server (or several application servers), and a persistence (database) server. Figure 2 presents a simplified view of a web application. There may be a firewall between web client and web server.

4. What is web application firewall

Web application firewalls (WAFs) are hardware or software devices positioned to monitor website traffic, with the ability to enforce policy on browser/server transactions. WAFs are similar, though not identical to, network firewalls where policies are typically applied to IP addresses, ports, and protocols.

WAFs are specifically designed to inspect HTTP(s) traffic and regulate data contained within headers, URL parameters, and web content. Another similarity: network firewalls are used to protect insecure hosts from remote exploitation. WAFs do the same for insecure websites. With a WAF in place, malicious hackers may target insecure websites, but attacks are intercepted and denied before reaching the custom web application code.

WAFs at their core are designed to separate safe web traffic from malicious traffic before it's received by the website. And, if an attack does find a way to sneak past a WAF, it still has the ability to prevent sensitive information from leaving the trusted network. To get a better understanding of how the technology works, it's helpful to view a WAF's functionality as three discrete components—policies, policy generation, and policy enforcement. Depending on the particular WAF in use, they may go about implementing each component in a number of different ways. No one particular way has proven to be the right way, as each has its pros and cons. Some instances of web applicationfirewalls are listed below.

3.1 AQTRONIX WebKnight

AQTRONIX WebKnight is an application firewall for IIS and other web servers and is released under the GNU General Public License. More particularly it is an ISAPI filter that secures your web server by blocking certain requests. If an alert is triggered WebKnight will take over and protect the web server. It does this by scanning all requests and processing them based on filter rules, set by the administrator. These rules are not based on a database of attack signatures that require regular updates. Instead WebKnight uses security filters as buffer overflow, SQL injection, directory traversal, character encoding and other attacks. This way WebKnight can protect your server against all known and unknown attacks. Because WebKnight is an ISAPI filter it has the advantage of working closely with the web server, this way it can do more than other firewalls and intrusion detection systems, like scanning encrypted traffic. These are some features of WebKnight.

- Open Source

WebKnight is free software under the terms of the GNU General Public License.

- Logging

By default all blocked requests are logged. In addition all allowed requests can be logged as well, or you can run WebKnight in logging only mode. This last operation mode allows you to see the attacks in the log files without blocking them. WebKnight can also prevent blocked attacks from being logged to the web server log files. This way your web server log files will be kept clean and accurate.

- Customizable

The firewall can be customized for any need, including blocking certain 0-day exploits before the vendor released a patch.

- Compatible with Web-Based Applications

WebKnight is compatible with Frontpage Extensions, WebDAV, Flash, Cold Fusion, Outlook Web Access, Outlook Mobile Access, SharePoint...

- HTTP Error Logging

WebKnight can be configured to log the HTTP errors from the web server. This way you can log common errors like '404 Not Found' or more severe ones like '500 Server Error' to the logfile. Doing so allows you to detect errors in scripts or attacks on them. You can also use it to simply find broken links in your web site or configuration mistakes.

- SSL Protection

Unlike traditional firewalls, WebKnight can protect encrypted sessions over HTTPS.

- Third-Party Application Protection

WebKnight not only protects the web server, but can also be configured to protect third-party web server applications, e-commerce web sites or your custom web site.

3.2 Guardian

Guardian@JUMPERZ.NET is an open source application layer firewall for HTTP/HTTPS. It works as a reverse proxy server. It analyzes all HTTP/HTTPS traffic against rule-based signatures and protects web servers and web applications from attack. When unauthorized activity is detected, Guardian@JUMPERZ.NET can disconnect the TCP connection before the malicious request reaches the web server.

3.3 IronBee

IronBee implements a robust framework for application security monitoring and defense. It provides a layered set of features at different levels of abstraction, enabling its users to choose the approach that works best for the work they need to accomplish.

3.4 SonicWall

As Web 2.0 applications emerge as the platform of choice for businesses and consumers, they increasingly become a target for criminal attacks such as SQL injection, parameter manipulation, cross-site scripting and Denial-of-Service (DoS). While small- to medium-sized businesses (SMBs) are increasingly adopting a Web presence, they often lack the in-house capabilities to keep up with the rapidly evolving challenges of Web security. Regulatory compliance mandates make Web application attacks particularly onerous for financial, healthcare, and application service providers, as well as e-commerce businesses. A complete, affordable, out-of-box compliance solution, SonicWALL® Web Application Firewall Service leverages your existing infrastructure as a licensable add-on module to the SonicWALL Secure Remote Access platform. Utilizing a dynamically updated signature database to detect sophisticated Web-based attacks and protect Web applications including SSL VPN portals, SonicWALL Web Application Firewall Service applies reverse proxy analysis of Layer 7 traffic against known signatures, denies access upon detecting Web application malware, and redirects users to an explanatory error page.

3.5 Barracuda

The Barracuda Web Application Firewall protects Web sites and Web applications from attackers leveraging protocol or application vulnerabilities to instigate data theft, denial of service, or defacement of an organization’s Web site. Unlike traditional network firewalls or intrusion detection systems that simply pass HTTP, HTTPS, or FTP traffic for Web applications, the Barracuda Web Application Firewall proxies this traffic and inspects it for attacks to insulate Web servers from direct access by hackers.

3.6 ModSecurity

Running public web applications may seem like playing Russian roulette. Although achieving robust security on the Web is possible in theory, there's always a weak link in real life. It only takes one slip of the code to allow attackers unrestricted access to your data. If you have a public web application of modest complexity running, chances are good that is has some kind of security problem. Take this URL for example:

http://www.webapp.com/login.php?username=admin';DROP%20TABLE%20users--

If your application is vulnerable to SQL injection, invoking the URL above may very well delete all user data from your application. Do you make regular database backups? Fortunately, the mod_security Apache module can protect you from this and other forms of web attacks.

3.7 Imperva Secure Sphere

The Secure Sphere Web Application Firewall (WAF) protects applications from current and future security threats by combining multiple security engines into a cohesive Web defense. Certified by ICSA Labs, Secure Sphere provides ironclad protection against the OWASP Top Ten, including SQL Injection, XSS and CSRF, and it addresses PCI 6.6. The SecureSphere WAF offers organizations drop-in deployment, automated, adaptable security, and low operational overhead, providing your business with a practical and highly secure solution that ensures your Web applications and data are safe. As the market leading Web Application Firewall, more organizations rely on Imperva to monitor and protect their critical Web applications than any other vendor. These are some features of Imperva.

- Automated Learning of Applications and User Behavior

A Web application firewall must understand application structure, elements and expected user behavior in order to accurately detect attacks. Imperva's patented Dynamic Profiling technology automates this process by profiling all application elements and building a baseline or “white list” of acceptable user behavior. It also automatically incorporates valid application changes into the application profile over time. Dynamic Profiling eliminates the need to manually configure and update application URLs, parameters, cookies, and methods.

- Research-Driven Security Policies

Powered by the Imperva Application Defense Center (ADC), an international security research organization, SecureSphere offers the most complete set of application signatures and policies available. The ADC investigates vulnerabilities reported by Bugtraq, CVE®, Snort®, and underground forums and performs primary research to deliver the most up-to-date and comprehensive Web attack protection available.

- Ironclad Defense Against Malicious Users

ThreatRadar Reputation Services, an industry-first reputation-based Web security service, identifies and stops known attack sources. ThreatRadar Reputation Services mitigates automated, large-scale attacks by integrating credible information about attacking IP addresses, bots, and anonymizing services into SecureSphere WAF defenses. Threat Radar Reputation Services delivers the following security feeds in near real-time:

a) Malicious IP addresses that recently attacked other Websites
b) Anonymous proxy addresses
c) Tor networks
d) Phishing URLs
e) IP geolocation data

Geographic location data supplied by the leader in IP geolocation allows SecureSphere to restrict access by country with an exceptionally high rate of accuracy. ThreatRadar Reputation Services can quickly and accurately block traffic from malicious sources before an attack can even be attempted.

- Bot and Automated Attack Protection

The SecureSphere Web Application Firewall combines multiple defenses together to stop the automated attacks like site-scraping, application DDoS, comment spam, and automated SQL injection attacks. Automated attack defenses include:

a) Threat Radar Reputation Services which identifies and stops known attack sources
b) Anti-automation technology which detects automated clients, bots, and scripts based on Web browser capabilities
c) Site scraping, application DDoS, and Google hacking security policies which are specifically designed to stop automated attacks based on rate limiting and known attack attributes.

In addition to automated attack protection provided by SecureSphere, Imperva offers the Cloud DDoS Protection Service which mitigates DDoS attacks that exceed Internet bandwidth limits.

- Web Fraud Prevention

Web-based fraud costs organizations with an online presence hundreds of millions of dollars each year, damages reputation and reduces customer loyalty. ThreatRadar Fraud Prevention Services enable organizations to rapidly provision fraud detection solutions without needing to update Web applications. By integrating with leading fraud security vendors, SecureSphere can identify and stop fraudulent transactions. ThreatRadar Fraud Prevention Services can also centrally manage WAF and fraud policies together.

- Virtual Patching Through Vulnerability Scanner Integration

For immediate patching of application vulnerabilities, SecureSphere can import assessment results from WhiteHat, IBM, Cenzic, HP, NT OBJECTives, Qualys, and others and create custom policies to block known vulnerabilities. Virtual patching reduces the window of exposure and the cost of emergency fix and test cycles.

- Platform and XML Attack Protection

SecureSphere protects Web applications and underlying infrastructure by detecting application, Web services, server, and network attacks. With over 8,000 signatures that are continuously updated by the Imperva ADC, SecureSphere fortifies all application layers against online threats. HTTP protocol validation prevents protocol exploits and evasion techniques. Flexible, rapidly-updated defenses allow SecureSphere to protect Web 2.0 applications and XML without requiring any application changes.

- Granular Correlation Policies Reduce False Positives

SecureSphere distinguishes attacks from unusual, but legitimate, behavior, by correlating Web requests across security layers and over time. This Correlated Attack Validation technology examines multiple attributes such as HTTP protocol conformance, profile violations, signatures, special characters, and user reputation, to accurately alert on or block attacks with the lowest rate of false positives in the industry.

- Customizable Reports for Compliance and Forensics

SecureSphere's rich graphical reporting capabilities enable customers to easily understand security status and meet regulatory compliance requirements. SecureSphere provides both pre-defined and fully-customizable reports. Reports can be viewed on demand or emailed on a daily, weekly or monthly basis. A real-time dashboard provides a high level view of system status and security events. Alerts are easily searched, sorted, and directly linked to corresponding security rules. SecureSphere's monitoring and reporting framework provides instant visibility into security, compliance, and content delivery concerns.

- Zero Impact Deployment and Ultra High Performance

SecureSphere provides the most flexible deployment options of any Web Application Firewall in the industry, including a unique drop-in deployment that requires no changes to existing applications or network. SecureSphere delivers multi-Gigabit throughput and tens of thousands of transactions per second while maintaining sub-millisecond latency.

- The Trusted Choice for Web Security

As the market-leading Web application firewall provider, more organizations rely on Imperva to monitor and protect their critical Web applications than any other vendor. Imperva SecureSphere provides your business with a practical and highly secure solution to ensure that your Web applications and data are safe.

5. Conclusion

We defined web application scanners and presented some vulnerability that this class of tools should detect. We plan to develop a specification for web application scanners. The specification will give a precise definition of functions that the tools in this class must perform. We will develop suites of test cases to measure conformance of tools to the specification. This will enable more objective comparison of web application scanners and stimulate their improvement.

References

[1] Angelo Ciampa,Corrado Aaron Visaggio,Massimiliano Di Penta ,“A heuristic-based approach for detecting SQL-injection vulnerabilities in Web applications” .

[2] Frank S. Rietta,”Application Layer Intrusion Detection for SQL Injection “

[3] Vulnerability Discovery with Attack Injection Joa˜o Antunes,Nuno Neves,Miguel Correia

[4] Automatic Creation of SQL Injection and Cross-Site Scripting Attacks Adam Kie˙zun,Philip J. Guo,Karthick Jayaraman,Michael D. Ernst

[5] An Architectural Approach to Preventing Code Injection Attacks Ryan Riley,Xuxian Jiang,and Dongyan Xu.

[6] Preventing SQL Injection Attacks Using AMNESIA William G.J. Halfond and Alessandro Orso

[7] Security in Open Source Web Content Management Systems, MichaelMeike , Johannes Sametinger and Andreas Wiesauer

[8] Web-Application Security: From Reactive to Proactive, John R. Maguire and H. Gilbert Miller

[9] FAULTS, INJECTION METHODS, AND FAULT ATTACKS , Chong Hee Kim and Jean-Jacques Quisquater

[10] New Threats and Attacks on the World Wide Web, HORSTEN HOLZ, SIMON MARECHAL, FRÉDÉRIC RAYNAL

[11] www.hackingarticals.com

[12] Jeremiah Grossman, The Five Myths of Web Application Security, WhiteHat Security, Inc, 2005.

[13] Michael Howard, David LeBlanc, and John Viega, 19 Deadly Sins of Software Security. McGraw-Hill Osborne Media, July 2005.

[14] G. McGraw, Software Security: Building Security In,Addison-Wesley Software Security Series, 2006.

[15] OWASP,WebScarab http://www.owasp.org/software/webscarab/

[16] SAMATE project, http://samate.nist.gov/

[17] National Vulnerability Database (NVD), http://nvd.nist.gov/

[18] “Web Application Scanners: Definitions and Functions” Elizabeth Fong and Vadim Okun, Proceedings of the 40th Hawaii International Conference on System Sciences - 2007