SURVEY ON WEB APPLICATION VULNERABILITIES PREVENTION TOOLS

Student, Nilesh Khochare*

Student,Satish Chalurkar*

Professor, Dr.B.B.Meshram*

Abstract—

There are many commercial software security assurance tools that claim to detect and prevent vulnerabilities in web application software. However, a closer look at the tools often leaves one wondering which tools find what vulnerabilities. This paper identifies various software security tools that can detect and prevent web application vulnerabilities.

Index Terms—Firewall; Software assurance; software security; software security assurance tool; WAF; web application; vulnerability.

* Computer Departmen, VJTI, Matunga, Mumbai.

I. INTRODUCTION

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) represented by cross-site scripting and SQL injection vulnerabilities. [16]

[20]The NVD contains no reports for XSS and SQL

Web se 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 vulnerability in web application.

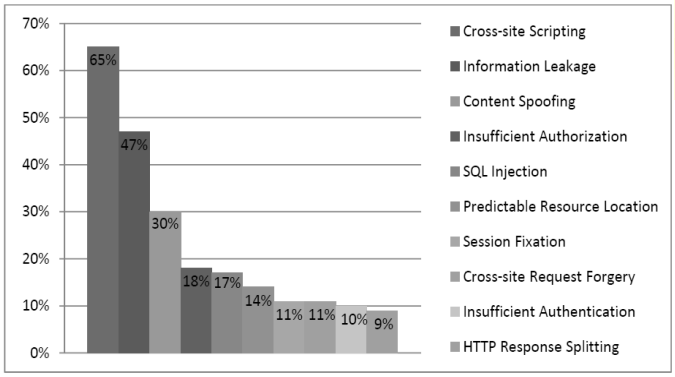

Fig. 1 : Web Application Vulnerabilities

II. THE SAMATE PROJECT

[17]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.

III. WHAT IS WEB APPLICATION?

[4]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 web browsers were invented as a means of retrieving and displaying those documents. The flow of interesting information was one-way, 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. 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.

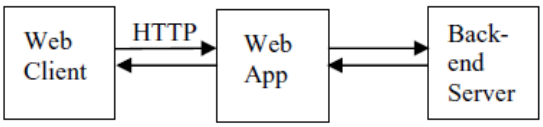

Fig. 2 : Environment of Web Application

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.

IV. 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.

V. Existing Web Application Firewall

A. *Citrix Application Firewall*

[10]The Citrix Application Firewall prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information. It accomplishes this by filtering both requests and responses, examining them for evidence of malicious activity and blocking those that exhibit it. Citrix application firewall protects web application from following attacks:

1) Buffer overflow attacks.

2) Cookie security attacks.

3) Forceful browsing.

4) Web form security attacks.

5) XML security attacks.

B. Sonic Wall Application Firewall

[11]SonicWall application firewall is complete, affordable, out-of-box compliance solution. SonicWall uses 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. SonicWall application firewall provides following features:

1) Cross-site request forgery protection

2) Strong Authentication and Authorization

3) Information disclosure protection

4) Session management

5) HTTPS inspection

C. Barracuda Web Application Firewall

[12]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. The Barracuda Web Application Firewall provides award-winning protection from all common attacks on Web applications, including SQL injections, cross-site scripting attacks, session tampering and buffer overflows. The Barracuda application firewall provides following protection features:

1) SQL Injection

2) OS command injection

3) Cross-site Scripting

4) Cookie or Forms Tampering

5) Brute force protection

D. Armlogic Profence Application Firewall

[13]Profense Web Application Firewall is implemented in the network as a filtering gateway which validates all requests to the web systems. Profense provides Proactive protection of web servers and web applications by employing a positive security model providing defenses against all OWASP top ten vulnerabilities. Armlogic Profence application firewall provides following protection features:

1) Session and CSRF protection

2) Positive and negative URL filtering

3) Positive and negative query filtering

4) Data leak prevention

5) DoS mitigation

6) SSL Client Authentication

E. ThreatSentry Web Application Firewall

[14]ThreatSentry is a multi-layered Web Application Firewall that protects Microsoft Windows Web servers from a broad range of web application threats including Cross Site Request Forgery (CSRF/XSRF), Structured Query Language (SQL) Injection, Cross-Site Scripting (XSS) and other attacks. ThreatSentry application firewall provides following protection features:

1) SQL Injection

2) Cross Site Scripting

3) Distributed Denial of Service

4) AI-Based neural behavioral engine

5) Protection from internal and external threats to IIS

F. *FortiWeb Application Firewall*

[15]The FortiWeb web application firewall provides specialized, layered application threat protection. FortiWeb's integrated web application and XML firewalls protect your web-based applications and internet-facing data from attack and data loss. FortiWeb application firewall provides following protection features:

1) SQL Injection

2) XML Schema Poisoning

3) Cross-site request forgery (CSRF)

4) Cross-site scripting (XSS)

5) Information Leakage

6) SYN Flood DoS Attack

7) Brute force login attack

VI. CONCLUSIONS

We studied various 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] Chong Hee Kim and Jean-Jacques Quisquater, "FAULTS, INJECTION METHODS, AND FAULT ATTACKS", Journal IEEE Design & Test archive Volume 24 Issue 6, November 2007.

[2] MichaelMeike , Johannes Sametinger and Andreas Wiesauer, "Security in Open Source Web Content Management Systems", Journal IEEE Security and Privacy archive Volume 7 Issue 4, July 2009

[3] HORSTEN HOLZ, SIMON MARECHAL, FRÉDÉRIC RAYNAL, "New Threats and Attacks on the World Wide Web, Journal IEEE Security and Privacy archive Volume 4 Issue 2, March 2006

[4] Elizabeth Fong and Vadim Okun,"Web Application Scanners: Definitions and Functions" Proceedings of the 40th Hawaii International Conference on System Sciences – 2007

[5] Angelo Ciampa,Corrado Aaron Visaggio,Massimiliano Di Penta ,"A heuristic-based approach for detecting SQL-injection vulnerabilities in Web applications", Proceeding SESS '10 Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems

[6] Frank S. Rietta,"Application Layer Intrusion Detection for SQL Injection ",Proceeding ACM-SE 44 Proceedings of the 44th annual Southeast regional conference

[7] Ryan Riley,Xuxian Jiang,and Dongyan Xu, "An Architectural Approach to Preventing Code Injection Attacks", Proceeding DSN '07 Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks

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

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

[10] Citrix Applicaton Firewall www.citrix.com/appfirewal

[11] SonicWall Application Firewall http://www.sonicwall.com/us/products/SRA_Web_Application_Firewall.html

[12] Barracuda Application Firewall http://www.barracudanetworks.com/ns/products/web-site-firewall-overview.php

[13] Profence Application Firewall http://www.armorlogic.com/web-application-firewall.html

[14] ThreatSentry Application Firewall http://www.privacyware.com/intrusion_prevention.html

[15] Fortiweb Application Firewall http://www.fortinet.com/products/fortiweb/index.html

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

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

[18] www.modsecurity.org/

[19] www.imperva.com/

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