Cyber Security Coursework - Essay

James Donohue - james.donohue@bbc.co.uk

Contents

1 Introduction **1**

2 Executive Summary of Paper **1**

References **3**

1 Introduction

The paper to be evaluated is 'Web Application Firewalls: Enterprise Techniques' by Jason Pubal (2015), as published on the SANS Institute Reading Room.

2 Executive Summary of Paper

The domain of the paper is the use of a Web Application Firewall (WAF) to monitor network traffic to a web application for the purpose of preventing malicious activity. WAFs represent a relatively new category of security product specifically designed to apply rule sets to HTTP or HTTPS traffic, including those designed to prevent common web application vulnerabilities such as SQL injection or cross-site scripting (XSS) (Conklin et al. 2016).

The paper starts by describing the recent historial context of network-based attacks and firewalls, with a rise both in the prevalence of web applications and the quantity of attacks directed against them, reportedly accounting for 35% of data breaches in 2014. The author suggests that organisations can help to manage the risks associated with Internet-facing web applicatons by using a WAF both to block malicious traffic and also to perform 'virtual patching' when a vulnerability is discovered, in order to reduce remediation time.

The author reports that WAFs provide increased visibility of application traffic compared to a network firewall or IDS, and are therefore capable of preventing application-level attacks that network firewalls cannot, without requiring any modification to the web application itself. They achieve this by examining HTTP requests and comparing them to attack 'signatures', either blocking such attacks or raising alerts. Positive and negative security models are contrasted, defined as whether the WAF rules define what is allowed (whitelisting) or what is disallowed (blacklisting). These two models are analogous to the possible default policies of a packet filtering firewall, 'discard' or 'forward' (Stallings and Brown 2015). The author suggests that WAFs using negative security models are both easier to set up and have

a lower configuration maintenance burden than those using positive models, but are not able to protect against unknown attacks and therefore less secure.

The paper then describes various approaches to deploying a WAF in a network environment (section 1.3). In a 'Reverse Proxy' configuration the WAF sits inline between the web server being protected and the network's external firewall, intercepting and proxying requests to the server, with rule sets applied. In a 'Layer 2 Bridge' deployment, the WAF is also inline but operates at a lower network layer, blocking traffic as required by simply dropping packets. In 'Out-of-Band' mode, rather than being inline the WAF receives a copy of network traffic which it monitors passively, interrupting malicious connections by sending TCP reset packets. The 'Server Resident' configuration means that WAF software is installed on the web server itself, removing the additional point of failure of a separate network device. Finally, 'Internet Hosted/Cloud' deployments rely on software as a service (SaaS) from a third-party cloud provider, with the WAF conceptually inline, similar to the 'Reverse Proxy' option.

In the next section (1.4) the paper covers in detail the main drivers (motivations) behind the use of WAFs in organisational contexts. The benefit given for 'production' applications, even those developed using a secure software development lifecycle (SDLC) is that the time and cost of remedying security issues that are identified after the application is live can be reduced. This benefit is shown as particularly relevant both to legacy applications that were developed in house and commercial off-the-shelf (COTS) software, where the organisation's ability to address underlying security issues in code may be restricted due to loss of relevant development skills or lack of vendor cooperation. The author next situates this approach as part of the vulnerability management process, specifically the need to 'shield' a vulnerable application from attacks while the affected code is fixed or updated, describing the technique as 'virtual patching'. The accuracy of a WAF is said to increase if it can import results from a dynamic application security testing (DAST) tool.

The need to ensure compliance, such as with the Payment Card Industry Data Security Standard (PCI DSS) for organisations that need to handle payment card information, is given as another significant driver of WAF usage. The PCI DSS requirement to review web applications using vulnerability assessment tools after any changes, and the high fines for which such organisations are liable, are given as reasons for using WAFs as an alternative way of achieving compliance. Next the paper describes the role of WAFs as sensors within a larger intrusion detection system. Data from a WAF can be sent to an organisation's security incident and event management (SIEM) system for correlation with other data, which the author says expands such a system's capabilities into the area of detecting attacks against the organisation's web properties. A brief depiction of major WAF vendors at the time of writing is then given.

The rest of the paper describes a lab environment created by the author to demonstrate how a WAF can be used to support virtual patching and security monitoring. The open-source ModSecurity WAF is installed on the same Linux-based virtual machine as a PHP/MySQL web application specially designed to exhibit common security vulernabilities as a teaching aid, the Damn Vulnerable Web App (DVWA). This illustrates the Server Resident model described earlier, although it is stated that ModSecurity also supports the Reverse Proxy approach. The OWASP Cole Rule Set (CRS) is imported into ModSecurity for illustration.

On a separate virtual Linux server, the log management tool AuditConsole is installed to illustrate the aggregation of audit logs from potentially multiple instances of ModSecurity. This platform offers tools for creating notifications based on events or performing further analysis.

Lastly a Windows host is provisioned running a DAST tool for identifying web application vulnerabilities called Burp Suite, and another tool called ThreadFix that can aggregate results from various security testing tools including Burp Suite and use them to generate WAF rules, which ModSecurity can then

import.

The author then discusses virtual patching in more detail, describing the vulnerability management process and the possibility of using a WAF to fix web application vulnerabilies without changing the application's source code. This is because the WAF is able to intercept and prevent attacks that match a particular rule. The paper illustrates this with the example of an (intentional) XSS vulnerability in DVWA. Using Dynamic Application Security Testing (DAST) such vulnerabilities can be identified by 'spidering' the web application and recursively checking for security issues. In the lab, Burp Suite was used to test DVWA and identified the XSS vulnerability above. The Burp Suite results were imported into ThreadFix as an example of aggregating findings from multiple tools and web applications. ThreadFix then generates WAF rules corresponding to the vulnerability, which are deployed to ModSecurity. Following installation, the XSS vulnerability is manually re-tested and is blocked by the WAF. The author points out that although the advantage of this approach is the speed with which the attack surface can be reduced, it may not always be possible to remediate the vulnerability entirely in this way, and therefore virtual patching should only be viewed as temporary risk reduction.

In the next part of the lab, the author describes the concept of Network Security Monitoring (NSM) and its assumption that security breaches are inevitable. This approach shifts the goal to detecting and reacting appropriately to incidents. This approach is broken down into three phases: collection, in which sensors (which may be WAFs among other types) collect data for analysis, detection, in which collected data is examined and alerts generated, and analysis, when a human interprets the data produced and takes action as necessary. The paper states that the importance of WAFs are sensors within an NSM infrastructure depends on how critical web applications are to the organisation's goals. WAFs are shown to be particularly effective where inbound network traffic must be decrypted before inspection.

The lab demonstrates the use of ModSecurity as an NSM sensor sending logs and alert data to the AuditConsole management tool. The AuditConsole dashboard shows alerts produced via ModSecurity owing to the OWASP CRS as a result of the scan performed by the Burp Suite DAST.

In conclusion, the author re-emphasises the importance of web applications today and re-states that virtual patching can quickly reduce risk cause by vulnerabilities in production web applications, and may be the only option available for legacy or COTS applications. The author suggests that WAFs have visibility into application traffic that no other monitoring tool is capable of. Finally the author points out the specialist skill set required for application security monitoring as opposed to 'general' network monitoring, and recommends that suitable training is provided.

References

Conklin, Wm. Arthur, Greg White, Dwayne Williams, Roger Davis, and Chuck Cothren. 2016. *Principles of Computer Security*. Fourth edition. McGraw-Hill.

Pubal, Jason. 2015. "Web Application Firewalls: Enterprise Techniques." https://uk.sans.org/reading-room/whitepapers/application/web-application-firewalls-35817.

Stallings, William, and Lawrie Brown. 2015. *Computer Security: Principles and Practice*. Third edition; Global edition. Harlow, Essex: Pearson.