IMPERVA®

What Next Gen Firewalls Miss 6 Requirements to Protect Web Applications

WHITE PAPER

Introduction to Web Application Security

Web attacks like SQL injection, DDoS, and Man-in-the-Browser threaten nearly every organization with an online presence. Most websites receive a continuous barrage of attacks—to the tune of one attack every two minutes[1]—and an alarming number of these attacks succeed. In fact, over half of all organizations suffered a Web application breach in the past year, and many of these breaches resulted in severe financial losses.[2]

Deploying an application security solution can reduce the risk of a Web application breach. While some next generation firewall vendors contend that their products stop application attacks, they do not provide the accuracy, the granularity, or the breadth of protection to thwart Web-based threats. To safeguard Web applications, businesses should look beyond next generation firewalls and evaluate solutions that can correctly identify Web attacks and stop emerging application threats.

The Application Threat Landscape

Hackers attack websites to steal data, disrupt access, and commit fraud. Web attacks are responsible for most large-scale—and costly—data breaches. The numbers don't lie. Hacking, which includes attacks like SQL injection, remote file inclusion, and brute force, resulted in 81% of data breaches and 99% of compromised records.[3] According to Verizon Business, hacking outpaced every other attack vector, including malware, social engineering, and physical data theft, in terms of stealing valuable data.

Why are Web attacks so common? Because most websites today contain vulnerabilities,[9] and many of these vulnerable sites house sensitive data like credit card numbers and customer information. Public websites are also easily accessible from the Internet and, consequently, are wide open to attack; hackers do not need to infiltrate a corporate network before they can seek out sensitive data. Due to the prevalence of application vulnerabilities and the ease with which hackers can exploit these vulnerabilities, websites have become a top target of attack.

[1] Web Application Attack Report, Edition #3, July 2012, Imperva

[2] Infosecurity Magazine, "Web Developers' Application Security Sorely Lacking," citing Forrester Consulting

[3] "2012 Data Breach Investigations Report," Verizon Business, 2012

[9] WhiteHat Website Security Statistic Report," WhiteHat Security, 12th Edition

Why Next Gen Firewalls Are Not Enough

Business owners may mistakenly believe that their next generation firewalls or intrusion prevention systems (IPS) will mitigate Web-based threats. While these solutions protect networks and users, they are ill-equipped to stop attacks that target customers' own websites. Although next gen firewalls are "application aware"—meaning that they can prevent users from visiting phishing sites or tunneling applications in HTTP—they are not designed from the ground up to protect Web applications. As a result, they leave holes in their application defenses—defenses that are only addressed by dedicated Web Application Firewalls (WAFs).

These shortcomings may be one reason why Gartner contends that it "does not see NGFW and WAF technologies converging because they are for different tasks at different placements"[10] in the network.

What Next Gen Firewalls Miss: 6 Requirements to Protect Web Applications

Since Web application attacks account for the lion's share of breached records, organizations should undertake every means possible to stop them. With sophisticated attacks that manipulate Web application behavior and emerging threats like attack automation, businesses need to make sure they have the necessary defenses in place to stop Web attacks. Unfortunately, next generation firewalls lack six crucial features that leave websites exposed:

1. Input Validation
2. Correlation
3. Cookie and Session Protection
4. Anti-Automation
5. Fraud Prevention
6. Useful Alerts and Policies

A Closer Look at the Stats

99% of breached records are due to hacking[4]
75% of all cyber-attacks target Web applications[5]
79 is the average number of serious vulnerabilities per website[6]
55% of security professionals believe developers are too busy to address Web security[7]
$5.5 Million is the average cost of a data breach[8]

[4] "2012 Data Breach Investigations Report," Verizon Business, 2012
[5] Gartner Research
[6] "WhiteHat Website Security Statistic Report," WhiteHat Security, 12th Edition
[7] "State of Web Security," Ponemon Institute
[8] "US Cost of a Data Breach," Ponemon Institute, 2012
[10] Magic Quadrant for Enterprise Network Firewalls, 2013, Gartner

1. Input Validation

To stop both known and custom Web attacks, a security solution must understand the structure and the usage of the application it is protecting. Understanding the application helps detect attacks like parameter tampering and reconnaissance. It also improves the accuracy of identifying attacks like SQL injection and cross-site scripting (XSS).

To validate input, a security solution must inspect parameter and form field values for special characters like apostrophes and brackets, and it must know whether these characters are normal or indicative of an attack. Input validation catches evasion techniques used by hackers to outwit attack signatures.

While next generation firewalls are "application-aware"—meaning that they can identify and fingerprint different applications—they do not inspect application elements and detect unusual behavior. As a result, they are blind to application abuse like parameter tampering.

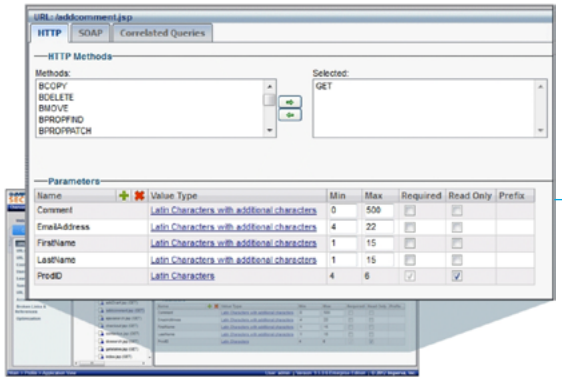
Validating input means not just verifying the length, characters, and format of website parameters, but also understanding the logic of the protected application. A security solution that validates input should understand which parameter values can be modified by end users and which ones cannot, thereby thwarting parameter tampering attacks. To put parameter tampering in perspective, in late 2011, a Fortune 100 bank was compromised when hackers altered account numbers listed in clear text in URL parameters. The hackers were then able to access other customers' accounts and transfer account funds.

Parameter tampering is a common attack; hackers frequently manipulate parameter values to change online account numbers, discount codes, user privileges, or other "hidden" settings in applications. Next generation firewalls cannot track read-only parameter values, so they cannot stop this type of attack. More generally, next generation firewalls cannot validate application input, so they cannot prevent a wide range of attacks that exploit input validation flaws in Web applications.

Web Application Firewalls Validate Input

Web application firewalls detect application abuse like parameter tampering by validating input. Because they build a baseline of expected input, Web application firewalls can also more accurately stop attacks like SQL injection and cross-site scripting.

By profiling Web application behavior, for instance, a Web application firewall can determine which users should not add brackets, braces, and semi-colons into a zip code field on a registration page, but can enter these same characters into a comment field. Validating input provides the context needed to differentiate between attacks and legitimate requests.



A Web application firewall can automatically learn application structure and expected user behavior to validate input.

2. Correlation

Organizations need to block Web attacks without blocking legitimate traffic. This may sound simple, but the solution is not. Next generation firewalls try to tackle this challenge by comparing Web requests to known attack signatures. These signatures attempt to enumerate all possible attacks while still maintaining high-throughput and low latency.

Unfortunately, because both JavaScript and SQL are so rich and complex, attackers can create millions of different attacks. To stop SQL injection attacks, next gen firewall vendors must develop signatures that detect SQL keywords like "select" and "union" and operators like apostrophes and semi-colons, but that will not block legitimate requests with these same words and characters. For example, if an online user submits a registration form indicating that she works at "O'Reilly Union High School," the submission will often be blocked because of the presence of an apostrophe and a SQL keyword in a form field.

Besides generating false positives, signatures have difficulty handling the sheer number of attack variations. As a result, hackers can craft attacks that evade signature detection. The solution to this challenge is to use an advanced correlation engine to examine suspicious Web requests.

Web Application Firewalls Correlate Suspicious Attributes to Correctly Identify Attacks

Many Web application firewalls provide correlation engines that examine multiple attributes of a request and multiple requests over time to accurately stop attacks. Correlation engines can build a risk score based on the presence of suspicious characters, signature violations, attack keywords, profile violations, and protocol violations and then further analyze high risk requests with regular expressions. These regular expressions can examine the exact order and presence of suspicious characters to better assess how the content would be interpreted by a back-end database—to detect SQL injection—or by a Web browser—to detect XSS.

By analyzing requests with an advanced correlation engine, rather than relying solely on static attack signatures, Web application firewalls can stop attacks while allowing legitimate traffic to flow uninterrupted.

"The firewall application control approaches by most NGFW vendors...are mostly about controlling external applications, such as Facebook and peer-to-peer (P2P) file sharing. WAFs are different: [they]... are concerned with custom internal Web applications."

MAGIC QUADRANT FOR ENTERPRISE NETWORK FIREWALLS, 2013, GARTNER

3. Cookie and Session Protection

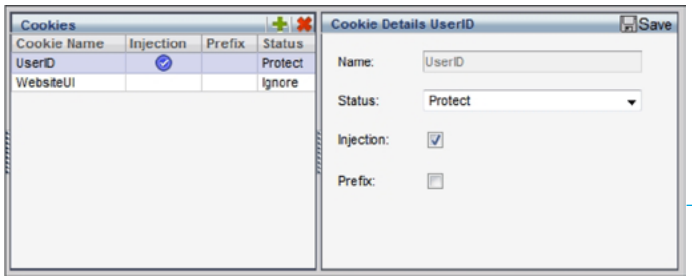
Hackers hijack Web sessions and manipulate cookies to gain unauthorized access to application content. Common attacks include cookie injection, cookie tampering, session replay, and session hijacking. Through these Web-based exploits, hackers can access valid users' accounts to steal data.

Next generation firewalls do not track sessions or cookies, so they can't protect Web applications from session-based attacks.

Web Application Firewalls Protect Cookies and Sessions

Web application firewalls, on the other hand, offer multiple defenses to protect sessions. They can track when cookies are set by the Web server and then observe any changes to these cookies. If a hacker steals active session data through a man-in-the-middle attack,[11] a Web application firewall can detect the attack and block the malicious user. If deployed as a reverse proxy or a transparent reverse proxy, a Web application firewall can also encrypt and sign cookies.

Hence, organizations concerned about cookie and session attacks must look past next generation firewalls and consider dedicated application security products like Web application firewalls.



Web application firewalls protect cookies and sessions.

[11] Not to be confused with a Man-in-the-Browser Trojan attack, described in section #6.

4. Anti-Automation

Cybercriminals today use armies of bots to unleash massive attacks like application DDoS or to scan websites for vulnerabilities. On top of the threat posed by cybercriminals, organizations must contend with other types of automated attacks, like competitors scraping Web content and comment spammers injecting ads into online forums and message boards.

Many automated attacks, like site scraping and HTTP floods, circumvent standard signature-based defenses. While next generation firewalls may be able to detect known "bad actors" through reputation and geolocation services, they cannot distinguish between human users and bots. Defeating automated attacks requires more: it requires identifying both automated clients and automated attack behavior. Because next generation firewalls cannot distinguish between automated and manual actions, they are unable to fully mitigate automated threats like application DDoS attacks.

Web Application Firewalls Protect Applications against Automated Attacks

Web application firewalls offer the necessary defenses to stop automated attacks. They can detect bots by transparently testing clients' Web browsers to determine if they are script-based bots. Moreover, they can detect users that request an excessive number of pages in a short period of time—the hallmark of a bot. They can also pinpoint suspicious activity such as repeated downloads of large-sized files. As a result, Web application firewalls thwart a wide range of automated attacks that can compromise, deface, and debilitate websites.

5. Fraud Prevention

Web fraud, which includes malware and transaction fraud, has become enemy number one for banks and retailers. Online businesses lose millions of dollars due to fraud remediation costs, chargeback fees, and customer turnover.

Next generation firewalls cannot stop Web fraud. Some "United Threat Management" appliances can scan application traffic for viruses, but they cannot detect end users infected with fraud malware or identify fraudulent devices.

Web Application Firewalls Can Mitigate Online Fraud

Solutions like the Imperva SecureSphere Web Application Firewall can prevent online fraud. Positioned between Web users and applications and with full visibility into layer 7 transactions, Web application firewalls can analyze end user attributes and Web traffic patterns for the tell-tale signs of malware infection. They can also integrate with leading fraud security solutions to detect fraudulent devices.

Once a fraudulent device or malware-infected client is identified, a Web application firewall can perform a number of actions such as blocking a transaction, monitoring a user for a specified period of time, generating an alert, or integrating with an external fraud management solution. Web application firewalls also identify and stop fraud without requiring any changes to the protected Web application.

[12] Some next generation firewalls may be able to record traffic for the purposes of debugging, but are not designed for security incident response.

6. Useful Alerts and Policies

To protect Web applications, organizations need granular application visibility and control. Organizations need to be able to review security incidents to understand the full details of both the attack and the perpetrator. Capturing event details like application user name, unusual header value, and browser agent can help security administrators define new policies with laser precision.

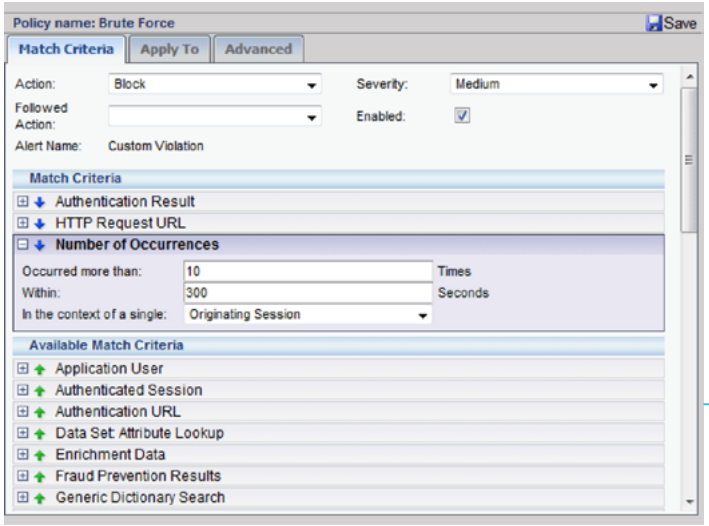
Next generation firewalls often record summary information like the source IP address, the type of attack, and the corresponding attack signature,[12] but they won't capture the Web request or the server response code. Next generation firewalls also will not record the full header and body of a request—information that is vital for forensics. The result is that next generation firewalls do not provide the visibility required to analyze attacks.

Next generation firewalls often only provide high-level rules to mitigate Web attacks. They cannot block requests based on attributes such as the presence of a cookie, the HTTP referrer, or the Web server response time. Because they are designed to stop network-level threats, they don't offer the policy granularity needed to stop custom Web attacks.

Web Application Firewalls Provide Application-Level Visibility and Control

Most Web application firewalls include flexible, custom Web application policies that enable administrators to define very specific policies to block attacks. Application policies can be built using dozens of criteria; policies can even analyze the number of requests in a period of time. Web application firewalls enable businesses to build fine-grained policies and apply them in real time, offering the defenses needed to combat Web attacks without blocking legitimate traffic.

When Web application firewalls record security events, they capture the entire Web request and the Web server response code in security alerts. WAFs also document important information like application user names and request headers and even identify the exact string in a request that triggered a violation, making it easy for administrators to analyze events.



Web application firewall administrators can build custom policies to mitigate threats specific to their Website.

Conclusion

Web application attacks represent one of the greatest security challenges facing organizations today. Hacktivists have emerged from almost nowhere to become a major threat, compromising hundreds of organizations' mission-critical websites. Cybercriminals have become industrialized, using automated tools to steal data while in turn optimizing their fraud operations. State-sponsored hackers have broadened their scope beyond military secrets to include commercial intellectual property.

While next generation firewalls are essential for protecting networks and for protecting users from external applications and threats, they do not offer the defenses necessary to safeguard customerowned Web applications.

To protect their business, organizations need to protect their Web applications and their application data. And to do this, they need a Web application firewall.

About Imperva

Imperva® (NYSE: IMPV), is a leading provider of cyber security solutions that protect business-critical data and applications. The company's SecureSphere™, Incapsula™ and Skyfence™ product lines enable organizations to discover assets and vulnerabilities, protect information wherever it lives - onpremises and in the cloud - and comply with regulations. The Imperva Application Defense Center, a research team comprised of some of the world's leading experts in data and application security, continually enhances Imperva products with up-to-the minute threat intelligence, and publish reports that provide insight and guidance on the latest threats and how to mitigate them. Imperva is headquartered in Redwood Shores, California.

© 2014, Imperva, Inc. All rights reserved. Imperva, the Imperva logo, SecureSphere, Incapsula and Skyfence are trademarks of Imperva, Inc. and its subsidiaries. All other brand or product names are trademarks or registered trademarks of their respective holders. IMPV-WP-What-Next-Gen-Firewalls-Miss-0116-v1