O'REILLY®

Protecting Your Web Applications

Solutions and Strategies to Combat Cybersecurity Threats

Gary Sloper & Ken Hess

REPORT

Protecting Your Web Applications

Solutions and Strategies to Combat Cybersecurity Threats

Gary Sloper and Ken Hess

Beijing • Boston • Farnham • Sebastopol • Tokyo
O'REILLY®

Protecting Your Web Applications

by Gary Sloper and Ken Hess

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisition Editor: Nikki McDonald	**Copyeditor:** Octal Publishing Services
Developmental Editor: Virginia Wilson	**Interior Designer:** David Futato
Production Editor: Kristen Brown	**Cover Designer:** Randy Comer

April 2019: First Edition

Revision History for the First Edition

2019-04-24: First Release
2019-06-04: Second Release

The views expressed in this work are those of the authors, and do not represent the publisher's views. While the publisher and the authors have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the authors disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

This work is part of a collaboration between O'Reilly and Oracle Dyn. See our statement of editorial independence.

978-1-492-05278-4

[LSI]

Table of Contents

Preface

The rise of cloud computing, use of open source technologies, new data-processing requirements, complexity of web applications, and an increase in the overall sophistication of attackers have combined to create an extremely challenging environment for IT security leadership.

Given how critical websites, applications, and online services have become to supporting revenue and productivity, there is nothing more important for your business than ensuring that your digital assets are available and protected at all times. Consider the impact of cyberthreats on your business: customer loss, brand reputation damage or permanent loss of revenue, and team culture demise.

In this report, we examine the increasing cyberthreat landscape and take a detailed look at the major threat patterns businesses and security professionals currently experience. We explain how attackers have become so successful and offer remedies to prevent attacks and fix existing vulnerabilities. And, finally, we look at current and emerging trends in efforts to move to cloud-based security, outsourced services, and third-party hosting options.

CHAPTER 1
Today's Threat Landscape

In this chapter, we examine today's web application threat landscape, focusing on the major vulnerabilities and threats that cost businesses, and ultimately their customers, billions of dollars per year. We also look at an organization and its members who have taken on the task of gathering threat data and helping businesses prevent web application vulnerabilities. Finally, we discuss the current business impact that these threats have on revenue and reputation.

How We Got Here

In the early days of personal computing, boot sector viruses took the title of top threat to security. As the internet matured, so did the threats to privacy, to raw data, to financial data, and to money itself. The cybersecurity threat landscape looks very different today than it did just five years ago. And if you look at the numbers, the threat landscape has evolved even further from what it was just two-and-a-half years ago when ransomware was the most feared of all malicious cyberattacks. But the one threat that has remained since the beginning of the internet until today is web application attacks.

Cybersecurity Experts Respond to the Growing Threats

In the 2018 SANS Institute Incident Response Survey, business applications, which includes web applications, are the top system type involved in breaches (at 62.1%). Web application security is such a high-profile topic that in 2001, computer scientist and cybersecurity expert Mark Curphey founded the Open Web Application Security Project (OWASP) to provide unbiased information about application security. OWASP tools and documents are free and open to anyone interested in improving application security.

Web security remains one of the top concerns of businesses of all sizes. Add the ongoing threat to web security to the new landscape of cloud-based, Everything-as-a-Service (XaaS) offerings, and it's clear that the threat landscape is as big and diverse as the internet itself. The wave of public compute, storage, and other cloud assets moves the integrity of hub-and-spoke datacenters of the 1990s and 2000s with strict governance to a world in which cloud definitions can be defined differently per provider. From a technical perspective, security breaches are expensive to mitigate. The Ponemon Institute's 2018 Cost of a Data Breach Study: Global Overview reveals that the average cost of a data breach is $3.86 million and the average cost per lost or stolen record is $148. A company that suffers a data breach, on any scale, should prepare for significant revenue losses from legal fees, free or discounted services to affected customers, and reputation damage.

NOTE

OWASP is a not-for-profit international entity that is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted.

There are risks associated with exposing any application to the internet or even to internal users via corporate intranet portals. Security researchers, hackers, nation states, and various other malicious attackers continuously search for vulnerabilities and exploits for those vulnerabilities. According to Verizon's 2018 Data Breach Investigations Report, web applications top the list for types of breaches. Maintaining vigilance, keeping systems and applications patched, and providing best available perimeter protection still does not guarantee 100% security for any environment. Although these measures certainly don't hurt security, new vulnerabilities can still arise with every code upgrade, update, and patch.

Security professionals know that cybercriminals can take many paths to breach data, exploit vulnerabilities, and compromise security. Web-based applications are especially vulnerable because of the many support layers beneath the application: operating systems, web servers, database servers, application servers, and services not associated with the application at all. Developers and support personnel alike need to integrate security into every aspect of an application. Securing the application and the data behind it must take top priority at every step in the process.

To illustrate the extent of the focus on web security, the O'Reilly/Oracle Dyn survey "AI brings speed to security" (May/June 2018) reveals that 64% of the 445 respondents list "Hackers gaining access to our data via our websites, applications, or APIs" as their top concern. 37% of the respondents listed "Web application attacks and vulnerabilities" as their second greatest security concern. And in a close third place, 34% report that denial of service (DoS) and distributed denial-of-service (DDoS) attacks are a top issue.

Current Top Threats to Web Applications

You've set up a web application that you believe is secure and released it for public consumption. The service appears to be down. The customer or user becomes discouraged and turns elsewhere for satisfaction. What kind of threats can you expect to bombard that application and threaten your security? The threat landscape has broadened in recent years to cloud-based attacks, DDoS attacks, and massive email phishing campaigns. The web security threat landscape has also broadened with the greater threat landscape. Some of those threats remain constant, but have become more sophisticated, more aggressive, and have increased in frequency. For example, SQL injection (SQLi) attacks have remained the top web application threat for at least the past 10 years (OWASP 2010, 2013, and 2017 Top 10 Lists).

The following discussion provides an overview of web application threats. We've highlighted the types of damages caused by each and steps to prevent these attacks. While these attacks affect online shopping and retail businesses, all business types can be affected by similar attacks.

Bots and Botnets

There has been significant coverage of malicious *bots* and the harm they have caused organizations, even over a short amount of time. A nefarious *botnet* is a formidable enemy on the internet because it is a highly distributed network of connected bots. Bots are individual malware-infected computers that are not willing participants in botnets. The fact that these bots are random, diverse computers owned by innocent users makes them all the more dangerous. The danger lies in their geographic diversity. Their owners have no idea that their computers and internet connection bandwidth participate in attacks.

NOTE

Bad bots account for more than one-fifth of all internet traffic.[1]

Small to mid-sized companies face the same challenges as do larger ones, but without the equally large budgets to address them. These companies must do the best they can with what they have, and malicious actors know this and take advantage of it.

Botnets carry out attack campaigns such as massive spam floods, shopping cart and credit card frauds, DoS and DDoS attacks, brute-force hack attacks, identity theft, click fraud/digital ad fraud, web scraping, competitive data mining, account takeover, and credential stuffing.

Attacks can last from hours to days against a target and are generally aimed at extorting funds from the target. This section examines bot-related attacks associated with web applications.

Industries among the most vulnerable include gambling, airlines, finance, health care, ticket vendors, insurance, financial services, and tech.

Some industries are hit harder than others, but it's clear that none are safe. Over the past three years, analysis of empirical data for web traffic over hundreds of sites shows that between 54.4% and 61.3% of all web traffic is from actual human users. The rest is comprised of bots.

[1] Source: 2018 Bad Bot Report: The Year Bad Bots Went Mainstream by Distil Networks

NOTE

"Not all bots are malicious. For example, the bots used by internet search engines find and index web content to make it easier and more convenient to find the things we're interested in. The bad bots are the ones to be concerned about—and they accounted for between 18.6% and 21.8% of all web traffic over the last three years."[2]

Ecommerce Shopping Cart and Credit Card Fraud

Retail and online shopping sites are the most susceptible to *cart fraud* from bots because items selected for pending transactions are removed from inventory so that an item isn't sold twice. Because the transactions are fraudulent, inventories look lower than they are, causing legitimate customers to look elsewhere. When the transaction goes stale from a "no sale" status, the item returns to inventory. There are two reasons why cart fraud is costly: lost sales and inventory understock/overstock issues.

Bots that perpetrate credit card fraud (carding bots) often attempt a small, random charge that might go unnoticed by some. Charges for amounts such as $1.01 are probes to check the validity of a card before larger purchases are made.

Price Scraping

There's a threat that's almost as rampant as credit card-related theft: *price scraping*. This occurs when a bot places items into a shopping cart to reveal prices and discounts given on a dynamic basis. Dynamic pricing is an important online sales strategy used by ecommerce portals to influence consumer-buying behaviors.

Content and price scraping not only leads to the aforementioned inventory problem, but it also allows competitors to capture (scrape) pricing and discount levels, which can give them a significant advantage. The data scraper analyzes the site's dynamic pricing intelligence and can override this strategy to strengthen its own pricing and gain an unfair advantage over victims. The content part of the equation is about gathering a company's product catalog so that the scraper can offer the same exact product at a lower price.

[2] Source: *https://solutions.aberdeen.com/oracle_web_security*

There are proprietary tools to prevent price and content scraping that allow you to post prices and content without fear of unauthorized access or theft. Most of the tools available are so-called *bot protection tools*. Behind the scenes, these tools recognize “bot patterns” that attempt to mimic human interactions.

Click Fraud

Click fraud has multiple definitions. One definition is when someone increases their online popularity by buying “likes” or clicks on a web posting. The other definition—the one we use for the purposes of this report—is using a botnet to rack up ad costs with fraudulent ad clicks. Bots are especially effective at clicking an ad to record an “impression” and incurring an ad charge. There are multiple ways in which this type of fraudulent behavior can financially harm its victim (although there is generally no financial gain for any of the malicious parties involved):

Malicious intent
Malicious actors can launch a campaign to increase charges to an innocent advertiser.

Friends helping friends
Friends attempt to help a publisher by clicking ads to boost revenue to the publisher. When discovered, the publisher is often accused of click fraud.

Competitors
These fall into two groups: advertising competitors and publishing competitors. Advertising competitors want the advertiser to pay for irrelevant ad clicks. In the case of publishing competitors, the competitor wants the publisher to be accused of click fraud.

The use of botnets for this type of activity is obvious—the difficulty is in tracking down the perpetrator. The only party who suffers is the one who pays for the advertising to drive traffic to a site. The advertising party pays regardless of whether the clicks are valid, which hurts business and profits, and the advertising party could be accused of click fraud, which would result in reputation damage. These bots invoke fraud, which could mean thousands of pretend clicks for which the advertiser must pay.

Similar to other types of attacks in this section, a botnet prevention solution is necessary. Examples of botnet prevention include *anti-malware software* installed on every endpoint, enabled *host-based firewalls*, disabled *autorun features* (Microsoft Windows), disallowed *automatic trusts* between computers, *virtual local-area network* (VLAN) implementation, and implementing the principle of *least privilege* for all accounts—especially service accounts.

Distributed Denial-of-Service Attacks

A DDoS attack is typically a flood of legitimate-looking requests that tie up computer resources to the point where legitimate requests go unanswered. DDoS attacks are not like other attacks in that they are not vulnerabilities in the traditional sense. A “normal” vulnerability is one that is present through an error in coding or configuration. The DDoS attack takes advantage of a different kind of vulnerability—changing the signal-to-noise ratio in favor of noise. For example, during a college football game a few years ago, the home team fans were so loud that the opposition’s players couldn’t hear the plays correctly and subsequently lost the game. After the game, the opposing coach commented that the fans were truly the “twelfth man” on the field. The action by the fans was a type of DDoS attack against the opposing team. They made so much noise that the signals couldn’t get through.

NOTE

DDoS attackers commonly use bots to act as their agents. Bots comprise systems that unknowingly participate in botnets that might include thousands of systems.

DDoS attacks can take the form of *distraction attacks*, meaning that the DDoS attack is a big fire to put out when the real menace lurks just below your radar, compromising systems or services.

DDoS attackers disrupt your service until the malicious payload successfully infects your systems, and then they disappear back into the internet’s traffic stream. You might not realize that another attack has occurred for months.

NOTE

Sometimes attackers will launch a DDoS attack to draw attention away from another attack. While security focuses on the noisy DDoS issue, attackers successfully exploit some other vulnerability, using the DDoS attack as a smokescreen.

Credential Stuffing

In a credential stuffing attack, a malicious actor purchases or extracts a set of user credentials and then employs a botnet to test those credentials against websites. This attack succeeds because people tend to reuse usernames and passwords on multiple sites. Open web forms are the most vulnerable because they don't offer any other validation such as a human verification or a two-factor option. These types of forms are highly vulnerable to credential stuffing.

The financial sector is a prime target for fraudsters. A June 2018 Ponemon Institute report ("The Cost of Credential Stuffing: Asia-Pacific") states that there were more than 30 billion malicious login attempts from November 2017 to June 2018. The attacks mostly originated from the United States, Russia, and Vietnam.

Retail sites are also vulnerable because most do not implement multifactor authentication. Multifactor authentication is a basic defense against these types of attacks. Attackers depend on sites that only use username and password authentication. A second factor, no matter how simple, is a good deterrent.

According to respondents to the Ponemon study, credential stuffing attacks lead to costly application downtime, customer loss, and expensive IT and security team remediation tasks.

Here's a quick summary of the Ponemon study:

- Companies experience an average of 12 credential stuffing attacks each month in which the attacker successfully identifies valid credentials.
- The volume and severity of credential stuffing attacks are increasing.
- It's difficult to differentiate criminals from legitimate users.
- Participants feel that cloud migration leads to increased risk of attacks.

- Companies have insufficient technologies or solutions for preventing and containing credential stuffing attacks.

Other Common Web-Based Attacks

The ecommerce-related attacks we've covered thus far, while common, are higher profile than the ones listed in this section. These attacks are just as common, but they receive little press even though they are no less significant in terms of financial losses due to stolen records and damaged reputations. A DDoS attack, for example, is big news, but SQL injection attacks rarely make media reports.

These types of attacks do hit news feeds when the size of the stolen or compromised data set is large enough to warrant it. Rarely, if ever, do the standard news outlets mention the mode of compromise to include terms such as SQL injection, XSS, or session hijacking. This section familiarizes you with these very dangerous but preventable exploits.

SQL Injection

SQL injection is an attack resulting from poor user data entry validation or other poor coding practices (e.g., a web form that allows a user to input untrusted data, tricking the application into executing unintended commands). Injections can be SQL queries, PHP queries, lightweight directory access protocol (LDAP) queries, and operating system commands.

Malicious users allowed to enter open-ended input into a web form, without any coding protection or input sanitizing, can launch injection attacks that result in data theft, data exposure, data loss, data corruption, denial of access, and host takeover. Security researchers find that injection flaws are very prevalent, especially in legacy code. Attackers find and exploit vulnerable code using scanners and *fuzzers*, which are software applications specifically designed to find such coding flaws.

Cross-Site Scripting

A *cross-site scripting* (XSS) attack is a type of injection that involves placing malicious scripts into websites. The attacker uses a web application to send malicious code to a user in the form of a browser-side script. XSS is the second most prevalent issue in the OWASP Top 10 Report for 2017. It's found in close to two-thirds of all applications. If you choose to rely on automated tools for detecting this vulnerability, realize that they will detect only some XSS problems—generally limited to those in PHP; Java 2 Platform, Enterprise Edition (J2EE); JavaServer Pages (JSP), and ASP.NET technologies. However, automated exploit tool frameworks can detect and exploit all three types of XSS. Exploitation frameworks and tools are readily available and many are free of charge and open source.

To illustrate how prevalent XSS attacks are, high-profile companies such as Facebook, Google, and PayPal have been focused on addressing this threat with their R&D to protect customers. Even though XSS is a type of injection, it does not attack the web application itself, as do regular injection attacks. Rather, the XSS attack infects web application users. These types of attacks target users to steal their credentials.

There are three types of XSS, and they typically target users' browsers:

Reflected XSS
: The application or API includes unvalidated user input as part of HTML output.

Stored XSS
: The application or API stores nonsanitized user input that is viewed at a later time by another user or by an administrator.

DOM XSS
: JavaScript frameworks, single-page applications, and APIs that dynamically include attacker-controllable data to a page are vulnerable to DOM XSS.

Most XSS attacks target users' browsers and are known as *client-side attacks*. Attackers might steal user sessions, take over a victim's accounts, bypass multifactor authentication, replace or deface Document Object Model (DOM) nodes (JavaScript HTML elements), spawn malicious downloads, log keystrokes, and so on.

Trusted User Session Hijacking

Session hijacking is a variant of the *man-in-the-middle attack* in which the attacker has access to the network via a rogue connection or through a compromised system. This type of attack is an active rather than a passive attack. This means that the attacker not only uses tools to collect data through *network sniffing*, but also must take an active role in using that information to disrupt an ongoing session—hence the term "hijack." There are two types of session hijacking: application and network. *Application hijacking* occurs when an attacker steals or predicts the valid session token. The attacker gathers (sniffs) HTTP network traffic to find a valid ongoing web session.

Prior to an actual hijack session, which can be labor-intensive and can increase the risk of exposure, the attacker sniffs the network for unencrypted protocols such as FTP, HTTP, Telnet, and the Berkeley r-commands such as `rlogin`, `rcp`, and `rexec`. These protocols send information in plain text, which is human readable as it's sniffed from network traffic. These protocols are low-hanging fruit that attackers use because they don't need to do any real work to gain access to a username and a password.

There are monitoring tools that detect new application installations on workstations, but they don't find and identify so-called "portable" applications that run without the requirement for a formal installation. An attacker can download portable applications and freely run them without detection because they are standard network tools available to anyone. One example is Wireshark Portable, a cross-platform network protocol analyzer that is useful in network troubleshooting. But like any good tool, malicious users and attackers use its powerful capabilities to do reconnaissance on networks to find exploitable weaknesses.

NOTE

Session hijacking is the act of taking over an ongoing, active connection between two nodes on a network. It requires that the intruder have access to the network because session hijacking requires a combination of sniffing and spoofing tools. User session hijacking is also known as *cookie side-jacking*.

Threats and Impacts to Business

You don’t need to look far to find victims of any of the web application attacks previously described. Large retail businesses, financial institutions, government entities, medical facilities, and even security companies have fallen prey to these attacks. Every victim has something significant to lose when a breach occurs. Retailers lose revenue due to down time. They lose customers because they’re seen as vulnerable. And the losses can pile up over time as more investigations take place that uncover stolen customer credit card data, personnel information, and damages to systems. The resulting losses can grow to tens of millions of dollars.

Likewise, financial institutions, government sites, and medical facilities all have experienced huge reputation damage and tremendous personally identifiable information (PII) theft. Stolen PII can result in identity theft and fraud that costs businesses and consumers billions of dollars each year. A few high-profile security companies have shuttered their doors due to embarrassing hacks and data exposure.

The losses to businesses and consumers alike are significant. The dollar-per-incident costs are overwhelming. And the damage to reputation is often irreversible.

Web application security is so important to an internet-based society that global groups have formed to focus on protecting businesses and consumers from attacks, fraud, and breaches. When the average cost of a breach is $3.86 million, it doesn’t take too many to add up to disaster for businesses and shareholders. Retailers pay a direct cost in loss of revenue and customer loyalty. Other businesses and government entities can experience both direct and indirect costs from breaches. Some indirect costs come in the form of identity theft, credit card fraud, and the release of confidential information. The potential for these types of costs exists as long as the data is available to the highest bidder.

Some less obvious costs to businesses occur through the purchase of new hardware, new software, and new services to mitigate, remedy, or deter future losses from breaches. Cleanup operations might take several months to a year or more and consume precious resources from other areas of a business.

Conclusion

By now, you understand the direct and the indirect financial burdens that breaches cause. You also have a feel for the scope and the depth of the threat landscape facing internet web applications. After the lid is off our data, putting it back on is costly, time-consuming, and has far-reaching implications for customer loyalty and business reputation.

The web application threat landscape is large, complex, and ever-changing. To stay ahead of new threats, security professionals must continuously work on addressing vulnerabilities and protecting against the methods attackers use to steal data and disrupt commerce. Unfortunately, we can’t depend on any single entity to protect our web applications and our data. We must use a multilayer approach to security inside and outside of the corporate network. The next chapter explores some of these protection strategies.

CHAPTER 2

Threat Protection Strategies

The threat protection strategies we present here are solutions to the problems presented in Chapter 1. Remember that no security strategy is permanent, and regular reviews of your security posture are a must. The threat landscape continuously changes, and protection strategies must change with it.

Cyberthreats come in all forms. But they have one goal in mind: to invoke a malicious outcome for the end user or organization. Even if you decide to use outsourced services, you must still maintain vigilance in your local environment to ensure that endpoints (laptops, tablets, and phones) aren't adding to the threat. Each of the solutions that follow has a place in an overall security strategy, and none can stand alone. What's required is an integrated approach to your web application, corporate network, and individual security.

The Security Operations Center

The purpose of the security operations center (SOC) is to detect, protect, mitigate, train, monitor, and remediate when necessary. It is made up of a team of highly trained security analysts and system administrators who use their expertise and some very expensive security tools to keep a company's data safe. Small to mid-sized companies face the same challenges as larger ones, but without large budgets to address them. These companies must do the best they can with what they have. Malicious actors know this and take advantage of it.

The SOC is a necessity at the large-enterprise level, but for small to mid-sized companies, it's an expensive luxury. At the same time, malicious actors know that mid-sized companies are high-value targets. There's enough information in some of these companies to keep an intruder happy for months. That information can include high value data like PII, intellectual property, proprietary code, drawings, diagrams, credit card data, health information, and dozens of other data types that attackers can sell on the dark web.

Organizations that produce intellectual property are high-value targets, as are government contractors, health care facilities, manufacturers, and security companies. Malicious actors love to pilfer data from security companies, especially high-profile ones, because it shows off their power.

The huge expense of spinning up an on-premises SOC is out of the question for many companies because of the considerable resources required for such an undertaking. The solution is to outsource the SOC function to a third party like a "SOC-as-a-Service" company. This type of service provides the skills and watchful eyes that smaller companies need, without the huge internal outlay for skilled resources, hardware, software, and training. Additionally, many SOC-as-a-Service companies monitor and protect customers' assets all day, every day.

Another positive aspect of a managed security service is that the SOC doesn't work in a vacuum. A threat at one client's location is immediately communicated to all client companies.

The consumption of any third-party as-a-service offering is like any other—you must do some shopping to find which one fits you and your company's needs.

NOTE

SOC-as-a-Service providers are also known as *managed security service providers* (MSSPs). Some of these providers offer a full range of on-premises and cloud-based consulting services. They might also offer a staff augmentation option—basically for-hire cybersecurity contractors who work in your office, which is a good option if your regulatory restrictions prevent you from outsourcing off-premises.

Outsourcing SOC duties is a lower-cost alternative to creating one yourself. With a third-party SOC provider, the resources are already in place. Some charge based on the amount of data ingested by their sensors each day. Others charge based on the number of users, endpoints, or sensors being used.

Should you decide that a managed security service is an option for your company, be sure to ask these questions when considering an outsourced partner:

- Is the service intrusion detection or protection?
- Does it offer incident response and remediation services?
- Does it monitor 24/7/365?
- What is its mean-time-to-detect for intrusions?
- Is its service agentless, or will you need to deploy agents on systems?
- Is there a dashboard that that you can monitor at your location?
- What is its response time and protocol for an incident?
- What type of metrics will you receive?
- What effect will its service have on your cybersecurity insurance?
- What Service-Level Agreement (SLA) options do you have?
- What is its data retention policy?
- What protection does the company have for itself?

The last question in the list might seem odd, until one considers that attempts on security companies are common and persistent. You need to know that your service provider will keep your data safe.

An outsourced, third-party SOC is a good option for companies without a budget to create one from scratch. However, outsourcing your SOC comes with a few cautions. First, you have no control over the SOC, its detection methods, its notification speed, or its remediation speed or path. Second, you have no input regarding hiring practices or delegation of resources. In other words, the SOC could be staffed with relatively untrained technicians. You need to decide which limitations you can tolerate in exchange for the convenience and cost savings of an outsourced SOC.

Web Application Firewalls

Web application firewalls (WAFs) are a strong defense against XSS, SQL injection, and cross-site forgery. A WAF is a Layer 7 (Application Layer) defense. It is not a security panacea. It is one piece of a layered approach to security—specifically application security. There are many types of attacks that it cannot defend against. The WAF acts as a reverse proxy, meaning that clients pass through the firewall before reaching the application server. The firewall filters out malicious traffic via a set of rules or policies.

Like other firewalls, WAFs can take one of three different forms:

Host-based
: These are typically integrated into the application itself. They require time and expense to implement.

Network-based
: Implementation requires the deployment of a hardware WAF appliance. This is the most expensive option.

Cloud-based
: Implementation of a third-party WAF is done with minimal upfront costs and with little effort from the customer.

There are pros and cons to any of these options. But the cloud-based option is very appealing because of its low entry costs, quick time to deploy, access to immediate updates, and ease of implementation. Plus, in the case of workloads within a datacenter, the WAF is protecting threats prior to routing to that node, preventing latency and saturation. Cloud-based options aren't for everyone. Some organizations still prefer to manage a hardware/network-based option. In-house solutions are still viable, but they do require ongoing maintenance, expandable storage, and upfront expense. Government contractors who work on defense projects are not good candidates for cloud options because of regulatory compliance mandates that deal with confidentiality, export controls, and restricted information.

Bot Management Solutions

Bot management plays a significant role in building and maintaining corporate defenses. Although bot protection is not the silver bullet in edge protection, it's a very good start—especially for those who conduct ecommerce and other critical transactions via their web apps. The key is making it a comprehensive solution, with human interaction and ongoing stewardship to address the issue (or prevent one). These bot detection mechanisms can help prevent nefarious bots from wreaking havoc on your site:

JavaScript challenge
: This is sent to every client, attacker, and real user. Legitimate browsers will pass the challenge without the user's knowledge, whereas bots, which are typically not equipped with JavaScript, will fail and be blocked.

Human interaction challenge
: This identifies normal usage patterns for each web application based on legitimate user or visitor behavior analysis, and provides customizable security postures for bots that deviate from the standard usage behavior, activity, or frequency.

Good bot whitelisting
: This gives users the ability to recognize and remember good bots and allow them access.

CAPTCHA
: This is a challenge intended to differentiate between computers and humans. In general, scripted bots are unable to solve the CAPTCHA and repeat the words and numbers used, whereas humans are easily able to do so.

Bot traffic shaping
: This is a traffic control mechanism used to detect and delay traffic created by suspicious bots, while at the same time prioritizing and whitelisting authorized traffic.

Device fingerprinting
: This generates a hashed signature of both virtual and real browsers based on more than 50 attributes. These proprietary signatures are then used for real-time correlation to identify and block malicious bots.

The threat information that comes from the aforementioned detection mechanisms is extremely helpful to the SOC. Unfortunately, it is challenging for an SOC to gather this information by itself. One option is to subscribe to threat intelligence services, which allow MSSPs to incorporate their data into corporate alerting services.

This is a great first step to utilize the latest information and act on it quickly and methodically.

An Integrated Approach

There is no single security panacea for web applications. The threat landscape is too large and too varied for a single solution. We suggest using an integrated approach to all security issues, but specifically for those related to web applications. A combination of secure programming, data encryption, WAFs, operating system security, least-privilege user security, segmented networks, and so-called "demilitarized zones" for corporate hosted, internet-facing applications to name a few.

An integrated approach is good news for the business consumer because it means vendor lock-in is not an issue. Vendor lock-in occurs when organizations are bound to a single vendor because it offers a one-size-fits-all proprietary solution. No single company does everything well. Addressing security needs with integration in mind is a better method of serving customers and protecting assets.

Conclusion

Today's threat landscape is too large, too complex, and changes too quickly to approach it with a single strategy or solution in mind. It requires automation, best practices-based implementation, strong software solutions, and the right people to manage those resources. An integrated approach is the best remedy for maintaining vigilance and implementing a multilayer security strategy in a business environment under constant attack.

CHAPTER 3

Threat Prevention Technology

"An ounce of prevention is worth a pound of cure."

Benjamin Franklin must have had a premonition of today's cybersecurity threat landscape about 280 years ago when he said that, because it's still true: remediation is far costlier than prevention. For example, while SQL injections are totally preventable, they are quite costly to repair. If a single incident results in the release of 5,000 customer records at an estimated $148 per record, the cost is $740,000. Preventing the SQL injection vulnerability would have almost no cost because the remedy is simple: allow no user-generated input into forms. Working from the planning stage through to deployment to maintenance with security in mind provides the required ounce of prevention.

Not all security vulnerabilities are as simple to prevent as a SQL injection, but the cost of prevention is a tiny fraction of the cost of a single breach. Remember: There's no direct correlation between threat severity and prevention expense. Each threat is different and must be approached individually.

NOTE

There is a rise in multivector attacks—those that combine multiple types of DDoS attacks into a single assault. From a mitigation or remediation perspective, you should separate and focus on each type of attack individually.

In this chapter, we explore the technologies that you should include in your protection strategy—bot management, artificial intelligence (AI), and machine learning—and we offer concrete prevention and mitigation methods for common web-based attacks.

Artificial Intelligence and Machine Learning

NOTE

AI is computer programming that provides a system or systems with the capability of making decisions that mimic human intelligence based on programmed experiences. Machine learning is a subset of AI because those systems can make decisions based on patterns and inferences that are not explicitly programmed into the system.

One intriguing direction in application security (and security in general) is that of machine learning and AI. For example, next-generation WAFs use AI to dynamically and automatically update security postures to protect web applications from vulnerabilities. Machine learning algorithms and big data analytics combine to inspect web traffic in real time to identify threats and behavior anomalies.

AI and machine learning have replaced the old resource-intensive application learning methods. Application learning methodology allowed too many false positives because there is no good method of accounting for every variation of normal application activity and usage. With machine learning, a new approach takes the place of application learning's observational model by replacing it with a statistical model to determine "normal" usage versus anomalies. If an anomaly is detected, more analysis reveals whether that abnormal behavior is a threat or is benign activity.

One example of AI and machine learning in action for security-related issues involves watching behavior patterns. "Intelligent" systems can spot nonhuman behavior patterns and distinguish those from human behavior patterns. More specifically, an AI system can detect a live attacker based on behavior patterns and can separate those patterns from a malicious script that is running on a system. It can also ignore the actions of legitimate users and scripts. The intelligent system can take different actions based upon its behavioral observations in the moment.

As AI and machine learning technologies continue to evolve, threat detection and prevention will also improve. Also, the larger the samples from which these technologies pull their statistics, the lower the number of false positives.

Prevention and Mitigation Methods for Web-Based Attacks

Online applications and their users are especially vulnerable to certain types of attacks due to many factors, such as the sheer volume of transactions, a disparately skilled user base, and an ever-growing threat landscape. Financial and health care sites also get their share of attacks. These tips, although generally oriented toward retail sites, can also be used for other industries.

Injection Prevention and Mitigation

SQL injection attacks are easy to prevent by using standard secure programming techniques and security measures for infrastructure. The OWASP SQL Injection Prevention Cheat Sheet describes these techniques in detail for developers and system administrators. Here are some of the recommendations:

Prepared statements
: These prevent any user-initiated queries or alterations to the SQL.

Stored procedures
: For example, prepared statements; procedures are part of the database code called by the application.

Whitelist input validation
: This limits possible variations of input to a few preselected options.

Escaping all user-supplied input
: Escaping cleans any open-ended user input, but is far less effective in preventing injections.

Least privilege
: This principle requires that privileges granted to the user are the fewest required to carry out the assigned task.

Other preventative techniques include:

- Consider using a data subset rather than the entire database for external users.
- Encrypt database connection strings.
- Change default security settings.
- Use LIMIT where possible to reduce the number of possible exposed records.
- Avoid wildcard statements such as SELECT * FROM.
- Change application account passwords often.
- Create custom error codes that don't expose table structures.
- Use a web application firewall (WAF).

Most of this preventative advice is basic security practice. But when a breach occurs, we're always surprised at how many default security settings are still in place or how many test and development passwords and queries went into production.

Cross-Site Scripting Prevention and Mitigation

The same rules apply to preventing cross-site scripting (XSS) as they do to other vulnerabilities—use secure coding practices and make securing your application the top priority. Scan your applications with the same tools used by malicious attackers. Use exploitation frameworks and freely available security tools to find vulnerabilities in your code. In other words, see what the attacker sees.

Some programming frameworks are better, by default, at preventing such vulnerabilities. The latest Ruby on Rails and React JS, for example, are two frameworks that automatically escape XSS. However, don't rely too heavily on a framework's built-in tools and capabilities. You still need to do your security due diligence by carefully examining your code using multiple tools and techniques.

As stated previously, secure programming techniques prevent many of the attacks described in this report. The same is true to keep reflected and stored XSS flaws from creeping into your web applications. Although there's a large number of XSS attack vectors, you can prevent these threats by following these few simple rules. The OWASP XSS Cheat Sheet provides 13 programming rules that are easy to implement and should be included in every web application.

- Never insert untrusted data except in allowed locations.
- HTML escape before inserting untrusted data into HTML element content.
- Attribute escape before inserting untrusted data into HTML common attributes.
- JavaScript escape before inserting untrusted data into JavaScript data values.
  - HTML escape JSON values in an HTML context and read the data with `JSON.parse`
  - JSON serialization
  - HTML entity encoding
- CSS escape and strictly validate before inserting untrusted data into HTML-style property values.
- URL escape before inserting untrusted data into HTML URL parameter values.
- Sanitize HTML markup with a library designed for the job.
- Prevent DOM-based XSS (DOM-based XSS has its own Cheat Sheet).
- Use the HTTPOnly cookie flag.
- Implement content security policy.
- Use an auto-escaping template system.
- Use the X-XSS-protection response header.
- Properly use modern JS frameworks like Angular (2+) or ReactJS.

We won't dig deep into each of these rules, but you can see a pattern in them: Escape untrusted data. *Escaping* means to ignore special characters by preceding them with "escape" characters. Escape characters vary depending on the programming language. For example, in some scripting languages such as PHP, programmers use the "\" to escape characters introduced in the course of normal programming or by malicious users.

There are some flaws that you can't fix by using escaping, such as allowing JavaScript code to run from an untrusted source. The rule here is to deny all untrusted script elements and then selectively allow input as needed. As the OWASP team suggests, the first three rules might be sufficient for your organization, and you don't necessarily need to implement all 13 to adequately protect applications.

Session Hijacking Prevention and Mitigation

Encrypting all traffic in user sessions using HTTPS is one simple method of thwarting network traffic sniffers. Ban the use of all unencrypted protocols on the network unless they're sent over a Secure Sockets Layer (SSL) tunnel. The risk of exposing usernames, passwords, and other valuable information is too high. An attacker, having a short amount of time to find a valid, ongoing web session, will look only for unencrypted streams.

Security personnel should teach users to close their browsers after completing their application work because this destroys the session. An attacker can steal or predict a session cookie only during an ongoing session. If the user leaves the browser open after a session has completed, the session cookie is still active.

Conclusion

Web-based applications are under constant threat from attackers. In this chapter, we offered strategies to prevent and mitigate the impact of those attacks to protect data and users from various types of fraud and theft. AI and machine learning are both promising technologies that should relieve some of security professionals' burden because of the vast amount of data that can be scraped and analyzed in an automated fashion.

CHAPTER 4

Next Steps for Businesses

Some companies have already moved their threat protection to the cloud and their security services to third parties. Those that haven't are likely considering a migration away from on-premises hosted applications and on-premises security management. For many companies, it's a question of control. Relinquishing control over infrastructure, security, and personnel is difficult. In this chapter, we discuss our predictions about how these transitions might take place for companies of different sizes.

Moving to the Cloud

We believe that companies will transition security services and web application threat protection to reside closer to application in the cloud. The trend toward moving security to the cloud is not a surprising prediction. But what's surprising is how we'll get there. The transition is a multiphase one. A first step is private cloud. Web applications will continue to reside in corporate demilitarized zones in the short term. As companies grow, demands on infrastructure grow, and further commoditization of cloud services continues, the public cloud, or at least a hybridized version of it, will prove too compelling to ignore.

Moving security services and support closer to where the applications reside makes sense on multiple fronts. For example, companies won't need to have a demilitarized zone, which is a security problem because it provides a certain amount of access into the corporate network. After services and security are moved to the cloud, corporate security can be tightened to allow only outgoing access because there are no corporate-hosted services that require access through a corporate firewall. This move greatly enhances internal network security.

Another example is that criminal hackers, hacktivists, and advanced persistent threat groups might infiltrate or compromise a portion of a corporate network, but the highly secure business applications will be protected off-site and separately from other internal corporate assets. Responsibility for data theft will shift to the third-party providers who are responsible for protecting their customer's data.

The transition from traditional, internally supported web applications and internal security to the cloud and to third-party providers is the direction many businesses have taken. But this does not shift all of the responsibility, compliance requirements, or damages to a third party in the case of a breach or a compromise. Although some downsizing of IT and security departments is a possible side effect of a cloud initiative, it will not altogether alleviate the need for in-house trained professionals. Businesses must retain trained security and IT professionals to monitor, inspect, and occasionally audit their third-party providers.

We foresee, over the next three to five years, that large companies will transition toward cloud-based security, managed security services, and support models—transferring the bulk of their compute, hosting, and security operations to third-party providers. Small to medium-sized businesses, being more agile and less entrenched in on-premises solutions, will make the transition much faster and with fewer barriers. Startup, cloud-native, and so-called "virtual" companies will launch in the cloud and likely never own or control their own infrastructures. All security, IT, and web application services will live entirely in the cloud from day one.

Third-Party Outsourcing

We also believe that this transition to the cloud will include a move to outsourced services, such as SOCs. Again, this move will also begin as a hybrid scenario in which companies will augment their in-house SOCs with outsourced ones to attain 24/7/365 monitoring, protection, reporting, and remediation of incidents. The complete transition to a fully outsourced solution might take several years to complete. A 100% reliance on outsourced services requires that company officers and technicians relinquish a certain amount of control of their computing environments to third parties. We recognize that this is not an easy transition.

The size of a company has a significant impact on the speed of this transition. The move to outsourced services will occur at different rates depending on how large a company is, how it's been in business, and how much control over infrastructure, services, and people the company is willing to relinquish. Smaller and newer companies will make the move to outsourced services with fewer conflicts. New companies will use commoditized third-party resources to get started and remain agile.

Conclusion

Moving threat protection to third-party entities and to the cloud should result in better coverage, fewer incidents, and lower costs. The benefits to online shoppers, brick-and-mortar retail customers, financial institutions, and health care facilities are better fraud protection, reduced incidents of identity theft from online leaks, better privacy protection, and a smaller target surface for attackers when the corporate network is removed from the picture.

Web application attacks are on the rise. The attacks are more sophisticated and use more brute-force attack strategies than seen in previous years. Organizations must continually examine and reexamine strategies for protection, mitigation, and remediation. To stop web application attacks, organizations need to deploy a multilayer approach to security that includes WAFs, multifactor authentication, artificial intelligence, machine learning, secure programming, and big data analytics.

About the Authors

Gary Sloper is a Vice President at Oracle Dyn. Gary brings over 20 years of experience to his leadership of the global solutions engineering and customer success teams. His organization architects and implements cloud-based Edge Services, including providing deliverability and security services to help customers monitor, control, and optimize their CDN and hybrid cloud workloads.

Kenneth “Ken” Hess is a full-time system administrator and a freelance technology writer and journalist. He writes on a variety of topics including security, virtualization, Windows, open source software, databases, storage, and networking. In his spare time, Ken is an avid and award-winning filmmaker and a dabbler in the visual arts.