Brought to you by:

Web Application Firewalls (WAFs)

for dummies®
A Wiley Brand

Recognize essential WAF capabilities

Integrate WAF into application architecture

Understand WAF deployment models

Lawrence Miller
Chad Wise
Matthew Wedlow

F5 Special Edition, Updated

About F5

F5 (NASDAQ: FFIV) is a multi-cloud application security and delivery company that enables our customers — which include the world's largest enterprises, financial institutions, service providers, and governments — to bring extraordinary digital experiences to life. For more information, go to f5.com. You can also follow @F5 on Twitter or visit us on LinkedIn and Facebook for more information about F5, its partners, and technologies.

Web Application Firewalls (WAFs)

F5 Special Edition, Updated

by Lawrence Miller, Chad Wise, and Matthew Wedlow

for dummies®
A Wiley Brand

Web Application Firewalls (WAFs) For Dummies®, F5 Special Edition, Updated

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-90351-2 (pbk); 978-1-119-90352-9 (ebk)

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub.

Publisher's Acknowledgments

Project Editor: Susan Pink
Acquisitions Editor: Ashley Coffey
Editorial Manager: Rev Mengle
Business Development Representative: Molly Daugherty
Content Refinement Specialist: Saikarthick Kumarasamy

Introduction

Web application attacks are a top pattern in security incidents and data breaches, according to Verizon's 2021 "Data Breach Investigations Report." Modern web applications take advantage of highly distributed multicloud environments, dynamic microservices architectures, and third-party integrations and content. These complex application deployments are inherently more difficult to protect against a constantly and rapidly evolving threat landscape, despite industry efforts to bolster secure application development practices.

A web application firewall (WAF) prevents successful attacks against your web applications by

- Providing virtual patching for code and application-level vulnerabilities
- Inspecting ingress application traffic to identify and block scans, attacks, and bots
- Inspecting egress application traffic to prevent sensitive data exposure
- Securing application programming interfaces (APIs)

About This Book

Web Application Firewalls For Dummies consists of five chapters that explore the following:

- Why protecting web applications in the current threat landscape is so challenging (Chapter 1)
- What essential capabilities a WAF delivers and how a modern WAF differs from other firewalls (Chapter 2)
- How to integrate a modern WAF with different application architectures (Chapter 3)
- How to deploy a modern WAF to protect your web applications (Chapter 4)
- What to look for in a modern WAF for your organization (Chapter 5)

Each chapter is written to stand on its own, so if you see a topic that piques your interest, feel free to jump ahead to that chapter. You can read this book in any order that suits you (though we don't recommend upside down or backwards).

Foolish Assumptions

It's been said that most assumptions have outlived their uselessness, but we assume a few things nonetheless!

Mainly, we assume that you're a developer, network or security manager, DevOps engineer, or security engineer. As such, we assume that you understand basic networking and security fundamentals and technologies.

If any of these assumptions describe you, this is the book for you! If not, keep reading anyway! It's a great book and after reading it, you'll know how to secure your web applications with a WAF.

Icons Used in This Book

Throughout this book, we occasionally use special icons to call attention to important information. Here's what to expect:


REMEMBER

This icon points out important information you should commit to your nonvolatile memory, your gray matter, or your noggin — along with anniversaries and birthdays!


TECHNICAL STUFF

If you seek to attain the seventh level of NERD-vana, perk up! This icon explains the jargon beneath the jargon and is the stuff legends — well, legendary nerds — are made of.


TIP

Tips are appreciated but never expected — and we sure hope you'll appreciate these useful nuggets of information.


WARNING

These alerts probably don't point out the stuff your mother warned you about, but they do offer practical advice to help you avoid potentially costly or frustrating mistakes.

Beyond the Book

There's only so much we can cover in this short book, so if you want to learn more about securing your web applications with a modern web application firewall, check out `www.f5.com/solutions/application-security`.

IN THIS CHAPTER

- Leveraging the open-source community and APIs
- Using encryption to provide cover for threats
- Exploiting known software vulnerabilities
- Understanding the evolution of bots and botnets

Chapter 1
Recognizing the Current Threat Landscape

In this chapter, you survey the constantly evolving threat landscape and discover how attackers take advantage of the relentless pressure put on application development teams to deliver better software faster, and how they hide threats in encrypted Internet traffic. You also see how vulnerabilities become exploits that lead to breaches, and how botnets are used to automate an attack at a massive scale.

Application Building with FOSS Components

For modern businesses competing in the digital economy, agility and rapid time-to-market are critical to success in application development. Many companies have embraced Agile development methods and DevOps cultures to further accelerate the development and deployment of their business-critical applications. In many companies, continuous integration/continuous deployment (CI/CD) pipelines enable multiple code releases every day.

Application development teams build applications in much the same way that car manufacturers build cars. When possible, car makers use a supply of parts, for example, tires and airbags, to build cars faster. Application developers use components, such as containers, microservices, and frameworks. Many of these components are available as free, open-source software (FOSS) components. Although the open-source community is a great supplier of innovation and generally does a good job of self-policing source code to ensure that vulnerabilities are quickly identified and patched, the use of FOSS components creates a greatly expanded attack surface that extends across a much larger supply chain.

TIP

The Synopsys 2021 "Open Source Security and Risk Analysis (OSSRA)" report states that 84 percent of codebases contain vulnerable open-source components.

Closely related to the concept of reusable FOSS components in the software supply chain, businesses are increasingly participating in the application programming interface (API) economy to facilitate integration and faster time to market. APIs have become a critical component of modern applications and virtually all new applications are built with accessibility via an API. In some instances — open banking, for example — APIs are the vehicle for monetization.

API calls are similar to general web requests but in a different context. However, like the pages of a website, APIs are susceptible to exploits, such as injection, and abuse, such as credential stuffing. One key difference is in their structure: the schema, protocol, and content. Because APIs aren't intended for direct user interaction, they may not be in the purview of security teams. That's especially true of third-party API calls buried deep in application logic.

TECHNICAL STUFF

An *injection* exploit introduces unexpected data or code in an application, a database, or an API to expose a vulnerability. For example, if a database expects a 4-character numeric entry and an attacker instead enters a 40-character text entry, the database may unexpectedly crash or expose a vulnerability. Even if the exploit doesn't cause the database to crash, the returned error codes may help an attacker determine the backend operating systems and application types that are being used and further refine the attack vectors. A WAF can be used to mask or alter returned error pages. *Credential stuffing* is a type of automated attack that

attempts to guess a valid username and password on a target network using lists of compromised user account names and passwords (or password hashes) commonly purchased on the dark web.

REMEMBER

Open-source software components and APIs make application developers' lives easier by significantly speeding the pace of development. But they also change your risk management profile because you can't use the same security controls as those used for software developed in-house (such as test-driven development). API management and security needs to be implemented at strategic points in the development pipeline.

Unfortunately, speed and security rarely co-exist in harmony. The push to deploy new code releases as quickly as possible often leads to the incorrect or non-secure use of FOSS components and APIs. CI/CD pipelines that automate application development and deployment often lack adequate security testing. This omission often causes friction between application developers and security teams because of the perception that time-consuming security testing processes are obstacles to agility. And the testing that is conducted, such as unit testing and regression testing, typically focuses on functionality rather than security, potentially allowing new vulnerabilities to be created or re-introducing previously patched vulnerabilities in an application update. Additionally, threats from insecure software supply chains are a growing concern and a new risk category in the 2021 OWASP top 10.

REMEMBER

As evidenced by the rapidly growing volume of security incidents and breaches due to misconfigured APIs, the management and security of APIs needs to be prioritized for protection.

Threats Hiding in Encrypted Traffic

Practically every website today uses Hypertext Transfer Protocol Secure (HTTPS) to encrypt Internet traffic. In 2014, Google called for "HTTPS everywhere" to make the Internet more secure. To drive rapid adoption of HTTPS, Google began using it as a ranking signal in its search algorithm. An unfortunate consequence of this well-intentioned goal is that malicious activity and threats can now hide in encrypted traffic.

TIP

Despite widespread industry adoption of HTTPS everywhere, F5 Labs analysis shows that HTTPS misconfigurations can happen to organizations of any size — a single certificate error can take out an entire platform, affecting millions of customers. You may have a huge blind spot if you aren't looking at your encrypted web traffic.

Most traditional network firewalls (discussed in Chapter 2) cannot decrypt payloads and inspect only the first few bytes of a network packet to determine source and destination IP address, port, and protocol information. These firewalls can determine application types based only on port and protocol information (assuming the applications use standard ports and protocols); nothing in the payload can be inspected. Next-generation firewalls (NGFWs, also discussed in Chapter 2) are capable of decrypting payloads, but NGFWs are typically deployed as a traditional network perimeter firewall protecting corporate traffic and preventing data loss, rather than as a reverse proxy protecting web applications. Traditional network firewalls and NGFWs also suffer significant performance degradation from resource-intensive decryption and encryption operations.

Application Vulnerabilities and Exploits

Shortly after a software vulnerability is discovered, attackers begin scanning the Internet using automated tools to find web servers and applications that haven't been properly patched or updated to protect against the vulnerability. Figure 1-1 shows an example of how attackers take advantage of a known vulnerability (that is, a one-day exploit) to attack a target that has not been properly patched, in this case, the 2017 Equifax breach.

In late 2021, a CVE (common vulnerabilities and exposures, a method for publicly sharing information on cybersecurity vulnerabilities and exposures) was released for some versions of Apache's widely used Log4J2 logging utility. The open-source components are vulnerable to an exploit that can result in code execution from third-party LDAP servers without proper checks. Almost immediately after CVE publication, attackers began to scan the Internet looking for vulnerable systems.

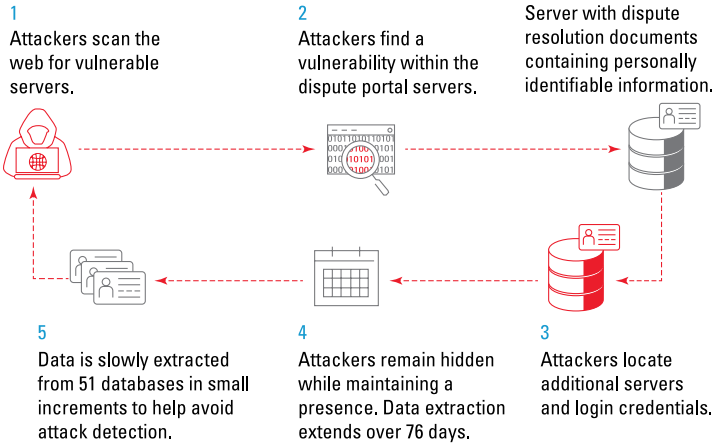

FIGURE 1-1: How attackers used a one-day exploit in the 2017 Equifax breach.

Not all attacks are zero-day threats (that is, newly discovered vulnerabilities). If high-value information can be stolen by exploiting known vulnerabilities, that's where attackers will invest their time, underscoring the need for security to "shift left" in the software development lifecycle to gain visibility into potential vulnerabilities earlier.

Ultimately, organizations are responsible for safeguarding customer data because the people in those organizations will have to deal with the fallout if a breach occurs.

Malicious Bots

Security and risk teams also need to protect from abuse of critical business logic, such as logon, create account, and transfer money functions. In modern applications, these functions are commonly available, and therefore exposed, through APIs and third-party integrations. Although shopping carts and other interactive customer functions are not software weaknesses or vulnerabilities, these endpoints are inherently vulnerable to abuse through automated attacks (such as credential stuffing). These attacks are often carried out by bots (*robots*) or malicious software programs and scripts, which can ultimately lead to compromise, account takeover (ATO), and fraud.

Bots were designed to do simple, mundane, repetitive tasks on the Internet that humans don't want to do or can't do as quickly. For example, bots were used in the early 1990s to crawl the Internet to improve early search engine results. Other examples of good bots include the following:

- **Chatbots** interact with humans through text or sound. One of the first uses of chatbots was for online customer service and text-messaging apps such as Facebook Messenger and iPhone Messages. Siri, Cortana, and Alexa are a few prominent examples of chatbots.
- **Shopbots** scour the Internet looking for the lowest prices on items you're searching for.
- **Monitoring bots** check on the health (availability and responsiveness) of websites. Downdetector.com is an example of an independent site that provides real-time status information, including outages, of websites and other kinds of services.

Beyond enabling automation, modern bots are often designed to simulate human behavior. Bots account for a significant amount of Internet traffic, with estimates ranging from 21 percent to more than half of all traffic today. F5 often finds that malicious automation accounts for more than 90 percent of traffic to online retailers.

But bots have a dark side as well. A *malicious bot* is a computer that has been infected by malware that allows an attacker to control the computer. Malicious bots are used as part of a broader automated network (known as a *botnet*), often consisting of tens of thousands of other infected bots, controlled by an attacker through command-and-control (C2) servers. The botnet can also be used to attack other targets. According to a recent Verizon "Data Breach Investigations Report," in the same way automation may be helping you scale up your defensive operations, it may also help attackers scale up their offense.

Malicious bots and automated attacks are increasingly difficult to detect and defend against. Moreover, they provide attackers with a lucrative return for minimal investment, which makes them a go-to favorite in attackers' arsenals. Some examples of how malicious bots and botnets are used in an attack follow:

- **Distributed denial-of-service (DDoS) attacks:** Attackers can use botnets to overwhelm an application or network, causing it to crash or perform poorly and thereby preventing authorized users from accessing the application or network. Today, many DDoS botnets are made up of infected Internet of Things (IoT) devices instead of PCs. With literally billions of vulnerable IoT devices on the market, attackers can build massive botnets to carry out enormous DDoS attacks that flood web applications and networks with more than one terabyte per second (Tbps) of traffic.
- **Credential stuffing:** Taking advantage of billions of stolen account credentials, attackers use bots to launch automated attacks by "stuffing" stolen username and password combinations in the logon pages of different websites (see Figure 1-2). The goal is account takeover (ATO) — and because so many people use the same credentials for many accounts, the success rate and payoff for attackers are high.

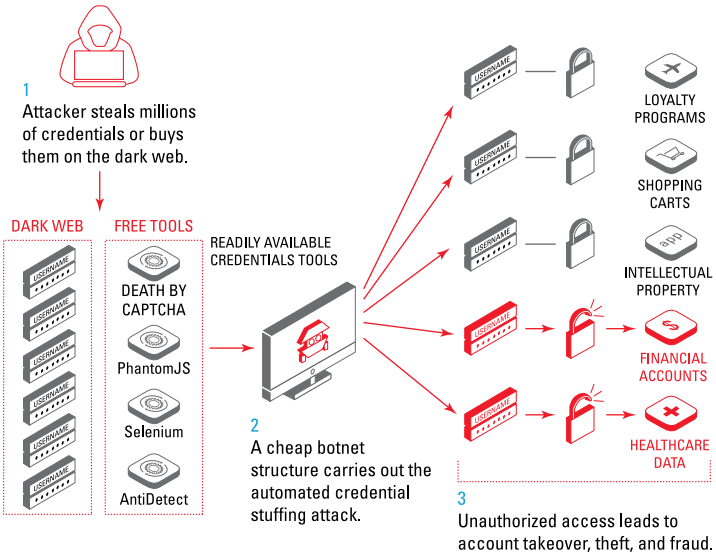

FIGURE 1-2: A botnet credential stuffing attack.

- **Gift and credit card fraud:** Attackers use bots to break into gift card accounts looking for credentials; then they create counterfeit cards and steal the cash value of the card. In the case of credit cards, attackers use bots to test stolen credit card credentials with small transactions that don't usually attract attention (such as a nominal $1 charge). Upon success, hackers use the stolen credentials to make large purchases or drain credit accounts.
- **Spam relay:** Spam email is frequently used to confirm harvested email addresses and spread malicious website links. Other spam-like behavior includes filling inboxes with unwanted email containing malicious links, writing fake product reviews, creating fake social media accounts to write fake or biased content, racking up page views (for example, on a YouTube video) or followers (such as on Twitter or Instagram), writing provocative comments on forums or social media sites to stir up controversy, and rigging votes.
- **Web scraping protected content:** This attack method scans and extracts (steals) copyrighted or trademarked data from websites, stores it locally, and then reuses it — often for competitive purposes — on an attacker's own website(s). Scraped data can include intellectual property, product information, and product prices. Airline, hospitality, online gaming, and ticketing websites are particularly vulnerable to web scrapers.
- **Click fraud:** This type of botnet attack involves advertising fraud — the fraud being that a bot, not a human, is clicking an ad and therefore has no intention of purchasing the advertised product or service. Instead, the goal is to boost revenue for a website owner (or other fraudster) who gets paid based on the number of ads clicked. Such bots skew data reported to advertisers and cost companies a lot of money because they end up paying for non-human clicks. Even worse, these companies get no revenue from fake shoppers. Click fraud can be used also by companies to deliberately drive up the advertising costs of their competitors.
- **Auction sniping:** Bad actors place perfectly timed, last-minute bids on goods or services on online auction sites to prevent humans from bidding or winning an auction.

- **Intelligence harvesting:** This type of botnet is used in the initial reconnaissance phase of the attack lifecycle and involves scanning web pages, Internet forums, social media sites, and other content to find legitimate email addresses and other information that attackers can use for spam or phishing emails or fraudulent advertising campaigns.

WARNING

Malicious bot traffic can also drive up your cloud-based infrastructure costs because the cost of many cloud resources are consumption based.

WAFs provide a critical stopgap against vulnerabilities and malicious automation that looks to exploit them, and can integrate with specialized bot defenses to deter sophisticated attacks such as credential stuffing that could otherwise lead to fraud.

In Chapter 2, you learn about the important role of web application firewalls (WAFs) in protecting your business-critical web applications in this increasingly sophisticated and hostile threat landscape.

IN THIS CHAPTER

- Identifying core WAF requirements
- Looking at other security tools
- Defining the modern WAF

Chapter 2
Understanding WAF Basics

In today's digital economy, web applications are a primary target for attackers. Traditional security tools don't provide adequate protection for your business-critical applications. In this chapter, you learn about the core capabilities of a web application firewall (WAF), why traditional security tools aren't enough, and how a modern WAF can protect your web applications and add real business value for your organization.

Looking at Essential WAF Capabilities

Attackers target web applications because they're the gateway to your valuable data. A WAF protects applications by applying a set of rules to the Hypertext Transfer Protocol (HTTP) or Hypertext Transfer Protocol Secure (HTTPS) sessions that web applications use for communication.

TECHNICAL STUFF

A WAF protects web applications from a variety of application layer attacks, such as

- **Cross-site scripting (XSS):** A web application attack used to gain access to private information by delivering malicious code to end users via trusted websites.
- **Injection:** A security exploit in which an attacker takes advantage of an application that does not validate, filter, or sanitize user-supplied data. For example, with SQL injection, an attacker can gain access to back-end database data or application data or both. This request can cause unintended and malicious behavior by the targeted application.
- **Cookie poisoning:** An attack where cookies are intercepted before they return to the server to extract or modify information. Forged cookies can also be created to impersonate a user to access additional user data.

A WAF enforces a set of policies that help determine what traffic is malicious and what traffic is safe. Just as a proxy server acts as an intermediary to protect a web client, a WAF operates in the reverse (called a reverse proxy), acting as an intermediary that protects the web app. A modern WAF protects your web apps by filtering, monitoring, and blocking any malicious HTTP/S traffic traveling to the web application and prevents any unauthorized data from leaving the application.

RICACORP PROPERTIES LIMITED STRENGTHENS WEBSITE SECURITY

Faced with ever-evolving cybersecurity threats, Hong Kong's third-largest property agency, Ricacorp Properties Limited, needed to strengthen the protection of its main business website. By deploying an F5 WAF solution on Microsoft Azure, the organization boosted security without compromising user experience or reliability.

Business Challenges

Ricacorp, with about 220 branches and over 2,600 staff over the territories, receives hundreds of thousands of web visits daily from customers seeking property information, surveys, and mortgage referral services.

In the wake of increasing security threats, Ricacorp came to F5 to strengthen the security of its website infrastructure on Microsoft Azure with additional on-premises infrastructure. The team required a tailored solution that could be seamlessly deployed across their cloud, on-premises, and mobile environments.

Solution

After evaluating several competitors, Ricacorp selected F5 for its unique value proposition of being able to deliver every app anywhere — and securely. F5 worked closely with Ricacorp's partner to understand Ricacorp's challenges and recommend a solution optimized to their needs.

"As cyberattacks continue to increase and evolve at an alarming rate, we needed to find a reliable partner who could help us take care of this major concern. Lucky for us, F5 was an easy choice," said Dennis Tam, Associate Director at Ricacorp. "Given their existing relationship with our partner, Expert System, F5 was a natural fit. They have delivered much-needed security, as well as providing peace of mind."

The real estate giant selected WAF technology on Azure to solve their issues. F5 delivered application protection for their public website with consistent policy deployment in the cloud and on premises, as well as proactive bot defense to prevent bad bots from retrieving important property information.

"Thanks to F5, we now have a robust and agile WAF solution across multiple cloud platforms and our on-premises data center, which has given us a consistent level of security and led to increased customer confidence," added Tam.

Another reason Ricacorp selected F5 was the availability of future deployments of mobile protection solutions using the F5 Mobile SDK. The final phase in their security improvement program will be to deploy Mobile SDK for their mobile channel protection.

Benefits

By implementing an F5 WAF on Azure, Ricacorp stops attacks from any location, ensuring application availability. Its customers can conveniently access property information with peace of mind that their data is secure. Key benefits include the following:

- **Consistent policies across multicloud environments:** With F5's WAF, Ricacorp was able to deploy the same security policy across

(continued)

(continued)

Azure and its on-premises data center and maintain consistency across multicloud environments, reducing operational complexity and the risk of policy gaps.

- **Scalable defense to promote business innovation and agility:** One of the major benefits Ricacorp has realized from using F5's solution is the freedom to grow their on-premises and cloud infrastructures according to business priorities, with the confidence that their apps will be protected without being tied to a particular deployment model or cloud platform.

  Future mobile protection solutions using the F5 Mobile SDK: Mobile apps do not support the same security capabilities as web browsers. F5's Mobile SDK extends the protection capabilities of F5's WAF solutions to mobile applications to defend against bots, vulnerability scanners, content scraping, and other automated attack vectors by safely whitelisting sessions coming from the protected mobile app.

- **Catering to the Open API trend:** The Open API Framework is one of seven initiatives announced by the Hong Kong government in late 2017 to prepare Hong Kong to move into a new era of digital economies. By deploying F5 solutions, Ricacorp can now connect securely with banks and third-party service providers to provide innovative and integrated services, moving in step with the Open API Framework initiative and allowing customers to enjoy one-stop service and a better customer experience.

Comparing Common Firewall Architectures

You may be thinking, "I already have a firewall. Why do I need a WAF?" To answer this question, let's compare some different firewall (and intrusion prevention system) technologies.

Network firewalls

A traditional network firewall operates at Layers 3 and 4 (network and transport layers, respectively), and simply inspects traffic and enforces rules based on an Internet Protocol (IP) packet's source and destination IP address, port, and protocol. A traditional

firewall doesn't inspect the payload to determine if network traffic is malicious.

The effectiveness of a traditional network firewall is further limited because it identifies an application based on only its port number, which can be changed easily to evade detection.

Next-generation firewalls

A next-generation firewall (NGFW) is a network security platform that fully integrates traditional firewall and network intrusion prevention capabilities with other advanced security functions that provide deep packet inspection (DPI) for complete visibility; accurate application, content, and user identification; and granular policy-based control. An NGFW enforces user-based policies and adds context to security policies. It may also include capabilities such as web content filtering, antivirus or antimalware protection, and an intrusion prevention system.

NGFWs provide some application-aware features and can also stop some known injection attacks such as cross-site scripting (XSS) and Structured Query Language (SQL) injection. But an NGFW still relies on passive filter detection and doesn't examine every HTTP request. Instead, it works much like an intrusion prevention system, sampling requests and examining their first few bytes, not the full request payload. As a result, application layer bypass attacks against NGFW technologies are common. Plus, IP address reputation feeds implemented on NGFWs and other firewall technologies have proven ineffective against botnets and other automated threats.

Intrusion prevention systems

An intrusion prevention system (IPS) detects and automatically blocks suspected network or host intrusions. IPS is typically signature based, meaning it checks for known vulnerabilities and attack vectors based on a signature database.

In general, IPS operates at Layers 3 (network) and 4 (transport) and protects traffic across a range of protocol types, such as the following, but is not necessarily context-aware in terms of normal versus malicious content within the protocols:

- Domain name system (DNS)
- Simple Mail Transfer Protocol (SMTP)

- Telnet
- Remote Desktop Protocol (RDP)
- Secure shell (SSH)
- File Transfer Protocol (FTP)

Protecting Your Organization with a Modern WAF

WAFs were created to address the problem of web application servers running code that was vulnerable to a myriad of known attacks, particularly XSS and injection. WAFs have improved over the years, but they're still largely based on passive, filter-based methods to detect malicious payloads and check for protocol compliance in web requests.

A modern WAF, however, is an active security control that is capable of interrogating client endpoints and dynamically strengthening the security posture of web applications. A modern WAF employs countermeasures to detect and stop evolving application-layer threats. At a high level, a modern WAF integrates behavioral analysis and machine learning to more completely assess the threat associated with any given client session.

By profiling a baseline of normal application traffic behavior, anomalous traffic patterns become easier to spot. Just as automation has increased an attacker's capabilities, artificial intelligence and machine learning can differentiate normal from anomalous traffic in ways that aren't possible by a human security engineer. A modern WAF uses advanced analytics and machine learning to generate dynamic signatures that block malicious traffic — without administrator intervention.

Using JavaScript injections to assess whether or not a client is a browser with a human user, a modern WAF establishes a client fingerprint and enables easier detection of bots and other automated tools. With client fingerprints, attackers can also be tracked beyond an IP address.

The proactive bot defense of a modern WAF challenges each client session, detecting the nature of the client as well as differentiating friendly bots from malicious ones (see Figure 2-1). The challenges are transparent to the user, thereby reducing or eliminating the effect on the user experience (UX).

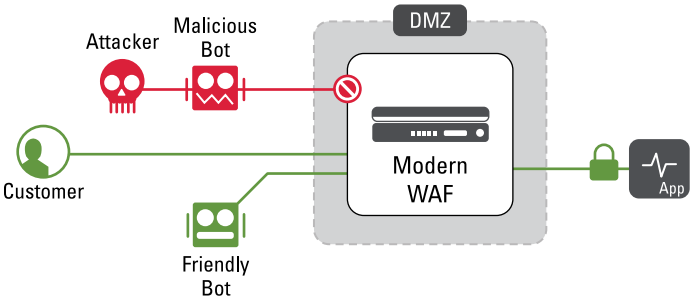

FIGURE 2-1: A modern WAF detects bots without requiring the use of server agents or dedicated appliances.


TIP

A modern WAF can also protect your application programming interfaces (APIs) from the same attacks that target your web applications by automating the creation of custom rules specific to each exposed API. A modern WAF deployed in front of your application or integrated into several components of a containerized application can help you manage API security more effectively.

A modern WAF provides the following:

- **Advanced application protection:** Combines machine learning, threat intelligence, and deep application expertise.
- **Proactive bot defense:** Protects apps from automated attacks by bots and other malicious tools.
- **Mobile software development kit (SDK):** Protects mobile apps from bots and abuse via an allow list, behavioral analysis, secure cookie validation, and advanced app hardening.
- **In-browser data encryption:** Encrypts data at the application layer to protect against data-extracting malware and man-in-the-browser attacks.
- **Behavioral denial of service (DoS) protection:** Provides highly accurate Layer 7 (application) DoS detection and mitigation.

- **API protocol security:** Deploys tools that secure Representational State Transfer (REST), JavaScript Object Notation (JSON), Extensible Markup Language (XML), and Google Web Toolkit (GWT) APIs.
- **Defenses for the Open Web Application Security Project (OWASP) top 10:** Defends critical applications from today's biggest security concerns, which are listed in the OWASP top 10 risks.
- **Leaked credential protection:** Prevents the use of known compromised credentials.

IN THIS CHAPTER

- Protecting traditional multitiered web applications
- Taking advantage of cloud-native technologies
- Spanning hybrid and multicloud environments
- Keeping mobile apps protected

Chapter 3
Integrating WAFs with Application Architectures

In this chapter, you explore the different ways in which a web application firewall (WAF) can be deployed and integrated in different on-premises, cloud, and hybrid environments.

Traditional Applications

Traditional multitier web applications deployed in on-premises data centers are a logical place to start deploying a WAF because the WAF deployment scenario is relatively easy to understand. A WAF can be deployed not only to protect your on-premises web applications but also to accelerate the performance of your web applications by, for example, performing SSL/TLS offloading and decryption for your web servers.

Many traditional applications were developed long before the advent of the cloud and Agile development methodologies. However, if the business application is critical, it is likely now part

of the application development backlog and therefore requires ongoing protection from vulnerabilities and exploits. As code is updated, breaking functionality in the application is always a concern, particularly with legacy applications that may not be as well documented or understood as more modern cloud-native applications. Also, a legacy application may use outdated components that have known vulnerabilities that were patched and forgotten long ago. Without thorough regression testing of not only functionality but also security, an old vulnerability can be exposed once again, creating an opportunity for attackers.

Additionally, many organizations plan to move some of these traditional applications to the cloud. This migration sometimes requires refactoring of the original code to ensure compatibility and performance in the cloud. During the refactoring process, new vulnerabilities may be introduced or old vulnerabilities re-introduced.

A WAF can be integrated into the application development pipeline to ensure that software updates do not expose known vulnerabilities and other security risks, such as those listed in the OWASP (Open Web Application Security Project) top ten web application security risks, because any code change can introduce new weaknesses or vulnerabilities.

Microservice Design Patterns

Modern cloud-native applications typically take advantage of a highly distributed microservices architecture. These applications may leverage thousands of microservices and runtimes — deployed across multiple public or private clouds — on new and evolving technologies, such as containerization and serverless computing. In many cases, these microservices are ephemeral, sometimes running only a few seconds. As discussed in Chapter 1, many modern applications leverage shared open-source components and application programming interfaces (APIs), which may not be as well vetted as an organization's own secure application development pipeline.

A modern WAF can be deployed to automatically identify and protect all the components of a cloud-native application, regardless of their location or how long they persist. Integrating a

modern WAF in your continuous integration/continuous deployment (CI/CD) pipeline also helps to ensure that code-level vulnerabilities are not introduced into the microservices architecture of your business-critical web applications and allows the security policy to baseline, stabilize, and adapt throughout the application lifecycle.

Hybrid and Multicloud Environments

A modern WAF must provide flexible deployment options to protect web applications wherever they run — whether on-premises, in the cloud, or across hybrid and multicloud environments. Often, multiple virtual instances of a modern WAF must be deployed as close as possible to the web application and its various components and work together to provide comprehensive end-to-end security for the web application.

A modern WAF also helps organizations control their costs in the cloud. When you're paying for every inbound request to your web applications — including requests from scanners, bots, denial-of-service attacks, and fraud (see Chapter 1) — being diligent and expedient when implementing security controls can save you money. Additionally, a modern WAF can simplify your hybrid and multicloud environments by allowing you to centrally configure and implement a uniform set of controls across your entire digital estate, rather than being beholden to the often native and proprietary controls of individual cloud providers.

Mobile Applications

Mobile applications have become an increasingly important part of every modern business's application portfolio. Today, more users browse the Internet from their mobile devices than their desktop computers.

Perhaps more so than other web applications, mobile applications require additional vigilance on the part of businesses. Personal mobile devices store some of the most private and sensitive information about individuals, including contacts, addresses, text messages, photos, videos, and banking information. Yet despite the sensitive nature of the data on our mobile devices, most

mobile users do not run antimalware protection on their devices and instead operate under the fallacy that their phones are inherently secure. Because many mobile apps leverage third-party integrations and functionality, such as a mobile phone's camera or global positioning system, businesses must ensure that their mobile applications do not become an attack vector leading to the compromise of individual mobile devices and the data stored on them. Given that mobile apps do not natively support JavaScript, an SDK is needed for bot defense.

AUTOMATING SECOPS AND DEVOPS DEPLOYMENTS

Applications are the heart of a company's profits, so new releases and updates must make it to market before their competition. How do you roll out new applications, update existing ones, and maintain a safe environment for your customers? By adopting SecOps practices in your DevOps culture to shorten your time-to-market while automating security checking practices (shifting security to the left). With the integration of SecOps, you no longer have to wait on old ticketing processes. You can now trigger policy changes from your pipeline (policy enhancements pushed by Git) or from security information and event management (SIEM) reporting and ticketing systems (such as ServiceNow).

Just as you use automation to accelerate an application's speed-to-market, you can leverage the same tooling to deploy security and policies for applications sitting behind your WAF. A modern WAF should be fully configurable using existing automation tools (such as Ansible or Terraform) as well as a Representational State Transfer (RESTful) API, which allows the programmability to be expanded to multiple methods.

Integrating these practices into your CI/CD pipeline provides automated and consistent deployments while making sure your security policies function as expected. Test procedures and artifacts by testing live within your application repository, which now becomes the centralized location for all documentation containing the who, when, and why of changes that are made. This testing also helps DevOps teams "fail fast" so application issues can be addressed rapidly and redeployed into the pipeline.

IN THIS CHAPTER

- Thinking about your web application environment and security resources
- Doing it yourself for maximum flexibility
- Getting fast out-of-the-box functionality with SaaS
- Partnering with a managed service provider

Chapter 4
Deploying WAFs

A web application firewall (WAF) can be deployed in several ways — it all depends on where your applications are deployed, how much architectural flexibility you require, and how you want to manage the WAF. In this chapter, you discover some important considerations that can affect your decision, as well as your different deployment options.

Addressing Key Considerations

When deciding which deployment option makes the most sense for your organization, the first question to consider is where your web applications are deployed. Are they all located in an on-premises data center? Are they in a public cloud? Or are they distributed across multiple clouds in a microservices architecture? The answers to these questions (you may have numerous web applications that use a variety of deployment scenarios) will help you determine whether an on-premises or cloud-based WAF is best for your organization and whether it should be a physical or virtual appliance.

You also need to consider the level of involvement you can commit to deploying and managing your WAF. A WAF isn't necessarily difficult to deploy and manage, but like any tool, you'll get more out of it when you put more into it — whether that means using the time and expertise of your existing security team or using a managed service provider.


TIP

According to the F5 "State of Application Services Report," more than three-quarters of organizations are modernizing their apps. This pervasive modernization effort, compounded by an industry skills gap, creates challenges for organizations looking to retrofit apps and operate the necessary security tools to safeguard confidential data. The skills gap is further exacerbated by the challenge of providing security parity across all application architectures and infrastructures — in many cases, across multiple cloud providers.

Going the Self-Managed Route

Whether you deploy on-premises or in a cloud environment, a self-managed WAF gives you complete, granular control of your WAF so you can tune it to meet your unique business needs. However, the self-managed model requires involvement from your security team and app owners to deploy and build the security policies that will be applied to your applications.

If you have a mature security organization with the necessary skills and expertise, going the self-managed route provides the most flexibility and may be the right choice for your organization.

Opting for a Cloud-Delivered (SaaS) Solution

A software-as-a-service (SaaS) WAF enables you to cut capital costs and operating overhead without compromising security. With a similar feature set as an on-premises WAF, this option provides out-of-the-box protection from application vulnerabilities and attacks. Without infrastructure overhead such as

hardware or software updates to manage, a SaaS-based WAF is a perfect fit that allows teams to integrate security in application development with relatively minimal effort.

The SaaS option provides flexibility and security policy portability for multicloud deployments, while allowing you to retain control of your traffic management and security policy settings. This option can also help you meet your most demanding deployment models where architectural flexibility, performance, and advanced security concerns are paramount.

Working with a Managed Service Partner

Working with a managed service partner is perhaps the easiest way to get started with a WAF in the cloud. Autoprovisioning allows you to deploy a security policy that meets your needs in a relatively simple and cost-effective manner to enable instant protection for your web applications and data. A fully managed WAF gives you access to 24/7 support so you can augment your own in-house security resources with a service that's wholly set up, deployed, and maintained in a security operations center (SOC).

Although fully managed service offerings can get your WAF up and running faster than other deployment models, you may not have as much architectural flexibility. Some offerings might not give you direct administrative control over your security policies. A fully managed service is also typically more expensive, but it should still be less expensive than hiring additional full-time staff if you don't have necessary skills and expertise on your security team.

IN THIS CHAPTER

- Adding a WAF to your security ecosystem
- Customizing your WAF to address unique web application requirements
- Keeping your WAF up-to-date and leveraging machine learning
- Preventing credential abuse and data theft
- Ensuring application development security
- Implementing security best practices

Chapter 5
Ten Key WAF Considerations

Following are ten points to think about when considering a WAF to protect your web applications:

- **A WAF should work with a traditional Layer 4 or next-generation firewall (NGFW), not replace it.** Traditional firewalls and NGFWs protect your users and network traffic. A WAF protects your web applications. It should be a part of your security ecosystem, helping to protect your entire attack surface.
- **A WAF should integrate with other security tools.** Integrations with other security tools such as intrusion prevention systems (IPS), vulnerability scanners, and security information and event management (SIEM) platforms are essential to ensure that your WAF doesn't become another siloed point security solution for your security team to manage.

- **A WAF should be customized to specific applications.** WAF policies and rules should be customized to meet the unique security, compliance, and performance requirements of your different web applications, as needed.
- **WAF policies are not "set it and forget it" rules.** WAF policies must be regularly updated to match application changes and remain effective against new attack vectors. A modern WAF offsets this manual burden by automating protections through AI-based anomaly detection and false positive suppression and by adapting the security policy in the face of attacker retooling and evasion.
- **A WAF should have behavioral learning capabilities.** Artificial intelligence and machine learning technologies enable a WAF to identify normal application behavior and adapt when anomalies, possibly indicating malicious activity, are detected.
- **A WAF should prevent credential abuse and protect personally identifiable information (PII).** Credential abuse is a major cause of application breaches, leading to unauthorized disclosure of PII and other sensitive application data.
- **A WAF should inspect both ingress and egress traffic.** A WAF acts as a reverse proxy to protect web applications against malicious inbound traffic. However, your WAF should also inspect outbound traffic to ensure that an application vulnerability hasn't been exploited and sensitive data isn't being exfiltrated.
- **A WAF should be DevOps friendly.** Your WAF should be part of your continuous integration/continuous deployment (CI/CD) pipeline, preventing code-level vulnerabilities from being deployed to production environments.
- **A WAF should protect your application programming interfaces (APIs).** The API economy is integral to application development and third-party integrations, so it's a rich target for attackers. Your WAF should provide robust protection for your web applications as well as your APIs.
- **The Open Web Application Security Project (OWASP) recommends using a WAF.** The OWASP Foundation (`https://owasp.org`) is an industry leader in web application development security and recommends that organizations deploy a WAF to protect against web application security risks.

Protect Your Applications and APIs with Best-in-Class Security

F5 application security solutions protect against application vulnerabilities, malicious bots, and sophisticated manual attackers while supporting great user experiences. That's why F5 is trusted to protect over 25,000 enterprise customers, including 48 of the Fortune 50.

Flexible and Portable

Deploy security controls for your apps across any infrastructure with consistent policies, whether on-premises or in public or private clouds.

Shift Left

Enable faster code pushes and eliminate deployment hassles by integrating security as code into existing dev pipelines.

Supports Modern App Architectures

Insert security into the architecture that makes sense for you, whether you're protecting monolithic applications or microservices environments.

Comprehensive Protection

Protect against the OWASP Top 10, common vulnerabilities, DDoS attacks, API attacks, account takeover attacks, and more.

Learn more: **f5.com/solutions/application-security**

Protect your apps and APIs with a modern WAF

Always-on, always-connected apps can help power and transform your business — but they can also act as gatekeepers to the data beyond the protection of your traditional security controls. With most attacks targeting applications — because that's where the money is — protecting the capabilities that drive your business means protecting your apps. In this book, discover how a modern WAF can help you mitigate risk, protect your apps, and prevent software vulnerabilities from being exploited.

Inside...

- Discover how modern app trends create risk
- Learn about different firewall types
- Protect cloud-native apps and microservices
- Deploy in hybrid and multicloud environments
- Get started quickly with a SaaS-based WAF
- Explore fully managed WAF options
- Evaluate key WAF capabilities and features

Lawrence Miller has written more than 200 Dummies books on numerous technology and security topics. **Chad Wise** is a Sr. Solutions Engineer for F5 and a retired United States Air Force Veteran. **Matthew Wedlow** is a Solutions Engineer for F5 whose background is in network security and data center design.

Go to Dummies.com™
for videos, step-by-step photos, how-to articles, or to shop!

ISBN: 978-1-119-90351-2
Not For Resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.