IMPERVA®

BUYER'S GUIDE

How to Select the Right Web Application Firewall:

10 Key Requirements to Consider

Executive Summary

Why You Should Read this Guide

Web applications are an essential part of today's interconnected world, providing end users with easy access to social, financial, legal, health, employment, and other online services. And, web applications are often the primary gateway to valuable or sensitive data.

As a result, web applications are a primary target of cyberattacks.

Indeed, it's not uncommon for there to be weekly reports of hacktivist attacks on corporate and government sites, including DDoS attacks against organizations, or massive web breaches compromising credit card numbers and personal data records. Moreover, hidden behind the front page headlines are tens of thousands of unreported breaches—unexplained website outages, temporary website defacements, and small-scale fraud incidents.

Cyber-attackers are constantly seeking out applications to compromise, disable, or deface. Their targets are often content management systems, database administration tools, e-commerce stores, online sales systems, hosted websites, and SaaS applications. Their weapons of choice are technical web attacks such as SQL injections, business logic attacks, distributed denial-of-service (DDoS) attacks, and online fraud attacks.

Unfortunately, network security defenses, such as firewalls, intrusion detection systems (IDS), and intrusion prevention systems (IPS), are largely incapable of preventing such attacks since they are unable to prevent SQL injections, DDoS attacks, evasion techniques such as encoding and comments that elude IPS signature detection, or other HTTP-based attacks. Consequently, organizations face potential data theft, lost revenue, damaged client, customer, or vendor relationships, revoked licenses, legal proceedings, and financial penalties.

What's needed is the right web application firewall.

Web application firewalls are hardware, software, virtual and cloud-based solutions that, unlike other network security solutions, understand web application usage patterns and validate inputs to stop malicious attacks before they can do harm. They block scanners and automatically patch application vulnerabilities. And they rapidly learn application behavior and automatically evolve to prevent new attacks and keep critical applications safe.

However, a web application firewall is not a one-size-fits-all option.

Organizations must carefully evaluate a web application firewall's deployment, configuration, management, and security capabilities to ensure it meets their web application security needs and is an integral part of an evolving application and IT infrastructure.

This guide, targeted to IT security staff, provides an overview of the threats to web applications and items to consider when selecting a web application firewall. It also details minimum feature and functionality requirements, and provides an at-a-glance checklist for evaluating web application firewalls.

What's Inside

Threats to Web Applications

Cyber-attackers consider web applications high-priority targets for a variety of reasons, including:

- Complexity of source code—For example, integration of third-party libraries and use of cutting-edge protocols or technologies increases the possibilities for exploitable vulnerabilities and code manipulation.
- Ease of execution—Most web application attacks can be easily automated and indiscriminately launched against thousands or even tens or hundreds of thousands of victims, using readily available tools and without needing to infiltrate the victim's internal network.
- Potential for high-value rewards—Attackers can, for example, sell sensitive private data stolen during a successful code manipulation or gain access to sensitive accounts through credential compromise.

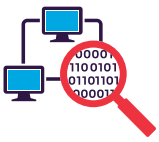
Cyber-attackers use four primary weapons of choice—technical web attacks, business logic attacks, distributed denial-of-service (DDoS) attacks, and online fraud.

Technical Web Attacks

If hackers were surveyed about their favorite attack vectors, technical web attacks, like SQL injection and cross-site scripting (XSS), would undoubtedly top the list. This assumption is borne out by analyses of hacker forums and application attack traffic. In fact, SQL injection alone accounted for almost one fifth of all hacker forum discussions.[1] And, according to security research, nearly two-thirds of organizations experienced one or more SQL injection attacks that evaded their firewall and other perimeter defenses in the past year (while the average time to detect these attacks was an astounding 140 days).[2]

To accelerate the rate of technical web attacks, cybercriminals have become "industrialized." They leverage a combination of off-the-shelf attack toolkits, infected 'bots', and search engines to quickly find and exploit web application vulnerabilities. The industrialization of hacking has made technical attacks much more automated and dangerous.

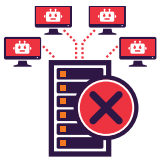
Business Logic Attacks



Hackers aren't stopping at traditional web attacks. Business logic attacks and fraud are also becoming increasingly popular techniques. Today, hackers exploit business logic flaws to post advertisements in online forums.

They scrape websites for valuable intellectual property. They perform repeated brute force attacks. They use wildcards in search fields to bring applications to a screeching halt. These attacks have left many organizations at wits' end, because application scanners cannot detect business logic flaws and secure development processes can't always mitigate them.

Distributed Denial of Service (DDoS) Attacks



An increasingly common cyber threat is a distributed denial of service attack (DDoS) that—as its name implies—renders websites and other online resources unavailable to intended users. DDoS attack threats come in many varieties, with some directly targeting the underlying server infrastructure. Others exploit vulnerabilities in application and communication protocols.

Unlike other kinds of cyberattacks, which are typically launched to establish a long-term foothold and hijack sensitive information, denial of service assaults do not attempt to breach your security perimeter. Rather, they attempt to make your website, servers, or even your entire network unavailable to legitimate users. In some cases, however, DDoS attacks are used as a diversion from other malicious activities, and to infiltrate web applications.

[1] *Monitoring Hacker Forums*, HII Report #5, Imperva

[2] *The SQL Injection Threat Study*, Ponemon Institute

A successful DDoS attack is a highly noticeable event impacting the entire online user base. This makes it a popular weapon of choice for hacktivists, cyber vandals, extortionists, and anyone else looking to make a point or champion a cause.

DDoS assaults often last for days, weeks and even months at a time, making them extremely destructive to any online organization. They can cause loss of revenue, erode consumer trust, force businesses to spend fortunes in compensation and cause an organization to suffer long-term reputation damage.

Online Fraud Attacks

In addition, hackers have turned their sights to unsuspecting websites, using millions of stolen user credentials to gain acces into sensitive accounts. Traditional malware steals user credentials and hijacks sessions by tracking keystrokes and manipulating website content. While malware targets end users, the true victims are the website owners, often banks and ecommerce sites, which must pay fraud restitution costs due to compromised customer accounts.

Need for Web Application Firewalls

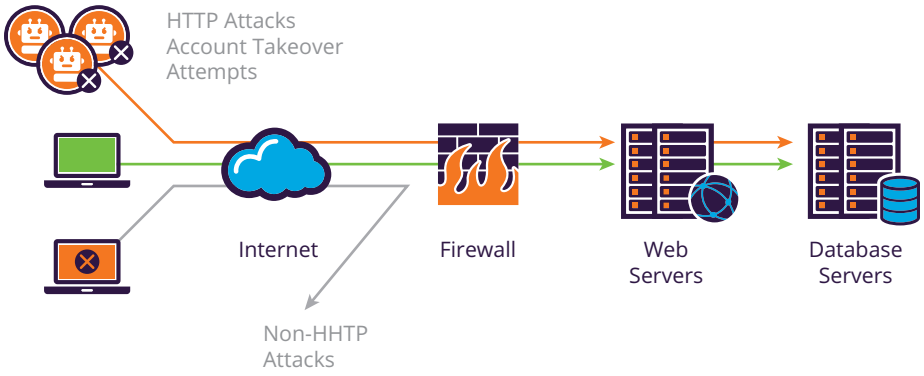
Breaches stemming from cyber-attacks can result in brand damage, customer churn, lost revenue, fines, and lawsuits. As a result, many targets of web application attacks have invested in customer notification and credit card monitoring services for their customers. Nevertheless, in several instances, large-scale breaches have driven companies out of business.

Why Traditional Network Security Fails

When businesses first connected to the Internet in the early 1990s, they encountered the precursor to modern day hackers: malicious users who probed computers for open ports and platform vulnerabilities. To prevent breaches, organizations deployed firewalls and intrusion prevention systems (IPS). However, when organizations allowed access to their web applications, hackers quickly circumvented the firewalls and IPS, using evasion techniques such as encoding and comments to elude IPS signature detection.

Next generation firewalls arrived a few years later and offered more capabilities–they could identify the type of application traffic such as HTTP or instant messaging. Although this application awareness provided access control it had zero benefit in terms of stopping web attacks. Next generation firewalls can block access, but cannot block attacks such as SQL injections, cross-site scripting, or other attacks that exploit web application vulnerabilities. They cannot detect cookie, session, or parameter tampering attacks. They cannot stop fraudulent devices or business logic attacks.

As a consequence, organizations that solely rely on network security solutions remain vulnerable to web application breaches.



Why Web Application Firewalls Are Needed

A web application firewall (WAF) is a hardware, software, virtual and cloud-based solution specifically designed to protect against web application attacks. A WAF can be deployed on a network, an application, in the cloud, or as a hybrid implementation. And it can run as a physical or virtual appliance, a server plugin, or a filter.

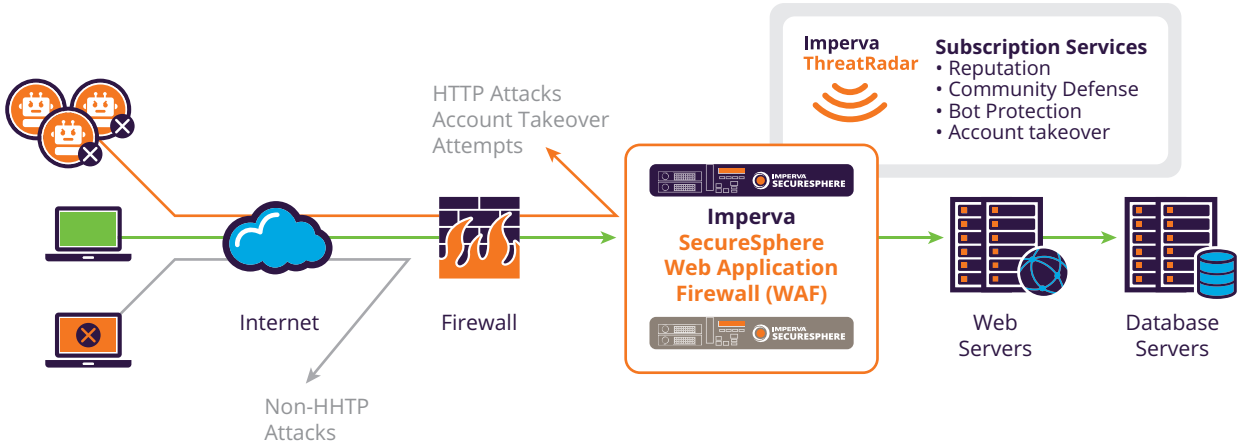
Regardless of how a WAF is deployed or run, a WAF protects web applications by completing the following sequential tasks:

1. Intercepting the network packets of all HTTP and HTTPS requests/responses between a client and web application.
2. Inspecting the network packet's data content–headers, session details, cookies, parameters, file uploads, formats, protocols, and so on.
3. Scanning the data content, using a rule set, to determine whether the request is from a legitimate or illegitimate client and/or contains benign or malicious content.
4. Responding to the request, based on the scan results, by allowing the client to access the web application, requiring the user to input a correct CAPTCHA before allowing access, or blocking the request, session, IP address, user, or file uploads/downloads.

Depending on the rule set configuration, a WAF scan can:

- Validate GET and POST inputs, and then flag any technical web or business logic attacks such as SQL injection, cross-site scripting, directory traversals, or wildcards
- Authenticate XML, JSON, REST, SOAP, and other protocol schemas, and then flag any malicious content or attachments
- Detect and flag the tampering of session, parameter, or cookie information
- Check for and flag known malware signatures or embedded URLs of known malicious websites
- Identify and flag known viruses, spams, or malicious executables or objects
- Discover and flag threats embedded within SSL-encrypted traffic
- Detect and flag automated clients or bots to protect against application-layer denial of service (DoS) and distributed denial of service (DDoS) attacks
- Monitor for and flag requests for sensitive data exfiltration

In short, a WAF will help prevent technical web, business logic, DDoS, and online fraud attacks. It can also help secure data from theft and manipulation. In doing so, a WAF deployment complies with several HIPAA and PCI DSS requirements.



Making the Right Choice

Web application firewalls are characterized by a large number of features and functions. But choosing a WAF doesn't need to be an exercise in guesswork.

The following WAF requirements are recommended to 1) ease the evaluation process, 2) help IT security teams make the best choice within the shortest timespan, and 3) ensure the WAF provides the following web application defenses:

Ten Requirements

1 Understand Web Applications

Challenge: Organizations face the growing specter of advanced, custom web attacks. The richness of JavaScript and SQL allows hackers to devise a virtually unlimited number of SQL injection and XSS attacks. Signatures can help detect web attacks, but they must either be written broadly to catch any potential threat–resulting in false positives–or they must define the exact syntax of the attack–resulting in false negatives. In addition, hackers can use encoding, comments, and obfuscation to evade signature detection. Using evasion methods and a little creativity, hackers can easily outwit traditional security solutions to compromise web applications.

Requirement: To accurately stop attacks, a web application firewall must understand the protected application, including URLs, parameters, and cookies. Understanding the protected application, and then validating input, helps stop attacks like SQL injection, parameter tampering, and cookie poisoning. To validate input, a web application firewall must inspect parameter values for special characters, like apostrophes and brackets, and know whether these characters are expected or indicative of an attack. Dynamic profiling of applications, or the ability to automatically build a baseline of acceptable user behaviors (i.e., requests and responses) is also important in this regard–not to mention critical to minimizing false positives. Since organizations frequently update applications, a web application firewall also needs to automatically learn application changes–without any manual intervention. Automated application learning makes managing a web application firewall manageable, while still providing the highest level of protection available.

2 Stay Ahead of Hackers

Challenge: Hackers constantly innovate. Whether it is because they are creating new attack tools, developing new ways to recruit volunteers, or honing existing techniques, application threats are always evolving. Nowhere is this more evident than on underground forums, where hackers continually unveil new attack vectors and vulnerabilities. Fraudsters are not standing still either; fraud malware developers have architected self-mutating files that can evade virus signature detection. Cybercriminals today can circumvent physical token, smartcard, and out- of-band authentication to perform fraud. Keeping up with the latest application threats—including vulnerability exploits, malicious users, and fraud schemes—is perhaps the most difficult hurdle for application security solutions.

Requirement: A web application firewall must have up-to-date protection to defeat the latest web-borne threats. It should leverage live attack, reputation, and fraud data from around the world to identify both attacks and attackers. Security signatures, policies, reputation data, and fraud intelligence should be updated automatically without human intervention. Besides the frequency of security updates, it is important to look at the research organization producing security content. Is the research organization focused on web application security? Is it equipped to defeat the latest application attacks? If the answer to either of these questions is no, then businesses ought to move on and investigate alternative solutions.

3 Thwart Evasion Techniques

Challenge: Organizations need to block web attacks without blocking legitimate traffic. This may sound obvious, but the solution, unfortunately, is not. A web application firewall must validate input (see requirement #1), but it shouldn't block requests with accidental typos or other legitimate input errors. If a hacker enters unusual characters such as brackets and apostrophes—characters often used in web attacks—into a zip code field in an online form, a web application firewall should detect the unusual behavior.

But what if a valid user inadvertently types five digits and a quote? How do you differentiate between a cybercriminal that executed an attack and a web user that accidentally submitted special characters in a form field? How do you construct attack signatures that detect SQL keywords, like "select" and "union," and "join," but that still allow legitimate requests with these same words? The answer: advanced analytics and correlation.

Requirement: A web application firewall must include an analytics engine that can examine multiple attack indicators to block attacks without incurring false positives, or identifying legitimate visitors as attackers. This analytics engine must be able to evaluate factors such as attack keywords, special characters, protocol violations, and known attack strings simultaneously. It should identify violations and then perform additional analysis using risk scoring and regular expressions to differentiate between malicious requests and unusual, but harmless traffic. A web application firewall must also correlate requests over time to detect repetitive attacks, such as brute force login or distributed denial of service (DDoS). Only a flexible and intelligent correlation engine will enable a web application firewall to stop sophisticated hackers without blocking legitimate users.

4 Prevent Automated Attacks and Bots

Challenge: Cybercriminals have become industrialized, using automation to improve efficiency and scale—and, in turn, transforming hacking into a multi-billion dollar industry. Armed with web scanners and bots, cybercriminals today can quickly discover vulnerable sites. They can leverage off-the-shelf toolkits, like the Havij SQL injection tool, to extract sensitive data. On top of the threat posed by industrialized hackers, organizations today are inundated with other automated attacks—like competitors scraping web content, comment spammers injecting ads into online forums, and disgruntled individuals or groups launching site-crippling DDoS attacks.

Requirement: A web application firewall must be able to stop automated attacks such as site scraping, comment spam, application DDoS, and vulnerability scans. Due to the explosion in automated attacks, stopping malicious users can be as important as stopping malicious requests. But correctly identifying the bad guys requires multiple defenses. First, a web application firewall should be empowered with real-time reputation intelligence that identifies known attack sources, bots, phishing URLs, and anonymizing services, and then allows it to block malicious traffic before an attack can even be attempted. Secondly, a web application firewall should be able to recognize bots—the automated clients that are responsible for the lion's share of automated attacks. This can be achieved by transparently testing web users' browsers to determine if they are standard browsers or simple bots or scripts.

Other giveaways—indicative of application DDoS attacks in particular—include repeated downloads of large files or requests that generate long response times. To mitigate network-level DDoS attacks designed to saturate your Internet connection and prevent legitimate traffic from ever reaching your site, a web application firewall should also include integral support for a high-capacity, cloud-based DDoS protection service.

5 Recognize Malicious Sources

Challenge: Not all web visitors are good. Some actively try to uncover vulnerabilities, steal data, commit fraud, or take down websites. Many of these malicious visitors aren't human at all—they are bots that continuously attack one site after another. Human hackers are sneakier and more sophisticated than bots; they use anonymous proxies or Tor networks to cloak their identity. And fraudsters have their own unique attack vectors, such as stealing user credentials through phishing sites. The problem organizations face today is that they cannot identify malicious users or illicit sites until the damage is done—a bot has requested too many web pages, a hacker has conducted reconnaissance, or a phishing attack has succeeded.

Requirement: A web application firewall must recognize known malicious sources and sites. It should identify users that are actively attacking other websites and stop them instantly, before they can inflict more damage. Because hackers often use anonymizing services, a web application firewall should detect access from anonymous proxies and Tor networks. To combat phishing, it should recognize users referred from a phishing site. Geographic location provides additional context about web users and a web application firewall should be able to restrict access by location both to eliminate unwanted traffic and to thwart DDoS attacks originating from a specific country. Since web application firewalls are the most effective solutions for detecting web-based threats, web application firewalls ought to collect information about attacks and attack sources and share among one another. A cloud-based community defense should deliver accurate, real-time information about hackers, bots, and fraudsters. Such intelligence-based solutions have become a must for application security.

6 Virtually Patch Vulnerabilities

Challenge: Despite the best efforts of application developers and IT security teams, most applications have vulnerabilities. In fact, according to one report, more than three quarters of scanned sites were found to have at least one vulnerability.[3] In addition, 1 in 8 were found to have a "critical" vulnerability—one that would make it trivial for a hacker to access sensitive data or alter the site's content. Worst of all, the average length of time to fix discovered vulnerabilities is 38 days or more, leaving applications exposed to attack for long periods.

Besides the cost and the time required to fix vulnerabilities, organizations must consider additional hurdles such as vulnerabilities in legacy applications—which may have been untouched for years—and in packaged applications—which may entail obtaining and implementing patches from application vendors (if available).

Requirement: A web application firewall must prevent attempts to exploit application vulnerabilities. Defenses such as input validation, HTTP protocol validation, and attack signatures must be able to block most vulnerability exploits out-of-the-box. However, organizations need granular control to ensure strict security measures are applied to known application vulnerabilities. To achieve this capability, a web application firewall can integrate with application scanners and build custom policies to virtually patch vulnerabilities discovered by the scanners.

7 Stop Malware

Challenge: Fraud malware has become enemy #1 for financial institutions. Cybercriminals are now leveraging their success with online banks to branch out into other applications like ecommerce and bill payment. So, how do cybercriminals carry out malware-based fraud? First, they infect machines with malware such as the Zeus or SpyEye Trojans. Then, when infected users log into a targeted web application such as an online banking site, the malware modifies web pages, performs unauthorized transactions, or steals login credentials. While fraud malware targets website users, the ultimate victims are the website owners—typically banks and e-tailers—who are forced to reimburse customers for fraudulent transactions.

[3] *Internet Security Threat Report*, Volume 19, Symantec

Requirement: A web application firewall must be able to mitigate the growing menace of fraud malware. Positioned between web users and applications and with full visibility into layer 7 transactions, web application firewalls can analyze end user attributes and web traffic patterns for the tell-tale signs of malware infection and block malware-infected devices. They can also perform a number of actions, such as monitoring the user for a specified period of time, generating an alert, or integrating with a fraud management solution to open an investigation case. A web application firewall must be able to provide this capability without requiring any changes to the protected web application.

8 Eliminate Payment and Account Origination Fraud

Challenge: Just like fraud malware, online payment fraud costs organizations millions of dollars. E-tailers must contend with expensive chargeback fees, notification costs, and unhappy customers due to a range of Internet fraud schemes. So how can organizations fortify their applications against credential compromise? And how can they roll out these fraud defenses quickly, without requiring expensive and protracted application development projects?

Requirement: A web application firewall must be able to mitigate payment and new account fraud without requiring application changes. A web application firewall must be able to extract and analyze a number of user and transaction attributes, including browser irregularities, known fraudulent devices, and credential compromise. The web application firewall should correlate fraud risk data with web attack, device, and user information to accurately identify and stop fraud.

9 Support On-Premises and Cloud Deployment

Challenge: Application architectures are as diverse and rapidly evolving as application threats. Consider the different configurations and security needs of organizations today. Some businesses host their applications on-premises; others host their applications in the cloud. Some organizations need a transparent security solution with zero impact on applications. Others wish to change application content by rewriting URLs and encrypting cookies. Some companies require a high-performance hardware appliance. Others desire a flexible virtual appliance or cloud-based solution. And large enterprises need it all: a variety of deployment options to support disparate applications hosted on-premises and in the cloud in varied or geographically dispersed locations.

Requirement: A web application firewall must provide flexible deployment and configuration options to satisfy every organization's unique requirements. As many businesses transition their application infrastructure to the cloud, web application firewalls must adapt, supporting private clouds and cloud-based services to protect hosted web applications.

Organizations that host their applications on-premises have at least as many demands as their cloud colleagues. Many of them require a high-performance solution with no changes to existing applications or network devices. Others need a web application firewall that can modify content, sign cookies, and rewrite HTML. And others require non-inline or cloud deployment. During evaluation, organizations should verify that potential solutions can support their on-premises and cloud requirements now and in the future.

10 Automate and Scale Operations

Challenge: Web application and DDoS attacks can be complicated. Stopping those attacks shouldn't be. Security administrators ought to be able to create custom security policies without learning a scripting language. Another challenge for many organizations is that they operate dozens—and even hundreds—of web servers, and these servers may be located in different data centers or even in different countries. Organizations must be able to centrally manage application security policies and monitor events at a global level. Lastly, organizations must be able to investigate security incidents. So, they need detailed security alerts and customizable reports for monitoring and forensics.

Requirement: A web application firewall must deliver point-and-click security policies or automated policy updates. Simple, but flexible policy configuration not only eases initial configuring, but also makes it easier for administrators to review security policies developed by their peers. In addition to custom policies, web application firewalls must also support centralized management so that businesses can synchronize policies and application profiles across all of their web application firewalls, even if those devices are located in separate data centers or on separate continents. Lastly, web application firewalls must provide detailed, actionable security event information. Armed with this data, administrators can understand how hackers are attempting to undermine their applications. Robust monitoring and reporting, along with centralized management and flexible policy configuration, provide organizations with the capabilities they need to successfully manage, monitor, and secure their web applications.

Checklist

The following processes are recommended when evaluating web application firewalls:

- ❑ Identify current location of your web applications (e.g., on-premise, remote data center, cloud-based, other).
- ❑ Assess each web application's risk factor, if it were attacked (e.g., a blog with no comments allowed would be low risk, while a blog that accepts donations via credit card payment would be high risk). Questions to consider are:
  - ❑ Does the web application provide access to sensitive or proprietary data?
  - ❑ Is the web application providing any network or system protection?
  - ❑ What would be the impact to the organization if the web application was compromised or became unavailable?
- ❑ Determine the WAF deployment type (e.g., network, application, or cloud), based on the gathered location and risk factor information.
- ❑ Determine the WAF form (e.g., appliance, virtual appliance, server plug-in or filter), based on gathered location and risk factor information.
- ❑ Evaluate the WAF features and functions, using the information detailed in the Making the Right Choice section of this document.

Conclusion

Web applications drive businesses more today than at any other time in history. Yet, these applications are threatened by a variety of attackers. Hackers are using automated tools to steal data, while hacktivists are compromising and disabling hundreds of well-known websites. And fraudsters are ratcheting up their schemes to perpetrate online fraud while evading detection.

Unfortunately, network security products such as firewalls and intrusion prevention systems are unable to stop these growing risks. To adequately protect these business-critical resources, organizations need to supplement their security solutions with a web application firewall.

However, not just any web application firewall will do. Establishing strong, thorough protection depends on selecting a web application firewall that fully meets and supports the requirements and essential capabilities identified in this guide.

About Imperva

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