Post by **Yash Roongta**

Hi Uvaraj,

This was a very fascinating post. You made a great point about how standard techniques of network filtering by firewalls do not adequately protect against the large surface assault that surrounds web applications.

When used as a reverse proxy, the WAF prevents clients from communicating directly with the backend system. Rather than that, they communicate only with the WAF. Typically, customers are unaware that this is taking place; to them, the process seems transparent.

The two fundamental WAF implementation models are negative security, in which the WAF is configured to prevent known vulnerabilities, attack signatures, and threat actors from accessing the web application based on pre-defined signatures, or positive security, in which the WAF is configured to allow only pre-approved traffic that meets specifically configured criteria. One of the most prevalent WAF configurations is a hybrid model, which incorporates both positive and negative security principles.

Probably the only two disadvantages of a WAF are that traffic latency will be decided by the WAF's performance and that WAFs may also be difficult to implement, where their effectiveness is dependent on the quality of the policies.

References

- Indusface (2019). What is WAF and How Does WAF Works? | Indusface Blog. Indusface. Available from: https://www.indusface.com/blog/how-web-application-firewall-works/ [Accessed 17 Apr. 2022].

- Reblaze (2022). How Does a WAF Work? - Reblaze. Reblaze. Available from: https://www.reblaze.com/wiki/waf/how-does-a-waf-work/ [Accessed 16 Apr. 2022].
- Avi Networks (n.d.). Web Application Firewall 101 - Learn All About WAFs. Avi Networks. Available from: https://avinetworks.com/what-is-a-web-application-firewall/ [Accessed 17 Apr. 2022].

Post by **Amit Pahuja**

Hello Uvaraj,

I really enjoyed the flow of your writing - the examples of frequent attacks and the product/solution you provided to counter them. To that, I'd want to add-

There are solutions in the Next Generation Firewalls (NGFW) which supports prevention of Distributed Denial of Service (DDoS) attacks using Zone protection. Zone protection protects against flood attacks, non-IP protocols attacks, and reconnaissance attempts. Zone protection combined with Flood protection configuration protects an entire ingress zone against TCP SYN, UDP, ICMP and other IP flood attacks. There are thresholds that can be configured, once the threshold is crossed the firewall blocks new connections. Most of the times, the default threshold is set to high so that the legitimate traffic is not dropped.

Connection traffic has to be monitored and thresholds be adjusted to the values appropriate to one's network.

Many of the Web Application Firewalls (WAF) safeguards the web applications by using Attack Signatures (traffic pattern which differentiates good packets and attack packets), AI driven pattern analysis and Layer-7 DDoS protection.

REFERENCES:

1. Cloudflare [N.D.]: What is a DDoS attack? **https://www.cloudflare.com/en-in/learning/ddos/what-is-a-ddos-attack/**
2. What Is WAF [N.D.]: **https://www.imperva.com/learn/application-security/what-is-web-application-firewall-waf/**

Post by **Gokul Kurunthasalam**

Hi Uvaraj,

I appreciate your post on WAF with an example. IIS and SQL Server are critical components of the Microsoft ecosystem, with widespread deployment in a variety of production systems and services. Microsoft Jet Database Engine, which includes MS Jet and ACE, is nearly two decades old, and a large number of Jet modules have been discovered to be easily susceptible due to insufficient exploit mitigations. Remote database access connects Jet vulnerabilities to IIS and SQL Server components, lowering their security to the level of the Jet Database Engine. Attackers might theoretically use this feature to compromise IIS and SQL Server and remotely obtain SYSTEM privilege via a single SQL injection.

NGFWs (Next Generation Firewalls) can assist in preventing attackers from compromising IIS and SQL Server using Jet vulnerabilities. By utilising App-ID and the

Threat Prevention security subscription, Next-Generation Firewalls can assist mitigate such assaults.

Application ID (App-ID), a patented Palo Alto Firewalls traffic classification system available only on Palo Alto Networks firewalls, identifies the kind of application regardless of the port, protocol, encryption (SSH or SSL), or any other evasive technique used by the application. It uses a combination of classification mechanisms—application signatures, application protocol decoding, and heuristics—to reliably identify applications in the network traffic stream. After identifying the programme, the policy check determines how to treat it, such as blocking or allowing and scanning for threats. App-ID provides visibility into the network's applications, allowing you to discover how they work and have a better understanding of their behavioural features and relative risk.

References:

"https://docs.paloaltonetworks.com/pan-os/9-0/pan-os-admin/app-id" by Palo Alto Networks; Published on Mar 14 15:47:45 PDT 2022.

"Next-generation firewall case study" - Bachelor’s thesis - Information Technology by Kalle Kokko in 2017.

"A comparative study of Palo Alto Networks and Juniper Networks next-generation firewalls for a small enterprise network" Student thesis - Malmgren, Andreas and Persson, Simon; Mälardalen University, School of Innovation, Design and Engineering; Published on 2016.

Post by **Shailender Kudachi**

Hi Uvaraj,

This was an interesting read, i like how you talked about few critical attacks and recommended WAF as a solution.

Because of the age of the functional interfaces in PHP and ASP applications, SQL Injection is a typical occurrence. ASP.NET and J2EE applications are less vulnerable to SQL injections than other web-based applications since their programmatic interfaces are accessible. Security-in-depth countermeasures such as low privilege connections to the database server can help to mitigate SQL Injection attacks, albeit to a lower extent than before. SQL Injections are often regarded as having a high severity and a high impact.

Cross-site scripting issues allow an attacker to impersonate a victim user and do any actions that the victim is capable of performing, as well as access any data that the victim possesses. As long as the victim has privileges to the application, the attacker may be able to gain complete control over its functionality and data. The sheer concept of vulnerability triggers red signals in our minds. As a result, this is a High Priority attack for me.

I like your suggestions for WAF. Any enterprise network organization, in my opinion, should have a WAF in place. When used in conjunction with an application layer firewall, it can protect web applications from application layer vulnerabilities including cross-site scripting (XSS), SQL injection, and cookie poisoning.

References :

https://portswigger.net/web-security/cross-site-scripting

https://owasp.org/www-community/attacks/SQL_Injection