Reblaze

Five Questions

you should ask your web security provider

The Internet threat landscape continues to grow both in quantity and sophistication of attacks. Now more than ever, executives must ask themselves: Does our current web security solution provide adequate protection against today's advanced and aggressive threats? Is it going to scale to meet our future needs? Is it going to adapt to new threats as they emerge?

In this white paper, we discuss five questions that executives should raise with their current security providers to ensure that they are not exposed (either currently or in the future) to potentially significant vulnerabilities.

Note: these questions are most relevant if your organization is already using a cloud platform for some or all of its web security (i.e., for scrubbing HTTP/S traffic). If instead your web security solution is appliance based, then you have additional vulnerabilities besides these to worry about.

For example, appliances cannot mitigate DDoS attacks above a certain size. On-premise appliances can only process request packets after they have traversed the incoming Internet pipe. A large-enough volume of traffic can saturate the pipe before appliances have an opportunity to scrub it. This can force the upstream ISP to defend itself by blackholing all traffic for the targeted network, which means the victim's web applications become unavailable to customers and users. (In other words, the attack succeeds.)

Other appliance issues include keeping them up-to-date (when new web threats emerge, updates must be issued and installed, which often does not happen immediately), scalability, and others.

Now that those issues have been mentioned, it's time for the questions.

#1. How do you isolate my resources?

The tenancy model for a web security solution has a big impact on how effectively it can isolate each customer's web assets. Most cloud-based web security solutions are deployed on shared compute and storage resources. These multi-tenant arrangements create several vulnerabilities.

First, there have been instances where cloud customers have been breached through the administrative tools shared with other tenants. (Granted, these incidents are rare, but why should any organization willingly accept a security posture that can be compromised?)

Second, there was a notorious (and months-long) incident where private customer data was exposed to the public Internet because of software bugs in the solution provider's multi-tenant platform. Countless customers had their data leaked merely because other customers had HTML errors in their web sites. Again, such problems would not have occurred in a single-tenant environment.

Lastly, and most commonly, in a multi-tenant environment your organization can be affected by attacks aimed at others. For example, if one customer gets hit with a DDoS attack, then all other customers sharing its cloud resources are also, in effect, under attack.

To avoid all this, your organization should have its web security platform deployed in a single-tenant environment: a VPC (Virtual Private Cloud), unique and dedicated to your exclusive use.

#2. How deep is your analysis?

To be effective, a web security solution must be able to perform thorough analysis on incoming requests and their payloads. This includes:

1. Header and parameter checking
2. HTTP GET support
3. Cookie signing
4. HTTP POST support
5. JSON payload support

Web security solutions differ in the depth of their analysis. Many solutions only offer the first two types of analysis. A few others also include cookie signing and HTTP POST support.

As for JSON payload support, it has become increasingly important for detecting certain types of modern web attacks. Despite this, very few security solutions offer it today.

If your security solution does not offer complete and deep request analysis, then it cannot deliver comprehensive protection.

#3. How effective is your Bot Detection?

Traditionally, hostile bots have been identified with a few different approaches. These include:

- **Signature recognitio**n: identifying patterns in the incoming requests which indicate that the originator is a bot.

- **Rate limitin**g: using rate thresholds to limit traffic from a single source.

- **Blacklisting**: refusing incoming requests from IP addresses known to be hostile.

- **JavaScript (JS) injection and other tests** such as cookie handling, in order to detect the absence of a normal web browser environment.

- **CAPTCHA and reCAPTCHA** challenges.

These approaches still work against older bots, and most web security solutions use them for this purpose.

Unfortunately, these are the only methods that most web security solutions use.

This is bad news, because threat actors have adapted. Newer generations of bots can avoid detection from these traditional methods:

- Signature recognition is defeated by spoofing user agent strings and other deceptive actions, so that the bot appears to be a legitimate human user.

- Rate limiting is evaded by rotating IPs and/or keeping the rate of requests to ‘reasonable’ levels.

- Blacklisting is also avoided by IP rotation.

- Some headless browsers (i.e., web browsers that are run programmatically without a GUI) can now pretend to be “real” web browsers: they can handle cookies, they can execute JavaScript, and so on.

- CAPTCHA and reCAPTCHA challenges can be solved automatically. Even the latest version of reCAPTCHA can be solved with automated methods 97 percent of the time.

To accurately detect the latest generation of bots, a web security solution must use methods such as Machine Learning and UEBA (User and Entity Behavioral Analytics) to construct and enforce biometric behavioral profiles of legitimate human users. However, most solutions do not have these capabilities.

#4: How do you protect my APIs?

Detecting and blocking API abuse is a vital, and increasingly challenging, requirement for robust security. As mobile/native applications have proliferated, so have APIs.

With APIs, threat actors have more opportunities and less risk of detection. Some methods of threat recognition within web applications (e.g., browser environment analysis) do not apply to APIs. Further, each time new mobile/native apps or features go live, an API’s attack surface expands.

For web security that can keep up with the fast pace of modern web and mobile application development (especially in a DevOps/DevSecOps environment), a security solution must provide a variety of features, including a robust client-side SDK, automated schema ingestion and enforcement, reverse-engineering prevention, and deep payload inspection.

If your solution does not provide all this (and few solutions do), it is restricting your ability to fully protect your APIs.

#5. Does your security solution play well with others?

It’s likely that you already have security toolchains in place, including SIEM and SOC solutions. You have a CDN provider that has its own built-in security features. Your DevOps teams have a well-oiled CI/CD pipeline. You use one or more public cloud providers, each with its own stack of security services.

Your web security solution should fit into and complement all of your existing infrastructure and tooling. If it doesn't, it might be time to find a new solution.

Bonus question: Even if your solution provides some (or all) of the above…

…does it provide everything as part of an all-in-one platform?

A few security solutions provide some or even all of the features described above. But most treat them as add-ons, and charge extra for each one.

If your organization is using one of these solutions, perhaps it's time to switch to a platform that provides everything in an all-in-one solution, at a lower price than *à la carte* providers offer.

A solution to evaluate

Reblaze is a comprehensive, single-tenant, all-in-one web security platform that provides the features discussed above, and more. For more information, visit https://www.reblaze.com.

About Reblaze

Reblaze provides comprehensive, cloud-based, robust protection for web applications and APIs. Core technologies include: WAF/IPS, Multilayer DoS/DDoS protection (network, transport, and application), Anti-Scraping, High-level ACL, Advanced Human Detection and Bot Management, Advanced Management Console, and Real-time Traffic Analysis. Added value services include: Layer 7 Load Balancing, Global Secured CDN, and a complete DNS solution.

Reblaze's clouds are fully compliant with SOC 1/SSAE 16/ISAE 3402, FISMA Moderate, PCI DSS, ISO 27001, FIPS 140-2, HIPAA, CSA (Cloud Security Alliance), and other standards and certifications. **Reblaze Technologies is ISO 27001 Certified, AICPA SOC 2 Certified, and is a PCI DSS Certified Level 1 and Level 2 Service Provider**.