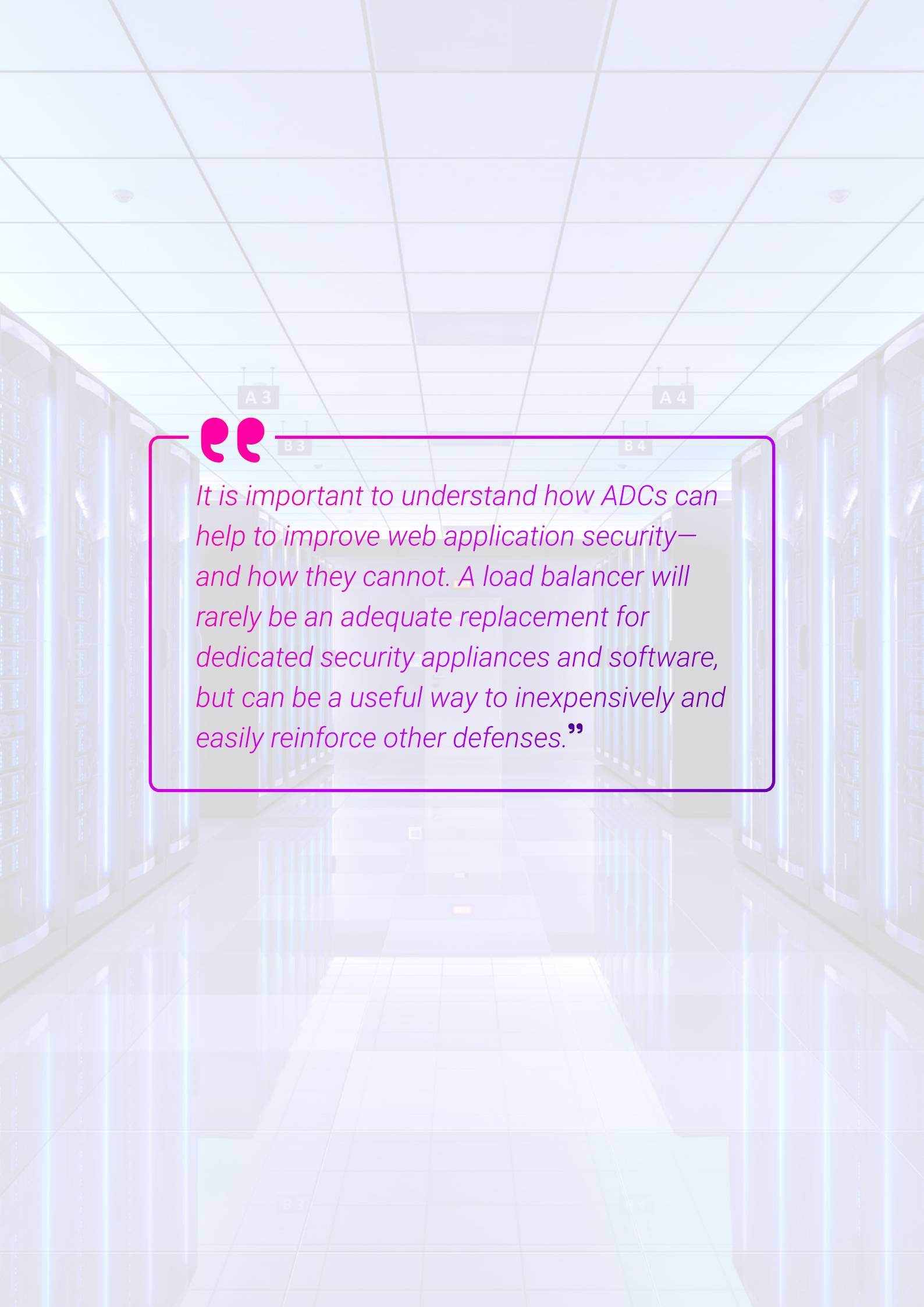
LOADBALANCER

WHITE PAPER

The OWASP Top 10 web application security threats

How Application Delivery Controllers can (and cannot) help to mitigate these vulnerabilities



> *It is important to understand how ADCs can help to improve web application security—and how they cannot. A load balancer will rarely be an adequate replacement for dedicated security appliances and software, but can be a useful way to inexpensively and easily reinforce other defenses.*

Executive overview

When it comes to web application security threats, OWASP is the industry authority. The Open Worldwide Application Security Project (or OWASP for short) is a not-for-profit foundation focused on improving software security for all, comprised of industry experts and members of the free and open source software community — including some of Loadbalancer.org's own staff.

OWASP regularly publishes a globally recognized, data-driven awareness document about the "Top 10" web application security threats, and updates this list every few years as new threats and vulnerabilities emerge. The current list includes a diverse range of issues, from broken access controls and authentication failures, to exposure to injection attacks and unsafe software engineering processes.[1]

ICT Directors, CTOs and other senior managers responsible for web application security should refer to the OWASP Top 10 and consider how they can reduce their exposure to these common vulnerabilities, using a combination of security appliances, technologies, policies, and processes.

Despite what some Application Delivery Controller (ADC) vendors may imply, load balancers are not inherent security devices and were never designed for this purpose, so they should not be used as a first line of defense. After all, there are a whole host of security products, including network firewalls, that exist specifically for that purpose. However, ADCs can be intelligently configured and deployed to support overarching cybersecurity objectives, helping to minimize many of the top threats.

So for those organizations that already use ADCs or have a load balancing requirement, it is important to understand how load balancers can help to improve web application security — and how they cannot.

The OWASP Top 10 web application security vulnerabilities*

1. Broken access control
2. Cryptographic failures
3. SQL injection
4. Insecure design
5. Security misconfiguration
6. Vulnerable and outdated components
7. Identification and authentication failures
8. Software and data integrity failures
9. Insufficient logging and monitoring
10. Server-side request forgery

*March 2023

https://www.owasp.org/www-project-top-ten/

[1]OWASP Top 10:2021 [Internet]. [cited 2023 Mar 30]. Available from: https://owasp.org/Top10/

1. Broken access controls

Broken access controls are right at the top of the OWASP list, indicating that this is the biggest and most prevalent threat to web application security. It has grown in significance over recent years, moving from fifth to first place on the list since 2017.

Access controls exist to ensure that users don't perform tasks or access resources that they are not authorized to perform. For example, depending on their job role, a user should not have access to another user's account or permission to make changes to a web application. When access controls are broken, users can either unintentionally or maliciously access and corrupt servers, administration panels or databases, causing downtime in critical business applications or data loss.

> *Load balancers can be used to provide or reinforce certain types of access control."*

It is important to note here that any modern web application or API ought to feature some form of request rate limiting as a security measure. This helps to thwart automated Denial of Service (DoS) attacks, brute force attacks, and credential stuffing. However, where rate limiting controls are inadequate, too coarse, or completely missing, this security logic can be added at the ADC layer.

In this scenario, ADCs can be used to provide or reinforce certain types of access control. For example, many web applications use cross-origin resource sharing (CORS) to control or allow access from subdomains and trusted third parties. If CORS is not available as part of the web application (or is implemented inconsistently, inaccurately, or incompletely), CORS functionality can be added or fortified by a load balancer sitting in front of the application.

It is worth highlighting that access controls on the load balancers must also be secured, otherwise the load balancers themselves could give rise to vulnerabilities. It is therefore best practice for load balancer vendors to ship their products as 'secure by default.' This means that although access rights for the load balancer can be changed and granted as required by the administrator, they are initially fully locked down to minimize the risk of unauthorized access.

2. Cryptographic failures

In 2023, no application should be communicating over the public Internet in plain text. Data needs to be encrypted, transforming it into formats that cannot be deciphered easily or accessed by unauthorized users. In particular, passwords, credit card numbers, health records, and personal information should unquestionably be encrypted to make them secure when they are being transmitted electronically. Furthermore, some form of encryption is often required to meet national or international laws or industry regulations, such as the EU's General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS).

Best practice today is to use the latest version of Transport Layer Security (TLS), the encryption protocol that replaces the previous and more commonly known Secure Socket Layer (SSL) protocol. Old, legacy applications may not incorporate TLS or SSL cryptography however, or they may make use of older cryptographic standards that are no longer considered secure. Cryptographic failures can occur when the algorithms or protocols used become out-of-date, or vulnerabilities are discovered in their implementation, making them easier to break.

For applications that either cannot perform encryption or use outdated encryption methods, ADCs can be used to encrypt outgoing data and decipher incoming data using the latest standards and protocols, effectively taking over responsibility for cryptography from back-end servers.

Some organizations elect to off-load TLS/SSL encryption to load balancers, even when their web application servers could securely handle this function. Indeed some ADC vendors actively promote this approach to increase the computational load on their load balancers and therefore sell ever more powerful (and more expensive) load balancers with bigger licenses. In practice, it can be more cost effective and scalable for organizations to continue to handle TLS/SSL encryption on their back-end servers, if their applications have this capability. TLS/SSL off-loading should, therefore, only be considered if the web application cannot perform cryptography securely itself, or if there is another compelling business or technical reason to do so.

3. Injection

Injection refers to the process by which malicious code is 'injected' into an application, making it do something unintended. It often takes the form of an SQL injection — the insertion of an instruction in the SQL (Structured Query Language) programming language asking a database to release data without the correct authorization.

This category of threat in the OWASP list also includes cross-site scripting (abbreviated as XSS), through which malicious code is inserted secretly into trusted websites, rendering them unsafe. Cross-site scripting attacks can inadvertently be spread by users who click on unsafe links.

> *WAFs have specific functions for detecting SQL and XSS injections attacks"*

A web application firewall (WAF) (a core component in many ADCs) can be used to inspect the content of traffic before it reaches web applications, therefore helping to prevent injection attempts. WAFs generally have specific functions for detecting SQL and XSS injection attacks.

The free and open source ModSecurity WAF engine is used as the basis for the WAF that is present in most ADCs, including many proprietary ones. When suspicious web requests and responses are spotted, ModSecurity will automatically create an alert message and (depending on how the WAF is configured) block the injection attempt at the load balancer so that it never reaches the web application.

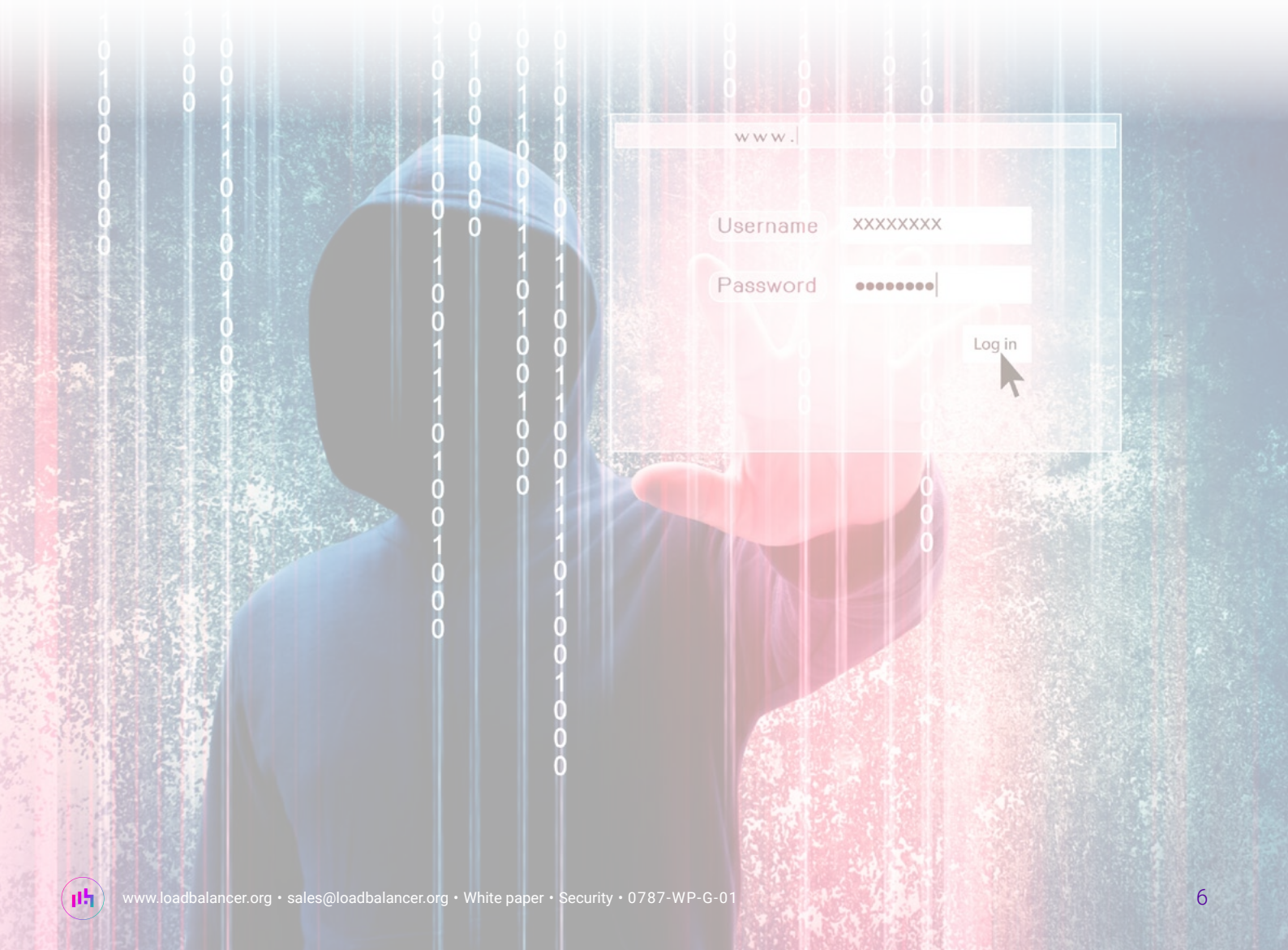
4. Insecure design

A broadly-defined category, insecure design refers to flaws in the solution or reference architecture, such as the absence of necessary security controls. This is a new category in OWASP's Top 10 and has prompted OWASP to call for "more use of threat modeling, secure design patterns and principles and reference architectures."

If an application has not been designed in a secure manner, having a load balancer in front of it can provide an extra layer of flexibility and the ability to remediate some of the issues. Of course it does, however, depend on the issue. An ADC will not provide a quick fix for every potential vulnerability that arises from insecure design, but there are a few instances when a load balancer might help.

For example, if the design of an application means that it can be accessed by all kinds of users in the same way, a load balancer could be used to artificially create different virtual services. Traffic from the public could be directed via a more secure route than traffic from internal staff members. It is possible to segregate different tiers of traffic or tenants to improve security and reduce risks by controlling who can access different parts of the application.

Another recognized design issue is when applications have been built without limits to the number of characters or the type of data that can be entered into form fields. Attackers can exploit this vulnerability by putting programs into username boxes, for example, in the hope that backend systems will read and initiate this code. If this design flaw exists, ADCs can be configured to proactively check user-inputted data to make sure that a submitted username, for example, is between 8 and 32 characters long and only contains letters and numbers. In this way, load balancers will preemptively validate all input from users and stop anything that violates policy before it reaches the backend web application servers.



5. Security misconfiguration

Generally speaking, security misconfigurations result from human error. Perhaps users make mistakes; perhaps they have insufficient training. Either way, if they don't complete configurations properly, if they accept default settings that are insecure, put sensitive information in error messages, or make any number of other oversights, they can inadvertently cause a significant data breach or web application failure.

It is worth noting that this category of threat is growing in concern. It has moved from position 6 up to position 5 since the previous OWASP list was published.

> *Load balancers can be set up to provide a layer of defense against brute force attacks."*

A common failing is leaving administrator accounts, unnecessary ports, and web pages open to infiltration through brute force — when someone repeatedly bombards a login field with hundreds, thousands, or millions of possible (commonly used) passwords in the hope of breaking in. ADCs can be set up to provide a layer of defense against brute force attacks. They can count the number of times a single client submits a password, and (when certain thresholds are met, such as 20 password submissions in 20 seconds) the client is then blocked.

Similarly, load balancers can help defend against distributed denial of service (DDoS) attacks. Like brute force attempts to penetrate servers, DDoS attacks take the form of millions of superfluous requests, although these are typically launched from multiple distributed clients. Core functionality in ADCs, such as rate limiting, can be used to block such attacks before they reach the web application and, indeed, before they even reach and disrupt the WAF functionality in the load balancer.

Finally, ADCs also have the ability to scan response traffic and detect the leakage of sensitive data as well as file suffixes that are not typically delivered via the web app. WAF engines can detect suspect traffic and issue alerts if it appears that a malicious actor is trying to exfiltrate data that isn't generally public-facing.

6. Vulnerable and outdated components

Web applications today often consist of multiple components, and if any one of those components has gone End-of-Life (EOL), this can create a vulnerability. That component could be a piece of software, a library, or a framework. Companies are therefore likely to be vulnerable if they do not monitor their software versions, upgrade when upgrades become available, and install patches straight away.

Organizations can find themselves in a difficult situation if no official patch is yet available for a newly identified vulnerability; if a legacy system is no longer receiving patches; if they don't have the legal or contractual right to patch or alter a software solution; if they cannot apply a patch until a scheduled maintenance window; or if the cost of resolving a security issue in a custom-developed web application is prohibitively expensive. In all these situations, organizations can use the WAF functionality in load balancers to do what is known as 'virtual patching'.

> *Organizations can use the WAF functionality in ADCs to do what is known as virtual patching."*

Using a WAF, it is possible to write specific rules to virtually patch and preemptively fix known issues before a full patch can be applied. These rules detect and block exploit attempts before they reach a web application. To do this, organizations need full details about what an attack would look like, although generally there is plenty of information available from industry experts, security researchers, and indeed sometimes from malicious actors themselves. Virtual patches are created individually for each identified vulnerability. Consequently, organizations will want to have a load balancer provider with a strong technical support team and the willingness to help customers address specific security concerns when they inevitably arise.

Of course, it is also important to ensure that ADCs themselves are kept up-to-date and patched as necessary. Organizations that use load balancers based on free and open source technology have more transparency about potential risks. Any issues that arise are quickly identified, publicized, and patched by the open source community — with thousands of eyes on the code and freely available solutions. ADC vendors with strong ties to the open source community are often able to prepare patches in advance of official security announcements so they can get fixes into the hands of their customers promptly. In contrast, organizations that use ADCs based on proprietary software have only hope and faith that their vendor will be honest about any vulnerabilities that arise, will notify them in a timely manner, and will invest in developing any and all security patches that are required.

7. Identification and authentication failures

Attacks can be made against the systems used for identifying and authenticating users. In such instances, passwords, security keys, and session tokens can become compromised, enabling hackers to assume the identities and permissions of their victims and criminally access, and corrupt and disable web applications. OWASP best practices to prevent such attacks include using multi-factor authentication, preventing the use of weak passwords, strengthening 'forgot password' processes, and monitoring failed login attempts.

When data breaches occur and large quantities of usernames and passwords are stolen, these identities generally end up 'for sale' on the dark web. Malicious actors can acquire these lists and speculatively use them on other websites to try to exploit people who have used the same password across multiple applications. For such attacks to succeed, the malicious actors need to be able to present hundreds or thousands of combinations of usernames and passwords in rapid succession — and this is precisely the kind of activity that load balancers can detect and block.

Using a variety of rate limiting techniques, load balancers can block large numbers of requests from the same IP address within a specific timeframe. If a given IP address makes more than 50 requests per minute, for example, this would be identified as suspicious and blocked. The ADC could also be set to slow down suspicious-looking requests, so if an IP address makes 20 requests in the first minute, it may then be restricted to 10 requests in the next minute and 5 requests in the minute after that.

Load balancers can also detect and prevent overuse of the same password. Consequently, an alert could be raised if an ADC identified multiple attempts to log into a succession of accounts with different usernames and the same weak password, such as 'password123'. There are many examples of ways to strengthen login and authentication mechanisms, and (if an organization has a specific problem) it is often possible for certain load balancing vendors to tailor a precise solution.

8. Software and data integrity failures

This is another broad category. In summary, it relates to failures in code and/or infrastructure that arise from assumptions being made about data or application integrity. For example, in continuous integration / continuous development (CI/CD) software development workflows, there may not be appropriate controls in place to ensure the integrity of open source or third party code that is being used in the build process. Equally, auto-update functions, which allow software updates to be downloaded without proper verification taking place, can create vulnerabilities that attackers can exploit to distribute their own malicious code.

Without doubt, organizations need to ensure the authenticity of any software that they use and carry out proper checks to make sure that code has not been maliciously tampered with. This isn't really something that a load balancer can help with; it's more of a discipline that needs to be rigorously applied throughout the web application development process from the very beginning.

It is, however, a risk that ADC vendors themselves are very aware of. Respected load balancing vendors will conduct the necessary due diligence when updating the software in their load balancing solutions to reduce the risk of incorporating code that may prove to be insecure. Organizations that use load balancers based on open source software can scrutinize the make-up of their load balancers and audit the legitimacy of the underlying code for themselves if they wish.

9. Insufficient logging and monitoring

It is vitally important to monitor web application traffic and log events so that if attacks do occur they can be detected and stopped quickly. A key part of the problem is that appropriate alerting thresholds and response escalation processes are often not in place or are ineffective, according to OWASP.

ADCs have the capability to generate and store detailed logging and auditing data about load balancer performance and traffic. System overview dashboards, for example, can provide a 24-hour picture of performance, highlighting spikes in activity in CPU, memory, and traffic volumes and providing an indication of unusual activity such as a possible brute force attack. The information provided by load balancers may, however, not always be sufficient to help organizations react quickly to attacks that are already in progress. At-risk organizations that require more detailed monitoring and event logging intelligence can, however, export data from their ADCs to third party event management systems that provide real-time analytics and alerting functionality.

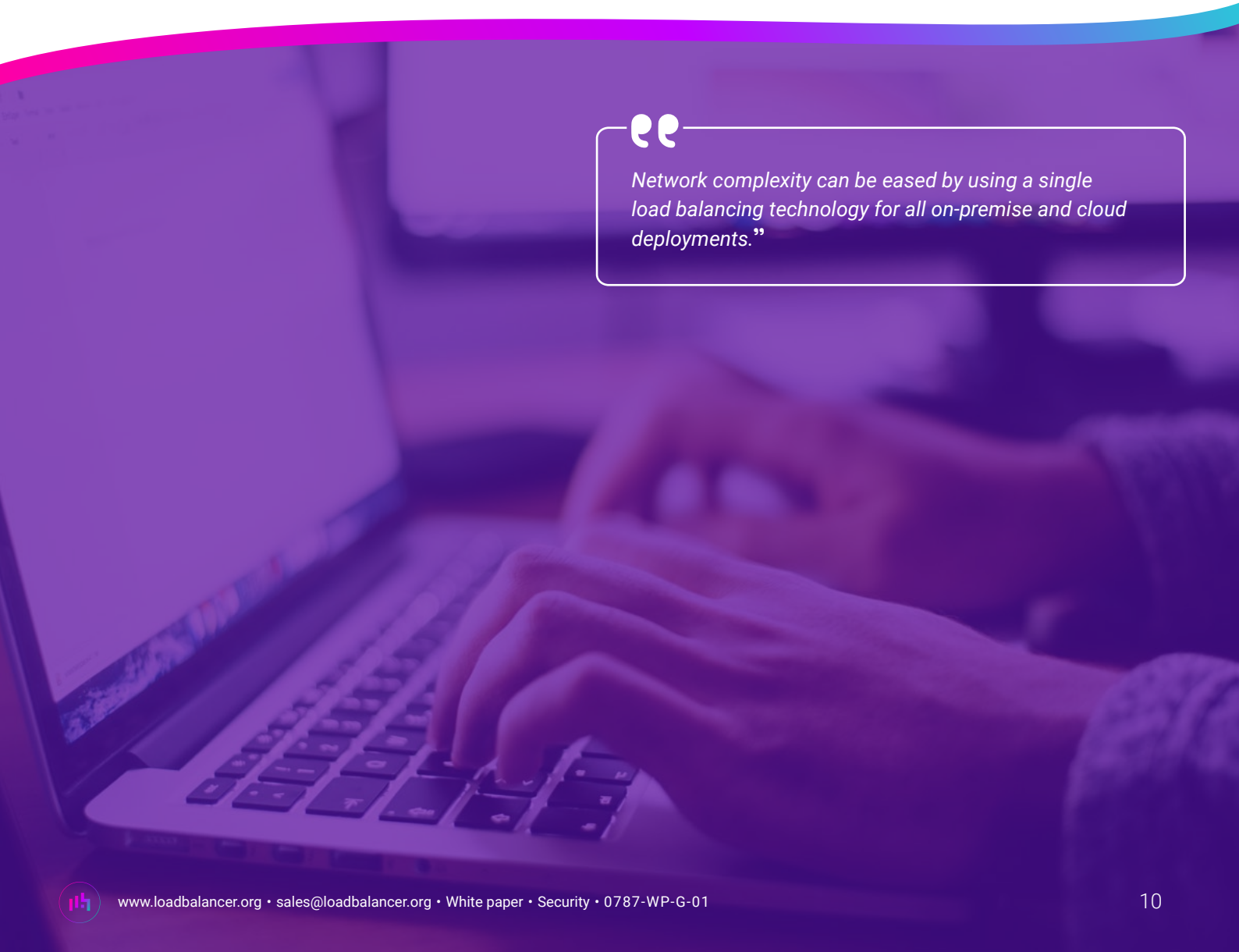
Most WAFs in load balancers feature some form of audit mode that will produce a full log of HTTP transactions, detailing precisely what users do in each session. Audit logs generate a vast amount of data, so (in practice) few organizations will use full audit mode logging in production. This feature can, however, be an invaluable tool for troubleshooting or dissecting an attack, as it enables IT and security teams to see, byte by byte, what data is being transmitted in both directions and helps them to identify any problems — whether these problems are caused by malicious activity, or something else.

10. Server-side request forgery

In at number 10 is server-side request forgery (SSRF), a form of attack in which a malicious actor is able to exploit a trusted server within an organization's infrastructure by tricking that server into sending requests to URLs of the attacker's choosing. According to OWASP, both the incidence and the severity of SSRF attacks is increasing.

Problems arise when the URL for an outbound request from the internal server is built using part of the original request sent to the server from the attacker. If an attacker can deduce that this is happening, then they can craft their requests to the server to make it access any URL they wish. So if the internal server is being blindly trusted, then the malicious requests it makes on behalf of the attacker will be blindly trusted too.

A logical response to SSRF is to take a zero trust approach and apply it to all applications, networks, data, devices, and identities. While purchasing an ADC does not a zero trust architecture make, the position of the load balancer in the network architecture can play a role in preventing SSRF attacks. For example, rather than using a traditional network setup where the load balancer might sit in a Demilitarized Zone (DMZ), with a zero trust setup it's more likely that there are many, smaller load balancers provisioned in segmented zones in order to keep traffic as separated as possible, with each ADC sitting in the same network segment as the servers it needs to load balance. Alternatively, it can be better to work with a single pair of load balancers that are allowed to communicate with each of the different network segments of interest. Then, on the load balancers, access control lists (ACLs) can be used to control precisely which load balanced services can be accessed from specific network segments and clients.



> *Network complexity can be eased by using a single load balancing technology for all on-premise and cloud deployments."*

Conclusion

It is important to remember that web application security risks are continuously evolving — and, in response, so too will the OWASP list.

In the meantime, ADCs can be used intelligently as part of a broader cybersecurity strategy to reduce the risks highlighted in the Top 10.

Load balancers can:

1. Compensate for broken access controls in web applications by providing or reinforcing access controls such as CORS.
2. Perform cryptographic functions when cryptography fails or is missing in web applications by encrypting outgoing data and decipher incoming data.
3. Help detect SQL and XSS injection attacks using web application firewall functionality.
4. Segregate different tiers of traffic or tenants to improve security and reduce risks posed by insecure design.
5. Protect against brute force and distributed denial of service attacks that might otherwise succeed due to security misconfigurations.
6. Create virtual patches to address weaknesses arising from vulnerable and outdated components.
7. Use rate limiting techniques to help prevent malicious actors from exploiting identification and authentication failures.
8. Not really help with software and data integrity issues.
9. Provide data that can be exported to third party event management systems for real-time logging and alerts.
10. Form an integral part of modern zero trust architectures that reduce the risk of SSRF.

The power of the open source community

OWASP is an open source foundation, beloved by the open source community. Its community-led approach, the brilliance of its members, and the free tools and advice it makes freely available to all are part of the reason why the open source community is second to none.

We are proud to be a small part of this amazing, worldwide community and are privileged to be able to support a number of OWASP projects around the world, including the development of the OWASP ModSecurity Core Rule Set (CRS), the de facto free and open source WAF rule set that provides protection against a wide range of web application attacks.

Want to know more?

If you would like more information about how to optimize the use of ADCs to help you address the OWASP Top 10 security challenges, contact: sales@loadbalancer.org.

About Loadbalancer.org

Loadbalancer.org's mission is to ensure that its clients' businesses are never interrupted. The load balancer experts ask the right questions to get to the heart of what matters, bringing a depth of understanding to each deployment. Experience enables Loadbalancer.org engineers to design less complex, unbreakable solutions - and to provide exceptional personalized support.

LOADBALANCER

Visit us: www.loadbalancer.org

Phone us: +44 (0)330 380 1064

Phone us: +1 833 274 2566

Email us: info@loadbalancer.org

Follow us: @loadbalancer.org