AIRLOCK®
SECURE ACCESS HUB

Airlock and the OWASP Top 10 2017
The Ten Most Critical Web Application Security Risks

The following table lists the ten most critical web application security risks, as identified by OWASP in their edition of "OWASP Top 10 2017". It explains how Airlock WAF addresses each of these risks to protect web applications from these types of attacks and which features are relevant.

Vulnerability	Description	How Airlock WAF prevents an exploit	Relevant Airlock Features
A1 – Injection	Injection flaws, such as SQL, OS, and LDAP injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.	Requests containing injections like SQL, XSS, HTML, LDAP, OS commands and others are detected and blocked by combining blacklisting and dynamic whitelisting. Additional protection like URL encryption, smart form protection and Dynamic Value Endorsement prevent any tampering with URL parameters and read-only form field values sent by the application. Attacks in headers or cookies are prevented by filtering and a cookie store. Airlock WAF protects itself against overflow and OS injection attacks by using strict security domain separation, SELinux to implement least privilege, ASLR, No-execute and strong stack protection. The ICAP interface allows checking content either with Airlock WAF add-on modules such as SOAP/XML/AMF-filters or third party malware scanners (AV). Other types of injections and protocol violations are prevented by the protocol termination and regeneration.	– Whitelist parameter Learning – Built-in blacklist filters – URL encryption – Smart Form Protection – Dynamic Value Endorsement (DyVE) – Cookie Store – CAPI interface – HTTP Protocol termination and regeneration – Add-on modules – Security domain separation – Principle of least privilege – Address-Layout-Randomization (ASLR) – No-Execute (NX) – Stack-protection (SSP)

About OWASP
The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organisations to conceive, develop, acquire, operate and maintain applications that can be trusted. All of the OWASP tools, documents, forums and chapters are free and open to anyone interested in improving application security. For more information visit the homepage at www.owasp.org

About OWASP Top 10
OWASP Top 10 is published roughly every 3 years and provides a powerful tool for raising awareness regarding web application security. The 10 issues listed represent a broad consensus on what the most critical web application security topics are at this time. For more information on the OWASP Top 10, visit www.owasp.org

Vulnerability	Description	How Airlock WAF prevents an exploit	Relevant Airlock Features
A2 – Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities.	Airlock IAM is a specialized authentication and authorization server, proven for many years in high security environments. Airlock WAF supports upstream authentication using Airlock IAM. Policies granting access to applications and resources only to authenticated users can be enforced centrally. This includes WebSockets and SSL VPN connections. Airlock IAM supports plenty of authentication means. Using risk-based authentication, second factors for strong authentication are only required if the risk score surpasses a certain threshold. As the HTTP protocol is stateless by nature, sessions are normally bound to a session ID contained in a cookie or in a URL parameter which is passed with each call. Any session ID manipulation is prevented by encrypting all URLs or the session cookie. By default, Airlock WAF replaces all application cookies with its own session tracking (based on the SSL session ID or a secure Airlock WAF session cookie). Using Airlock client fingerprinting, events indicative for session hijacking may be penalized and result in preventive actions (e.g., termination of a suspicious session).	– Upstream authentication with Airlock IAM – Risk-based/adaptive authentication – Cookie Store – Cookie encryption – URL encryption – Secure Session Management – Airlock Client Fingerprinting – SSL VPN
A3 – Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.	Requests containing XSS content are detected through a combination of blacklisting and dynamic whitelisting. URL encryption, smart form protection and Dynamic Value Endorsement prevent any tampering of URL parameters and read-only form field values sent by the application. Security headers such as X-XSS-Protection are added by default. Content-security-policy headers can be added to enforce a stricter source check for various kinds of content.Adding the HttpOnly flag protects the Airlock WAF secure session cookie from being used in JavaScript.	– Built-in blacklist filters – Cookie store – Cookie encryption – URL encryption – Smart form protection – Dynamic Value Endorsement – Airlock WAF secure session handling – Header rewriting

Airlock and the OWASP Top 10 2017

AIRLOCK®
SECURE ACCESS HUB

Vulnerability	Description	How Airlock WAF prevents an exploit	Relevant Airlock Features
A4 – Broken Access Control	Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/ or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.	As Airlock WAF enforces upstream authentication and authorization, no unauthorized external request may reach protected applications. Airlock WAF acts as a central policy enforcement point and checks whether a given user is allowed to access an API or a resource. Object keys and IDs exposed by the application can be protected against tampering using a number of techniques such as URL encryption, smart form protection or Dynamic Value Endorsement (DyVE).	– Upstream authentication and authorization – URL encryption – Smart form protection – Dynamic Value Endorsement (DyVE)
A5 – Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, platform, etc. Secure settings should be defined, implemented, and maintained, as defaults are often insecure. Additionally, software should be kept up to date.	Airlock WAF contains default rules which are regularly updated. The mapping-oriented configuration guides the administrator to configure access only to known applications intended to access. Error messages can be rewritten or replaced to eliminate exposure of stack traces and the like. Typical errors such as excessively permissive CORS headers or missing "secure" attributes in cookies are recognized and corrected. Configuration validators are checking the Airlock configuration and warn about common misconfigurations (Log only mode, certificate mismatches …) The smart policy learning automatically generates meaningful and well-balanced configuration suggestions for easily handling detected issues. This helps guiding administrators along best practices and prevents overreaction in stressful situations.	– Secure default configuration – built-in filters – Policy learning – URL encryption – Mapping-based configuration – Content rewriting – Error page replacement – Header rewriting – Configuration validation

Vulnerabilities	Description	How Airlock WAF prevents an exploit	Relevant Airlock Features
A6 – Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser.	If sensitive data is contained in the URL or in a cookie, it will get additional protection as Airlock WAF can encrypt these. Default rewrite pattern allows removing sensitive data out of responses – such as credit card numbers. Secure SSL/TLS configuration is hard. Ergon actively monitors the SSL/TLS layer technology and provides rapid fixes in code or configuration for newly discovered vulnerabilities. Airlock WAF, acting as reverse proxy between the application and the browser, can encrypt the connection using TLS. If necessary, application responses can be re-written to contain HTTPS URLs only, even if the back-end uses HTTP for performance reasons. The Strict-Transport-Security header (HSTS) is set by default. Public-key-pinning (HPKP) can be configured as a response action. Additionally, Airlock WAF forbids weak SSL/TLS ciphers by default. OCSP stapling simplifies validation of certificates. Password hashes are sensitive information, that's why they don't belong in the normal application database. Upstream Authentication solves this problem by separation.	– URL encryption – Cookie store – Cookie encryption – Response rewriting – SSL/TLS termination – Secure SSL/TLS configuration – Upstream authentication
A7 – Insufficient Attack Protection	The majority of applications and APIs lack the basic ability to detect, prevent, and respond to both manual and automated attacks. Attack protection goes far beyond basic input validation and involves automatically detecting, logging, responding, and even blocking exploit attempts. Application owners also need to be able to deploy patches quickly to protect against attacks.	Airlock WAF (as a web application firewall) is a highly specialized component dealing with detection and prevention of various types of attacks. Detected attacks are visible in the onboard reporting or can be forwarded to an SIEM system for further analyses. Virtual patches against attacks are quickly applicable, centrally and for all protected applications at once.	– Whitelist filter – Whitelist parameter learning – Built-in blacklist filters – Detection of automated scanning – Client fingerprinting – CSRF tokens – Blocking requests by IP Geolocation – URL encryption – Virtual patching – Secure session management – Logging or blocking of attacks – SIEM Integration (CEF log format, Splunk app)

Vulnerabilities	Description	How Airlock WAF prevents an exploit	Relevant Airlock Features
A8 – Cross-Site Request Forgery (CSRF)	A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. Such an attack allows the attacker to force a victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.	CSRF tokens are used for the protection of standard HTML webpages as well as for JavaScript / REST applications. URL encryption with a session based key blocks CSRF attacks too. Such a URL and CSRF tokens are only valid for one user (and only during his or her session). The X-XSS header is set by default. CSP (content-security-policy) headers can easily be switched on using standard response actions. The same-site attribute can be added to cookies by the WAF.	– CSRF tokens – URL encryption – Header rewrites
A9 – Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defenses and enable a range of possible attacks and impacts.	The Airlock security team actively monitors and analyses new threats and vulnerabilities in web applications. If required, security patches for Airlock WAF and IAM are released quickly and customers are notified. Airlock WAF protects itself against 0-day attacks with its fault tolerant architecture. Privilege separation (SELinux) enforces the request data to be handled in the correct chain. The web listener is not allowed to access session management or to craft backend request. Address-layout-randomization (ASLR), no-execute (NX) and stack-protection (SSP) are enabled on the WAF and reduce attack surface.	– Hotfixes – HTTP protocol termination – Security compartments – Address-Layout-Randomization (ASLR) – No-Execute (NX) – Stack-protection (SSP) – SELinux/Least Privilege

Vulnerabilities	Description	How Airlock WAF prevents an exploit	Relevant Airlock Features
A10 – Underprotected APIs	Modern applications often involve rich client applications and APIs, such as JavaScript in the browser and mobile apps, that connect to an API of some kind (SOAP/XML, REST/JSON, RPC, GWT, etc.). These APIs are often unprotected and contain numerous vulnerabilities.	RESTful web services often use JSON for data transfer. Airlock WAF's integrated JSON parser allows the consistent application of security policies both to standard HTML form posts and REST calls. That is, attributes of JSON objects in REST calls are verified against the same filter rules as standard HTTP parameters. Moreover, DyVE (Dynamic Value Endorsement) allows the dynamic endorsement of selected attribute values within a session's scope. Subsequent REST calls must comply by using endorsed values for the selected attributes. Mobile clients typically ignore cookies, which are traditionally used for secure session handling in web applications. In order to protect mobile sessions, Airlock WAF supports session management based on access tokens (e.g., Bearer tokens). Using Airlock IAM, Airlock WAF supports upfront authentication and authorization of REST calls. Airlock WAF's SOAP/XML filters also interpret WSDL and schema files to ensure that a web service API is used in the specified form.	– SOAP/XML filter – Full JSON support in filter rules – Whitelist parameter learning – Dynamic Value Endorsement (DyVE) – Upfront authentication and authorization of REST calls – Secure session management for REST calls

Ergon Informatik AG
Merkurstrasse 43
CH-8032 Zürich

Telefon +41 44 268 89 00
Telefax +41 44 261 27 50
www.ergon.ch

ergon