AIRLOCK

Airlock und die OWASP Top 10 2013
Die 10 grössten Sicherheitsschwachstellen von Webanwendungen

Die folgende Übersicht zeigt auf, wie Airlock Webanwendungen vor Sicherheitsrisiken schützt und welche Funktionalitäten von Airlock dabei zur Anwendung kommen. Die Tabelle folgt den 10 grössten Sicherheitsschwachstellen von Webanwendungen, wie sie von der OWASP in ihrer aktuellen Ausgabe der „OWASP Top 10“ (2013) definiert worden sind.

Schwachstelle	Beschreibung	Wie Airlock schützt	Funktionalitäten von Airlock
A1 – Injection	Injection-Schwachstellen, wie z.B. SQL-, OS- oder LDAP-Injection treten auf, wenn nicht vertrauenswürdige Daten als Teil eines Kommandos oder einer Abfrage von einem Interpreter verarbeitet werden. Ein Angreifer kann Eingabedaten dann so manipulieren, dass er nicht vorgesehene Kommandos ausführen oder unautorisiert auf Daten zugreifen kann.	Anfragen, die SQL-Befehle enthalten, werden durch eine Kombination von Blacklist-Filtern und dynamischen Whitelist-Filtern detektiert und blockiert. URL-Verschlüsselung und Smart Form Protection verhindern die Modifikation von URL-Parametern und versteckten Formular-Feldern. Angriffe über Header-Felder oder Cookies werden durch Filter und/oder den Cookie Store verhindert. Airlock selbst ist gegen Overflow und OS Injection-Attacken durch eine strikte Trennung der Security Domains geschützt. Die ICAP-Schnittstelle ermöglicht Inhaltsfilterung mittels Airlock Add-on-Modulen wie SOAP/XML/AMF-Filtern oder Virenscannern von Drittanbietern. Andere Arten von Injection-Angriffen oder Protokollverletzungen werden durch den von Airlock erzwungenen Protokollbruch verhindert.	– Eingebaute Blacklist-Filter – URL-Verschlüsselung – Smart Form Protection – Header Whitelisting – Cookie Store – Trennung der Security Domains – CAP-Schnittstelle – Protokollbruch – Add-on-Module

Über die OWASP
Das Open Web Application Security Project (OWASP) ist eine offene Community mit dem Ziel, Organisationen und Unternehmen bei der Verbesserung der Sicherheit von Webanwendungen zu unterstützen.
Im Vordergrund stehen dabei Werkzeuge, Methoden und Konzepte für eine sichere Entwicklung sowie der Schutz von Webanwendungen. Für weitere Informationen zur OWASP: *www.owasp.org*

Die OWASP Top 10
Die OWASP Top 10 werden ca. alle drei Jahre publiziert und stellen einen Überblick über die derzeit 10 grössten Schwachstellen und Sicherheitsrisiken für Webanwendungen dar.
Für weitere Informationen zu den OWASP Top 10: *www.owasp.org/index.php/Category:OWASP_Top_Ten_Project*

Schwachstelle	Beschreibung	Wie Airlock schützt	Funktionalitäten von Airlock
A2- Fehler in Authentifizierung und Session-Management	Anwendungsfunktionen, die die Authentifizierung und das Session-Management umsetzen, werden oft nicht korrekt implementiert. Dies erlaubt es Angreifern, Passwörter oder Session-Identifikatoren zu kompromittieren oder die Schwachstelle so auszunutzen, dass sie die Identität anderer Benutzer annehmen können.	Das HTTP-Protokoll selbst ist zustandslos. Deshalb werden Sessions normalerweise an die Session-ID gebunden, welche in einem Cookie oder URL-Parameter den Anfragen mitgegeben wird. Die Manipulation dieser Session-ID wird durch Verschlüsselung der URL und der Session Cookies verhindert. Airlock ersetzt standardmässig alle Applikations-Cookies durch sein eigenes Session Management (basierend auf der SSL Session-ID oder einem sicheren Airlock Session Cookie). Vorgelagerte Authentisierung stellt sicher, dass nur korrekt authentisierte Benutzer Zugriff auf die Applikationsserver erhalten. Es ist eine gute Idee, die Benutzerverwaltung an spezialisierte IAM-Komponenten zu delegieren. Airlock kümmert sich zentral um Idle Timouts, Session Lifetimes und Logout Propagation	– Vorgelagerte Authentisierung – Cookie Store – Cookie-Verschlüsselung – URL-Verschlüsselung – Sicheres Airlock Session Management
A3 – Cross-Site Scripting (XSS)	XSS-Schwachstellen treten auf, wenn eine Anwendung nicht vertrauenswürdige Daten entgegennimmt und ohne entsprechende Validierung und Kodierung an einen Webbrowser sendet. XSS erlaubt es einem Angreifer, Scriptcodes im Browser eines Opfers auszuführen und somit Benutzersitzungen zu übernehmen, Seiteninhalte zu verändern oder den Benutzer auf bösartige Seiten umzuleiten.	Anfragen, die XSS enthalten, werden durch eine Kombination von Blacklist-Filtern und dynamischen Whitelist-Filtern blockiert. URL-Verschlüsselung und Smart Form Protection verhindern die Modifikation von URL-Parametern und versteckten Formularfeldern. Die Herkunft von Inhalten kann durch den Einsatz von Content-Security-Policy Headern überprüft werden. Durch setzen des HttpOnly-Flags kann das Airlock Session Cookie vor Zugriffen aus Javascript Code geschützt werden.	– Cookie Store – Cookie-Verschlüsselung – URL-Verschlüsselung – Sicheres Airlock Session Management – Header Rewriting
A4 – Unsichere direkte Objektreferenzen	Unsichere direkte Objektreferenzen treten auf, wenn Entwickler Referenzen zu internen Implementierungsobjekten, wie Dateien, Ordner oder Datenbankschlüssel von aussen zugänglich machen. Ohne Zugriffskontrolle oder anderen Schutz können Angreifer diese Referenzen manipulieren, um unautorisiert Zugriff auf Daten zu erlangen.	Direkte Objektreferenzen können durch URL-Verschlüsselung und Smart Form Protection geschützt werden. Airlock blockiert Anfragen, falls sie manipulierte URL oder Formularfelder enthalten.	– URL-Verschlüsselung – Smart Form Protection

Schwachstelle	Beschreibung	Wie Airlock schützt	Funktionalitäten von Airlock
A5 – Sicherheitsrelevante Fehlkonfiguration	Sicherheit erfordert die Festlegung und Umsetzung einer sicheren Konfiguration für Anwendungen, Framework, Applikations-, Web- und Datenbankserver sowie deren Plattformen. Alle entsprechenden Einstellungen müssen definiert, umgesetzt und gewartet werden, da sie meist nicht mit sicheren Grundeinstellungen ausgeliefert werden. Dies umfasst auch die regelmässige Aktualisierung aller Software, inkl. der verwendeten Bibliotheken und Komponenten.	Airlock enthält Standardregeln, die regelmässig aktualisiert werden. Die mapping-orientierte Konfiguration ermöglicht es dem Administrator, selektiv nur den Zugriff auf bekannte Applikationen freizuschalten. Fehlermeldungen können umgeschrieben oder ersetzt werden, damit heikle Informationen (z.B. Stack Traces) nicht nach aussen weitergegeben werden.	– Standardkonfiguration – Mapping-orientierte Konfiguration – Content Rewriting – Error Page Replacement
A6 – Verlust der Vertraulichkeit sensitiver Daten	Viele Webanwendungen schützen sensitive Daten wie Kreditkarten- oder Authentisierungsinformationen ungenügend. Angreifer entwenden oder verändern solch ungenügend geschützte Daten, um Kreditkartenbetrug, Identitätsbetrug oder andere Straftaten zu begehen. Sensitive Daten müssen deshalb speziell geschützt werden, z.B. mittels Verschlüsselung von gespeicherten und übermittelten Daten oder über spezielle Vorkehrungen bei der Interaktion mit einem Browser.	Falls sensitive Daten in der URL oder einem Cookie enthalten sind, kann Airlock diese durch Verschlüsselung schützen. Standardmässig enthält Airlock Rewrite-Regeln, die es erlauben, sensitive Daten (wie z.B. Kreditkarteninformationen) aus Antworten herauszufiltern. Die korrekte Konfiguration von SSL ist nicht trivial. Ergon überwacht aktiv die Entwicklungen rund um SSL und stellt umgehend allfällige Sicherheits-Updates für Code oder Konfiguration zur Verfügung. Airlock, in seiner Funktion als Reverse Proxy, kann die Verbindung zum Browser mit TLS verschlüsseln. Fall notwendig können Antworten von Applikationen so umgeschrieben werden, dass sie nur HTTPS-URLs enthalten, selbst wenn die Applikation aus Performancegründen HTTP verwendet. Zusätzlich verbietet Airlock standardmässig den Einsatz von schwachen SSL-Verschlüsselungen und warnt vor abgelaufenen Zertifikaten. Passwort-Hashes sind ebenfalls sensitive Daten. Diese gehören nicht in die Applikationsdatenbank. Vorgelagerte Authentisierung löst dieses Problem durch die Delegation an einen spezialisierten Authentisierungsservice.	– URL-Verschlüsselung – Smart Form Protection – Cookie Store – Cookie-Verschlüsselung – Response Rewriting – SSL-Terminierung – Vorgelagerte Authentisierung
A7 – Missing Function Level Access Control	Die meisten Webapplikationen überprüfen die Zugriffsrechte auf Applikationsfunktionen, bevor diese im GUI angezeigt werden. Trotzdem müssen die Zugriffsrechte auf Serverseite nochmals geprüft werden. Geschieht dies nicht, können Angreifer gefälschte Anfragen erstellen, mittels derer sie Zugriff auf unautorisierte Funktionen erhalten.	Eine verschlüsselte URL mit einem sessionbasierten Schlüssel ist nur für eine einzige Benutzersession gültig. Der Schlüssel ist zufällig und kann nicht vorhergesehen werden. Damit wird der Workflow der Applikation geschützt, da der Benutzer nur Aktionen auslösen kann, die von der Applikation vorgesehen waren. Seiten, die dem Benutzer nicht von der Applikation präsentiert wurden, können nicht aufgerufen werden. Mittels vorgelagerter Authentisierung lassen sich feingranulare Zugriffsrechte für Applikationspfade definieren.	– Sessionbasierte URL-Verschlüsselung – Vorgelagerte Authentisierung

Schwachstelle	Beschreibung	Wie Airlock schützt	Funktionalitäten von Airlock
A8 – Cross-Site Request Forgery (CSRF)	Ein CSRF-Angriff bringt den Browser eines angemeldeten Benutzers dazu, einen manipulierten HTTP-Request an die verwundbare Anwendung zu senden. Session Cookies und andere Authentifizierungsinformationen werden dabei automatisch vom Browser mitgesendet. Dies erlaubt es dem Angreifer, Aktionen innerhalb der betroffenen Anwendungen im Namen und Kontext des angegriffenen Benutzers auszuführen.	Ein CSRF-Angriff kann verhindert werden, indem alle URLs mit einem sessionbasierten Schlüssel verschlüsselt werden. Die verschlüsselten URLs sind somit nur während einer einzigen Benutzersession gültig. Da die Verschlüsselung nicht vorhersehbar ist, können Angreifer keine gefälschten Anfragen erstellen, die für CSRF nötig sind. Da auch URLs in Formularen verschlüsselt werden, sind GET- und POST-Anfragen ebenfalls geschützt. Ein Nachteil der sessionbasierten URL-Verschlüsselung ist, dass die URLs nicht in den Favoriten gespeichert oder von Suchmaschinen indexiert werden können. Für die meisten Applikationen ist dies jedoch nicht relevant, da die URLs erst nach einem Login verfügbar sind.	– Sessionbasierte URL-Verschlüsselung
A9 – Benutzen von Komponenten mit bekannten Schwachstellen	Software-Komponenten wie Bibliotheken oder Frameworks laufen oft mit vielen Rechten. Sobald eine verwundbare Komponente ausgenützt wird, kann es deshalb zu ernsthaftem Datenverlust oder zur Übernahme eines Servers kommen. Applikationen, die verwundbare Komponenten enthalten, können andere Schutzmechanismen unterwandern und ermöglichen eine Vielzahl von Angriffen.	Airlock veröffentlicht regelmässig sicherheitsrelevante Updates und informiert Kunden, sobald Updates verfügbar sind. Airlock schützt sich selbst durch eine sichere Architektur gegen 0-day Attacken. Privilege Separation (SELinux) erzwingt die korrekte Abarbeitung von Anfragedaten. Der Web Listener darf z.B. nicht auf das Session-Management zugreifen oder Anfragen an das Back-end schicken.	– Protokollbruch – Security compartments – Update Mechanismus – Benachrichtigung bei Updates
A10 – Ungeprüfte Um- und Weiterleitungen	Viele Anwendungen leiten Benutzer auf andere Seiten oder Anwendungen um oder weiter. Dabei werden für die Bestimmung des Ziels oft nicht vertrauenswürdige Daten verwendet. Ohne eine entsprechende Prüfung können Angreifer ihre Opfer auf Pishing-Seiten oder auf Seiten mit Schadcode um- oder weiterleiten.	Airlock verifiziert Weiterleitungen, welche von den Applikationen kommen. Der Airlock Authentisierungsservice ermöglicht es, Weiterleitungen zu fixieren oder zu validieren. Location Parameters werden überprüft, z.B. bezüglich einer gültigen URL-Verschlüsselung.	– Authentisierungsservice – Filterung von Weiterleitungen – URL-Verschlüsselung

Ergon Informatik AG
Kleinstrasse 15
CH-8008 Zürich

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