# Starker Schutz für Web-Anwendungen

**Dr. Götz Güttich**

*Mit der Airlock Web Application Firewall bietet Ergon Informatik eine leistungsfähige Lösung zum Absichern von Web-Anwendungen. In der Regel müssen solche Applikationen nicht nur vor Angriffen und Sicherheitslücken geschützt werden, sondern es gilt auch dafür zu sorgen, dass nur diejenigen Anwender Zugriff auf die darin enthaltenen Daten bekommen, die dazu befugt sind. Deswegen kombiniert der Hersteller seine Web Application Firewall mit der Authentisierungslösung "Airlock Identity and Access Management". Gemeinsam sichern diese Tools nicht nur den Datenverkehr zwischen den Anwendern und den Applikationen, sondern sorgen auch dafür, dass eindeutig feststeht, welche Benutzer auf welche Anwendungen beziehungsweise welche Funktionen zugreifen können. Gleichzeitig lässt sich auf diese Weise eine einheitliche und zentrale Single Sign On-Infrastruktur einrichten. Wir haben die beiden Produkte im Testlabor unter die Lupe genommen.*

Wegen des großen Funktionsumfangs der beiden getesteten Lösungen mussten wir den Test in zwei Teile aufteilen. Dieser Teil befasst sich mit der Web Application Firewall (WAF), ihren Funktionen und ihrer Administration und der zweite Teil hat die Identity and Access Management-Lösung (IAM) zum Thema.


**Airlock WAF**

Die Airlock WAF arbeitet als Reverse Proxy und terminiert HTTP(S)-Verbindungen. Sie wird über ein zentrales Management-Interface gesteuert und bietet damit einen zentralen Punkt, um Policies für den Anwendungszugriff umzusetzen. Außerdem bringt sie eine große Zahl leistungsfähiger Filterfunktionen mit, unter anderem ICAP Content-Filtering und SOAP/XML-, AMF- und JSON-Filter.

Eines der Highlights der Airlock WAF 7 ist die API-Sicherheit mit dem Schutz von SOAP- und REST-Webservices. Dazu kommen Funktionen zum Policy Learning und zum dynamischen White Listing sowie Smart Form Protection, Cookie Protection und URL-Verschlüsselung.

Auch für die Integration in bestehende Sicherheitsinfrastrukturen stehen eine Vielzahl an Funktionen bereit: So lassen sich beispielsweise Malware-Scanner von Drittanbietern via ICAP integrieren, HSM-Geräte einbinden oder auch WAF-Meldungen an SIEM-Installationen schicken. Load-Balancing- und Failover-Features gehören ebenfalls zum Leistungsumfang. Für Unternehmen, die ihre WAF in Hardware-Form implementieren möchten, stellt Ergon Informatik zudem eine Appliance zur Verfügung.

**Funktionsweise**

Im Betrieb arbeitet die WAF zwischen der regulären Unternehmens-Firewall und den Anwendungen. Deswegen sollte sie in Produktivumgebungen über zwei Interfaces verfügen, um den Datenverkehr nach außen und innen sauber zu trennen. In Testumgebungen ist das nicht zwingend erforderlich.

Die Konfiguration der Absicherung der Datenübertagungen läuft über das so genannte Mapping ab. Dieses wird mit Hilfe einer übersichtlichen Seite innerhalb des Management-Werkzeugs eingerichtet, in der sich auf der einen Seite die virtuellen Hosts (die die geschützten Anwendungen nach außen hin repräsentieren) und auf der anderen – in der DMZ – die zu sichernden Applikationen befinden. Nach außen sind nur die Ports 80 und 443 offen.

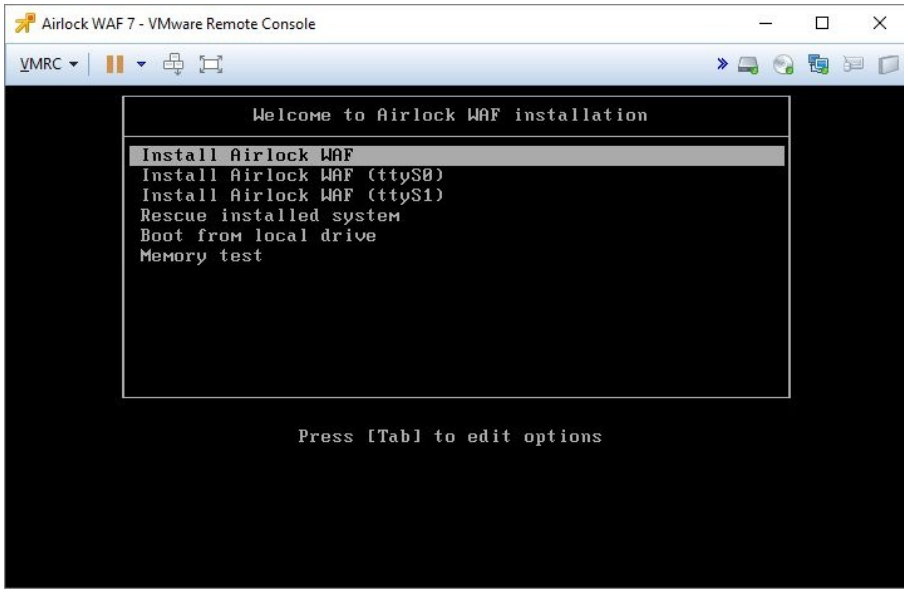
Um eine Anwendung abzusichern, ziehen die zuständigen Mitarbeiter mit der Maus Linien zwischen dem virtuellen Host, dem Mapping und der zu schüt-

zenden Applikation, um die Datenübertragungswege zu konfigurieren. Diese Verbindungen lassen sich jederzeit wieder trennen, beziehungsweise – je nach aktueller Anforderung – umleiten.

Aus Sicherheitsgründen sind bei einem neu angelegten Mapping zunächst immer alle dazugehörigen Regeln aktiv, damit keine unerwünschten Datenübertragungen stattfinden. Um das Mapping in den Regelbetrieb überzuführen, müssen die Administratoren folglich im nächsten Schritt die Konfiguration so anpassen, dass die von ihnen gewünschten Informationen fließen können.

Über das Threat Handling wird festgelegt, wie das System mit Policy-Verletzungen umgeht. Stellt die WAF beispielsweise fest, dass ein Anwender versucht, die Daten in einer Web Form zu manipulieren, so kann die Lösung entweder lediglich einen entsprechenden Log-Eintrag erzeugen, oder die Anfrage blockieren beziehungsweise gleich die ganze Sitzung komplett beenden.

Ein zentraler Bestandteil des Sicherheitskonzepts der Airlock WAF sind die „Deny Rules“. Diese realisieren ein so genanntes negatives Sicherheitsmodell, sie arbeiten also als Blacklists. Im Konfigurationsinterface repräsentiert jeder Deny Rule-Eintrag – also jede Zeile – eine Gruppe von Regeln. Jede dieser Gruppen befasst sich wiederum mit einem bestimmten Angriff. Der Eintrag "Exceptions" umfasst die angelernten und manuell erstellten Regeln.




**Die Installation der WAF sollte keinen IT-Administratoren vor unüberwindliche Hindernisse stellen**

Klappen die Administratoren in der Deny Rule-Konfiguration eine Gruppe auf, so sehen sie die verschiedenen Filter. Diese sind je nach Security Level unterschiedlich. Da strikte Filter mehr False Positives erzeugen als nicht strikte, ergibt es im Betrieb Sinn, das Sicherheitsniveau nur dort sehr hoch zu setzen, wo es unbedingt erforderlich ist. Der Security Level hängt folglich von der zu sichernden Anwendung und den jeweiligen Datenschutzanforderungen ab. In diesem Zusammenhang spielt es auch eine entscheidende Rolle, on ein Zugriff anonym aus dem Internet erfolgen darf, beispielsweise auf eine Login-Seite, oder ob bereits eine Anmeldung erfolgte und der Benutzer dem System somit bekannt ist.

Der Hersteller hat bereits eine sehr große Zahl an Regelgruppen für die verschiedensten Angriffe vordefiniert. Bei Bedarf sind die Anwender aber jederzeit dazu in der Lage, eigene Custom Rules anzulegen.

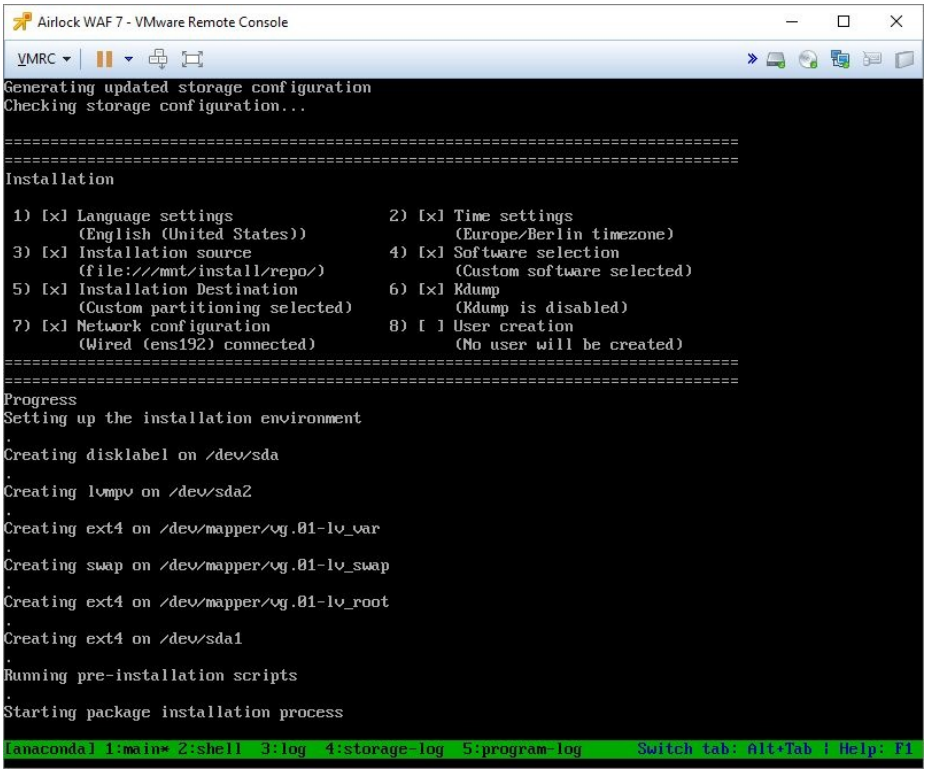
**Der Test**

Im Test installierten wir die Airlock WAF in unserem Testlabor, verbanden sie mit der zuvor erwähnten IAM-Lösung und nahmen die beiden Produkte in Betrieb. Für den Test sicherten wir mit der WAF den Zugriff auf das Web Interface der bei uns im Testlab verwendeten Network Monitoring-Lösung PRTG von Paessler. Zusätzlich nutzten wir eine vom Hersteller bereitgestellte Testumgebung mit einem Book-Shop als abzusichernde Applikation, um die Funktionalität der WAF im Detail unter die Lupe zu nehmen.

**Installation**

Die Airlock WAF basiert auf CentOS 7. Nach dem Download der Installations-ISO-Datei von der Webseite des Herstellers banden wir dieses File als Boot-Medium in eine virtuelle Maschine ein, die unter Vmware ESXi 6.5 Update 1 lief. Diese virtuelle Maschine verfügte über eine CPU mit 2,6 GHz Taktfrequenz und vier Cores, acht GByte RAM und 60 GByte Festplattenplatz. Nach dem Start des Systems von dem ISO-Abbild selektierten wir den Boot-Menü-Eintrag „Install Airlock WAF“. Daraufhin kam, wie unter Centos üblich, der In-

stallationsmanager „Anaconda“ hoch. Vor dem Beginn der eigentlichen Installation hatten wir die Möglichkeit, die Netzwerkkonfiguration für das Managementinterface anzupassen. Das System holte sich zwar eine funktionierende Konfiguration von unserem DHCP-Server, es dürfte aber in den meisten Fällen sinnvoll sein, der WAF eine feste IP-Adresse zu geben, was wir an dieser Stelle auch taten. Danach gaben wir die Zielfestplatte für die Installation an, was einfach war, da unsere VM nur über eine virtuelle Festplatte verfügte, und übernahmen die vorgegebene Partitionierung (diese lässt sich bei Bedarf auch anpassen). Zum Schluss legten wir die Zeitzone und das root-Passwort fest und definierten einen Benutzer für das Konfigurationswerkzeug. Danach lief die Installation durch, der ganze Vorgang nahm insgesamt etwa zehn Minuten in Anspruch. Nach dem Abschluss des Setups spielten wir erst einmal die aktuellsten Updates ein, um unsere Installation auf den letzten Stand zu bringen.




**Der eigentliche Installationsvorgang läuft über das Tool „Anaconda“ ab**

**Erste Konfigurationsschritte**

Nachdem das Setup durchgelaufen war, riefen wir über die URL https://{IP-Adresse des Management-Interfaces} das Verwaltungs-Tool der WAF auf. Dann meldeten wir uns mit unseren zuvor generierten Credentials an und wechselten nach „System Setup“, um unsere Lizenz einzuspielen.

Danach konnte es daran gehen, den Zugriff auf unsere erste Web-Anwendung abzusichern. Dazu war es lediglich erforderlich, im Management-Werkzeug im Menü „Application Firewall“ auf die „Reverse Proxy“-Seite zu wechseln und dort den Applikation Server mit seiner IP-Adresse und Portnummer als „Backend Server“ einzutragen. Existiert im Netz eine funktionierende DNS-Namensauflösung, so kann man anstelle der IP-Adresse auch den Namen des Systems angeben.

Sobald der „Reverse Proxy“ definiert wurde, kommt die Einrichtung eines neuen virtuellen Hosts an die Reihe, der die eingehenden Anfragen entgegen nimmt. Seine Definition erfolgt über einen Fully Qualified Domain Name und eine IP-Adresse mit Netzwerkmaske. Für die Verschlüsselung des Datenverkehrs über HTTPS können die zuständigen Mitarbeiter an dieser Stelle auch ein Zertifikat hinzufügen.

Nachdem das erledigt war, nahmen wir die Konfiguration des Mappings zwischen dem virtuellen Host und dem Backend Server vor. Dieses übernimmt – wie bereits angesprochen – die Verbindung zwischen dem virtuellen Host und der Backend-Anwendung und implementiert die Sicherheits-Logik. Es muss einen Entry- und einen Backend-Path enthalten. Der Entry-Path ist der Pfad, den die Benutzer von außen über ihren Browser aufrufen und der Backend-Pfad stellt den Pfad dar, an den die WAF die Anfragen weiterleitet. Auf diese Weise stehen eine Vielzahl an Konfigurationsmöglichkeiten offen. So ist beispielsweise ein symmetrisches Mapping denkbar, in dem jeweils ein Entry-Path die Daten auf eine Applikation – oder einen Teil davon – weiterleitet. So lassen sich ganze Anwendungen oder eben nur bestimmte Unterseiten absichern. Alternativ gibt es auch asymmetrische Mappings, in denen ein virtueller Host mit unterschiedlichen Entry-Paths wie "/Anwendung1", "/Anwendung2" und ähnlichem Daten entgegen nimmt, die dann an verschiedene Backend Server mit unterschiedlichen Applikationen weitergelei-

tet werden. Im Test konfigurierten wir zunächst einmal ein symmetrisches Mapping mit den Default-Vorgaben des Herstellers. Damit keine Datenübertragungen blockiert wurden, setzten wir dabei das oben bereits erwähnte Threat Handling auf „Log only“. Das ist bei neuen Mappings sinnvoll, da die zuständigen Mitarbeiter auf diese Weise analysieren können, was passieren würde, wenn die Blockierungsregeln der WAF wirklich aktiv wären. So lassen sich unerwünschte Verhaltensweisen verhindern und im Vorfeld die nötigen Anpassungen vornehmen.

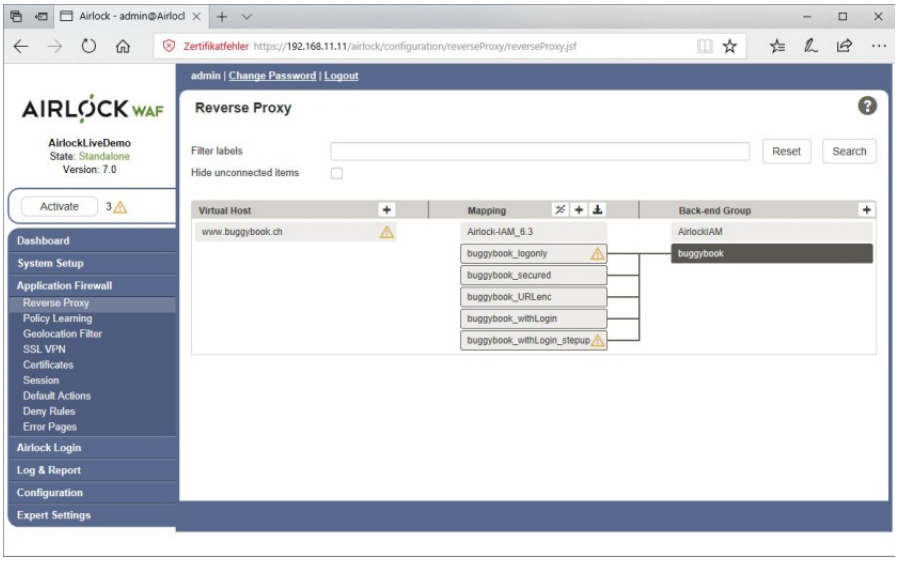
Zum Schluss erstellten wir mit der Maus noch die angesprochenen Verbindungen zwischen dem Mapping und dem Server sowie dem Host. Nachdem wir unsere Änderungen auf der WAF aktiviert hatten, war das System betriebsbereit und wir konnten uns mit dem virtuellen Host verbinden.

### Wichtige Funktionen

Gehen wir an dieser Stelle, bevor wir uns der Arbeit der WAF in der Praxis zuwenden, einmal kurz etwas genauer auf die wichtigsten Features des Produkts ein. In diesem Zusammenhang ist zunächst einmal der Schutz gegen Injection-Angriffe wie SQL-Injection oder XSS Schutz sowie gegen Forceful Browsing-Angriffe zu nennen. Forceful Browsing bedeutet, dass ein Angreifer versucht, Zugriff auf geschützte Seiten oder sensitive Informationen zu erhalten, indem er ihre URL direkt in die Adressleiste seines Browsers eingibt. Bei ungeschützten Systemen besteht die Möglichkeit, auf diese Weise etwa Zugriff auf Inhalte zu erhalten, die normalerweise erst nach einer Authentifizierung zur Verfügung stehen, oder detaillierte Informationen über die Konfiguration des betroffenen Web Servers in Erfahrung zu bringen. Um Forceful Browsing zu unterbinden, bietet die Airlock WAF eine Funktion zur URL-Verschlüsselung. Die verschlüsselten URLs sind dann im Betrieb nur während der laufenden Session gültig. Auf dieses Feature gehen wir später im Rahmen eines Use Cases noch genauer ein.

### DyVE

Bei DyVE (Dynamic Value Endorsement) handelt es sich um eine Funktion, die JSON-Objekte, die von Backends ausgeliefert werden, dynamisch nach Werten durchsucht, die für die aktuelle User Session erlaubt sind. Die Parameter oder JSON-Attribute von nachfolgenden Requests (wie etwa REST API-Aufrufen) lassen sich dann auf die Verwendung zulässiger Werte überprüfen. DyVE kann so unter anderem verhindern, dass ein Angreifer in einer Übertragung eine Kontonummer ändert, um beim Online Banking zu manipulieren.




**Die Konfiguration des Mappings definiert den Datenfluss zwischen virtuellem Host und zu schützender Anwendung**

Die Smart Form Protection sichert im Gegensatz dazu die Formulare der Web-Applikationen zur Laufzeit gegen Manipulationsversuche. Dazu stellt die WAF sicher, dass die Benutzereingaben den Vorgaben der Originalform der Anwendung entsprechen. Das verhindert zum Beispiel, dass ein Angreifer bei einem Eingabeformular, das vier Angaben erwartet, mit Hilfe von Manipulationen noch einen fünften Parameter übergibt. Außerdem schützt das genannte Feature vor Manipulationen der vorgegebenen Werte. Auch hierauf gehen wir später noch mit einem Use Case ein.

Session Fingerprinting bemerkt im Gegensatz dazu Änderungen von verwendeten Browsern oder IP-Adressen. In Verbindung mit Login-Informationen und ähnlichem, die das System von der IAM-Lösung bezieht, fällt damit im Betrieb auf, wenn ein Login beispielsweise um 9:00 Uhr aus Frankfurt und dann mit dem gleichen Benutzerkonto um 9:12 aus Bangkok kommt. In diesem Fall lässt sich das System so konfigurieren, dass beim zweiten Login automatisch ein zweiter Authen-

tifizierungsfaktor (etwa eine Sicherheitsabfrage) hinzukommt, um dafür zu sorgen, dass keine unbefugten Nutzer Zugriff auf die Applikation erhalten.

**Policy Learning**

Wird die Policy Learning-Funktion aktiviert, so untersucht die WAF den Datenverkehr, analysiert ihn und zeigt dann an, was genau von wo nach wo übertragen wurde. Außerdem macht sie Vorschläge, welche Regeln implementiert werden sollten, um das Sicherheitsniveau zu erhöhen. Diese können die Administratoren dann akzeptieren, modifizieren oder verwerfen. Diese Funktion lässt sich gut nutzen, um die Konfiguration im Detail an die jeweiligen Anforderungen anzupassen.

**Allow Rules**

Die Allow Rules dienen im Betrieb dazu festzulegen, welche Anfragen erlaubt sind. Sie stellen also White List dar. Die Allow-Regeln werden während der Analyse eines HTTP-Requests zuerst angewendet. Die Deny Regeln kommen erst zum Einsatz, wenn eine Allow Rule eine Anfrage zugelassen hat. Die Airlock WAF stellt auch eine Funktion zum Allow Rule Learning zur Verfügung, die sich beispielsweise nutzen lässt, um alle Eingaben während eines Login-Vorgangs zu erfassen und in Regeln umzusetzen, die die IT-Verantwortlichen dann akzeptieren können.

**Die Arbeit mit Labeln**

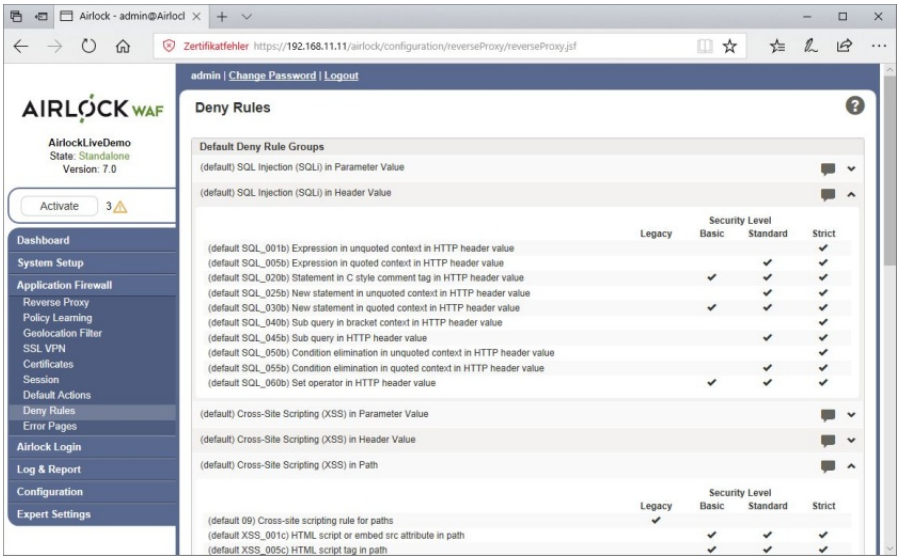
In der Praxis dürften die meisten Unternehmen in ihrer Konfiguration eine große Zahl der genannten Mappings – in manchen Umgebungen können das über 500 sein – verwenden, die die Datenübertragungen zwischen den virtuellen Hosts und den zu sichernden Anwendungen regeln. Um eine Übersicht über diese Mappings zu erhalten und die Konfiguration der Einträge zu erleichtern, haben die IT-Mitarbeiter die Option, sie mit Labeln zu versehen.

So gibt es zum Beispiel die Möglichkeit, einen Eintrag zu Exchange oder SharePoint mit "Microsoft" zu labeln oder auch einen Eintrag zu Exchange 2016 mit "Testumgebung" und einen Eintrag zu Exchange 2013 mit "Produktivumgebung". Gibt der Administrator dann den Label in der Suchzeile der Übersicht an, so zeigt ihm das System nur die dazugehörigen Mappings. Da sich die Label beliebig kombinieren lassen, steht den IT-Verantwortlichen mit ihnen ein sehr leistungsfähiges Feature zur Verfügung, das die Konfiguration sehr übersichtlich macht.

Möchte ein IT-Mitarbeiter zum Beispiel das Sicherheitsniveau für alle Microsoft-Anwendungen im Unternehmen ändern, so muss er lediglich den Microsoft-Label im Suchfeld eingeben und dann für alle Einträge auf einmal den Security Level modifizieren. Diese Funktion ließ sich im Test ohne Schwierigkeiten nutzen.

**Überwachung**

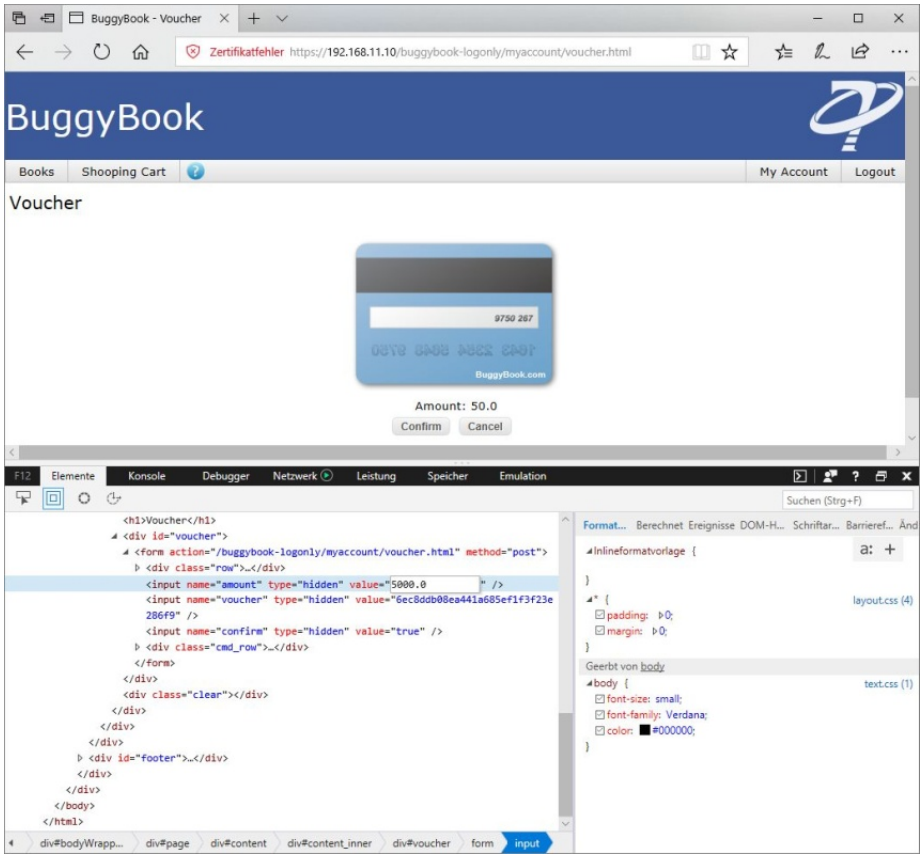
Was das Reporting und Logging angeht, so beherrscht die Airlock WAF wie angesprochen nicht nur die Weitergabe von Informationen an SIEM-Lösungen (mit den Log Formaten JSON und CEF), sondern bietet auch einen Log Viewer mit Filtern, dynamischer Darstellung relevanter Daten und vordefinierten Suchen. Dazu kommen diverse Dashboards und die Möglichkeit, eigene Auswertungen durchzuführen. Innerhalb des Log Viewers lassen sich darüber hinaus Drill Down-Vorgänge bis zu den eigentlichen Log-Einträgen hinunter realisieren. Zu den Überwachungswerkzeugen gehört unter anderem auch eine Karte. Wenn die Administratoren in diese hineinzoomen, erhalten sie zusätzliche Details zu den aufgerufenen Regionen und können so beispielsweise erfahren, dass ein Großteil der Angriffe der letzten Stunde aus einem bestimmten Land oder einer bestimmten Stadt kam.




**Die einzelnen Filter der Deny Rules wurden zu Gruppen zusammengefasst. Für jede Gruppe lässt sich das Sicherheitsniveau explizit festlegen.**

**Die Airlock WAF in der Praxis**

Für unsere nächsten Testschritte verwendeten wir die bereits angesprochene Demo-Umgebung mit dem Book-Store. Dieser nannte sich "BuggyBook" und verfügte, wie der Name es vermuten lässt, über eine große Zahl unterschiedlicher Sicherheitslücken. Zu Beginn wollten wir die Form Protection-Funktion unter die Lupe nehmen. Dazu loggten wir uns als Benutzer bei dem ungeschützten Book-Store ein und kauften ein Buch. Dadurch entstand auf dem Konto, das zu unserem Account gehörte, ein Minus. Um dieses Minus auszugleichen, wechselten wir anschließend in unsere Kontoverwaltung und gaben unter „Voucher“ einen Gutscheincode ein, der in dem Book Store 50 Euro wert war. Daraufhin zeigte uns die Online-Applikation eine Bestätigungsseite an, die uns den Betrag präsentierte, den der Voucher gutschreiben würde und es uns ermöglichte, diesen Betrag durch das Klicken auf „Confirm“ unserem Konto zu verbuchen. Das funktionierte einwandfrei.




**Um die Form Protection-Funktion zu testen, änderten wir über die Entwickler-Tools unseres Browsers den auf unserem Konto gutzuschreibenden Wert von „50“ auf „5000“**

Im nächsten Schritt wiederholten wir diese Vorgehensweise nochmals, klickten aber bei der Bestätigungsseite nicht auf „Confirm“, sondern riefen stattdessen die Entwicklungstools unseres Browsers auf und suchten im Quellcode nach dem Value-Eintrag, der die Höhe des gutzuschreibenden Betrags festlegte. Diesen änderten wir mit Hilfe der Entwicklungstools von 50 Euro auf 5000 Euro und klickten erst dann auf „Confirm“.

Daraufhin übergab der Browser diesen Wert an die Applikation. Da diese über keinen Schutzmechanismus verfügte, um den bestätigten Wert mit dem ursprünglichen Wert des Gutscheins abzugleichen, schrieb sie uns anschließend problemlos 5000 Euro auf unserem Konto gut. Die Test-Anwendung „BuggyBook“ benötigte also dringend die von der WAF bereitgestellten Sicherheitsfunktionen.

Um die eben demonstrierte Sicherheitslücke zu stopfen, ohne die Buch-Anwendung zu modifizieren, aktivierten wir in der WAF die Form Protection, die ja genau solche Aktionen unterbinden soll. Als wir nun unseren Betrugsversuch mit aktivem WAF-Schutz wiederholten, blockte die WAF die Übergabe des falschen Parameters und trug den Vorgang in ihre Log-Files ein.

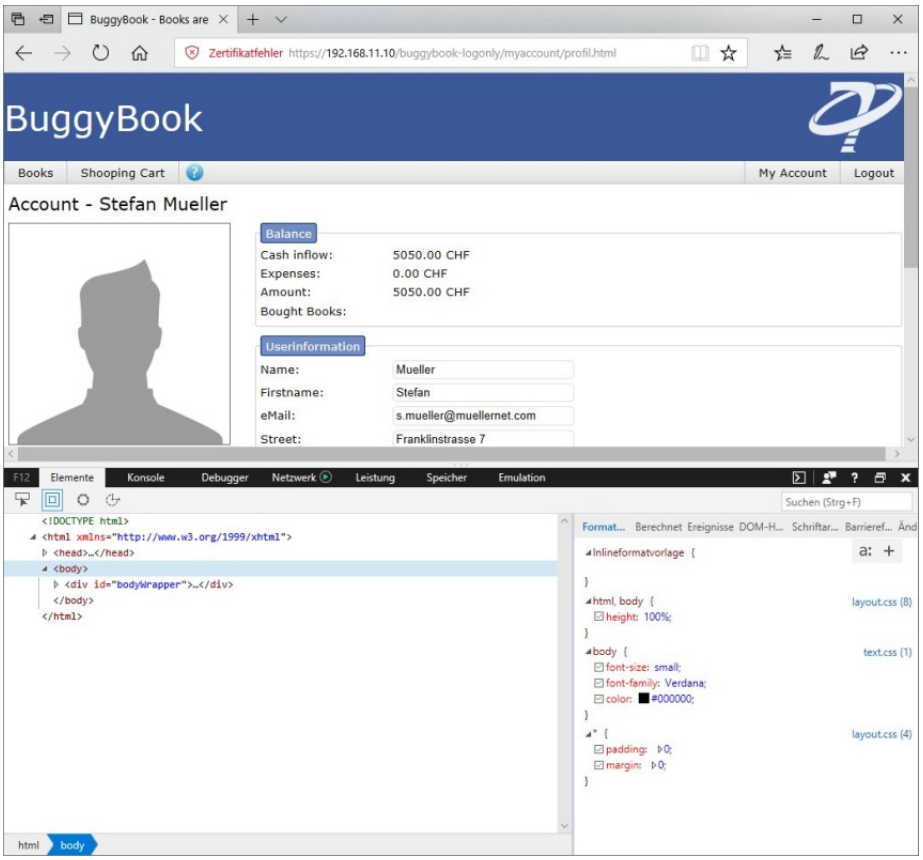
Im nächsten Use Case nahmen wir uns die Schutzfunktion vor Cross Site Scripting vor. Dazu schickten wir von unserem Book Shop-Benutzerkonto mit der shop-internen Message-Funktion eine Nachricht an einen anderen Benutzer. Bei dieser Nachricht fügten wir im Betreff ein Skript ein, das die Session Cookie-Informationen dieses Users auslesen sollte. Als wir uns nun als den anderen Benutzer anmeldeten und die Nachricht abriefen, öffnete sich ein Popup mit den Session-Infos des Anwenders. Mit Hilfe von Skripts lassen sich in der Praxis auf einem so unsicheren System eine Vielzahl von Angriffen realisieren.

Skripting-Angriffe wehrt die WAF über ihre Deny Rules ab. Im nächsten Schritt aktivierten wir folglich eine Deny Rule-Konfiguration, die verdächtige Muster in den Datenübertragungen erkannte. Konkret untersuchte diese Konfiguration die über-

mittelten Felder und die darin enthaltenen Parameter. Erkennt eine Deny Rule ein Muster, beispielsweise JavaScript, so blockiert sie unverzüglich den Request. Das funktioniert übrigens auch in JSON-Strukturen.

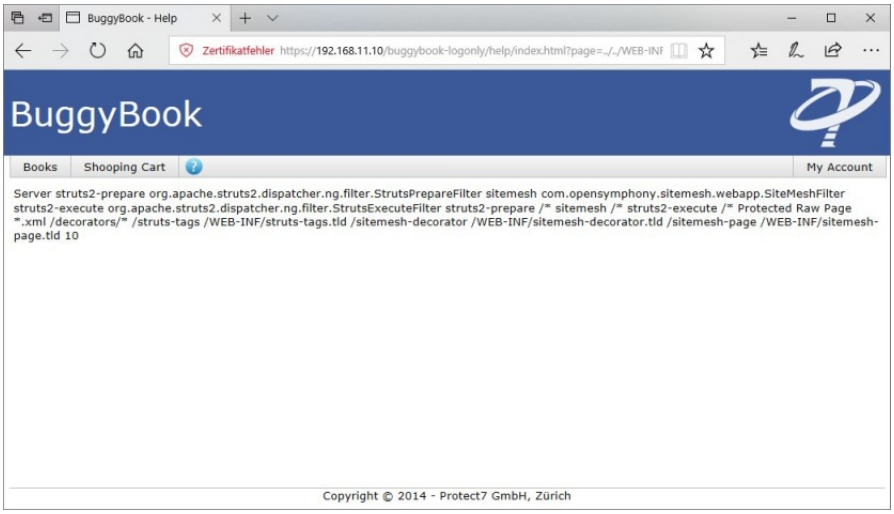
Nachdem unsere neue Konfiguration aktiv war, wiederholten wir unseren eben beschriebenen Angriff. Daraufhin blockte die WAF den Request und nahm ihre Aktion in die Log-Dateien auf.

Jetzt versuchten wir, über den Browser einen Forceful Browsing-Angriff durchzuführen. Dazu gaben wir die URL „https://{IP-Adresse des Servers} / {Bookstore} / help / index.html?page=../ ../ WEB-INF /web.xml;“ ein. Daraufhin präsentierte uns der Browser diverse Informationen über die Konfiguration des Systems, die wir als Hacker für weitere Angriffe nutzen könnten.




**Bei dem ungeschützten Book-Store war unser Betrugsversuch erfolgreich**

Diese Angriffe lassen sich über die URL-Verschlüsselungsfunktion der WAF unterbinden, die die Ziel-URLs auf Session-Ebene verschlüsselt und so sicherstellt, dass niemand direkt auf bestimmte Ziele auf dem Server zugreifen kann. Nachdem wir dieses Feature aktiviert hatten, lenkte das System unseren Zugriffsversuch einfach auf die Homepage des Webshops um, so dass kein Schaden entstand.




**Beim Forceful Browsing-Angriff lieferte der Online-Buchladen Informationen über die Konfiguration des Servers**

**Fazit**

Die Airlock WAF konnte uns im Test durch ihren großen Funktionsumfang überzeugen. Sie sorgt bei richtiger Konfiguration dafür, dass alles, was bei der zu sichernden Anwendung ankommt, bereits gefiltert und von potentiellen Bedrohungen gereinigt ist. Die geschützten Applikationen bekommen auf diese Weise Hacker-Angriffe überhaupt nicht mit.

Trotz des großen Funktionsumfangs wurde das Management-Interface verhältnismäßig übersichtlich gestaltet, so dass Sicherheitsadministratoren schnell dazu in der Lage sein sollten, mit der Lösung zu arbeiten. Besonders positiv fielen uns im Test das umfangreiche Reporting und Logging, die API-Sicherheit, die Smart Form Protection und die URL-Verschlüsselung auf. Administratoren, die gezwungen sind, ihre Anwendungen umfassend nach außen abzusichern, sollten unbedingt einen Blick auf das Airlock-Produkt werfen.