Best Practices Guide: Web Application Firewalls

Alexander Meisel
CTO art of defence

OWASP German Chapter

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

Ganz großes “Danke!!!” an die Autoren

- **Maximilian Dermann**
  - Lufthansa Technik AG
- **Mirko Dziadzka**
  - art of defence GmbH
- **Boris Hemkemeier**
  - OWASP German Chapter
- **Achim Hoffmann**
  - SecureNet GmbH
- **Alexander Meisel**
  - art of defence GmbH
- **Matthias Rohr**
  - SecureNet GmbH
- **Thomas Schreiber**
  - SecureNet GmbH

Inhalt

- Einführung und Zielsetzung
- Charakteristika von Web Apps bezgl. App Sec.
- WAF Fähigkeiten im Überblick
- Nutzen und Risiken von WAFs
- Schutz gegen OWASP TOP 10 (App vs. WAF vs. Policy)
- Kriterien zur Einsatz-Entscheidung von WAFs
- Best Practices bei Einführung und Betrieb

Einführung und Zielsetzung

- Einführung
  - Online Businesses
  - Schwachpunkt HTTP
  - Hinweis auf PCI DSS
- Definition des Begriffs “Web Application Firewall”
  - KEINE Netzwerk Firewall
  - Nicht nur Hardware
- Zielgruppe und Zielsetzung
  - Technische Entscheider
  - Betriebsverantwortliche, Sicherheitsverantwortliche
  - Applikationseigner

Charakteristika von Webapplikationen hinsichtlich Web Application Security

- Übergeordnete Aspekte im Unternehmen
  - Priorisierung hinsichtlich Bedeutung
    - Zugriff auf personenbezogene Daten von Kunden, Partnern ...
    - Zugriff auf Betriebsgeheimnisse
    - Zertifizierungen
- Technische Aspekte
  - Entwicklung: Test- und Quality-Assurance
  - Vollständige Dokumentation (Architektur, Code)
  - Wartungsverträge

Fähigkeiten von Web Application Firewalls im Überblick

- Einordnung von WAFs im Bereich Web App Sec
  - WAFs sind ein wichtiger Teil einer "Defense in Depth" Strategie
  - Hauptziele (nachträgliche Absicherung, Grundschutz)
  - Zusätzliche Funktionen (Session Management, ...)
- Typische Schutzmechanismen von WAFs
  - Tabelle mit (gewollter) Funktionalität
    - Beispiele: CSRF, Session fixation, *-Injection
  - Bewertung:
    - + kann eine WAF sehr gut
    - - kann die WAF schlecht oder gar nicht
    - ! abhängig von WAF/Anwendung/Anforderung
    - = kann teilweise von einer WAF übernommen werden

Tabelle (Nur ein kleiner Auszug)

Problem	WAF	Maßnahme
Cookieschutz	+	Cookies können signiert werden
	+	Cookies können verschlüsselt werden
	!	Cookies können vollständig versteckt bzw. ausgetauscht werden (Cookie-Store)
	!	Cookies können an die anfragende IP gebunden werden
Information-Leakage	+	Cloaking-Filter, ausgehende Seiten können "bereinigt" werden (Fehlermeldungen, Kommentare, unerwünschte Informationen)
Session-Riding (CSRF)	+	URL-Encryption / Token
Session-Timeout	!	Timeout für aktive und inaktive (idle) Sessions kann festgelegt werden (wenn die WAF die Sessions selbst verwaltet) Auch wenn die Sessions von der Applikation bereitgestellt werden, kann die WAF diese bei entsprechender Konfiguration erkennen und terminieren.
Session-Fixation	=	kann verhindert werden, wenn die WAF die Sessions selbst verwaltet

Nutzen und Risiken von Web Application Firewalls im Überblick (I)

- Hauptnutzen von WAFs
  - Grundschutz (Baseline Security)
  - Compliance
  - “Hotfixing” oder “Just-in-time patching”
- Zusatznutzen (abhängig von Funktionalität)
  - Zentrales Logging, Alarmieren und Reporting
  - SSL Terminierung
  - URL-Verschlüsselung
  - Authentifizierung
  - ….

Nutzen und Risiken von Web Application Firewalls im Überblick (I)

- Risiken beim Einsatz von WAFs
  - False positives
  - Erhöhte Komplexität in IT-Infrastruktur
  - Trainingsaufwand
  - Potentieller Einfluss auf Webapplikation (wenn WAF zum Beispiel Session-Management terminiert)

Schutz gegen OWASP-TOP 10 - WAFs und andere Methoden im Vergleich

- Drei verschiedene Webanwendungsklassen:
  - T1: Webapplikation in Desgin-Phase
  - T2: produktive Anwendung (MVC-Controller), einfach anpassbar
  - T3: produktive Anwendung, nicht oder schwer anpassbar
- OWASP TOP 10 - Tabelle mit Komplexitäts-abschätzung zur Behebung von Problemen
  - in der Anwendung selbst
  - mit einer Web Application Firewall
  - durch Umsetzung einer Policy

OWASP Top 10 (Auszug)

Top10		Typ	Kommentar	Aufwand
A1	Cross-Site Scripting (XSS)	T1	z. B. durch konsequenten Taglib-Einsatz (Java), oder Controls (ASP.NET), zusätzlicher Frameworks (PHPIDS).	1
		T2	Eingabe-Enkodierung lässt sich nur schwer (z. B. mittels OWASP Stinger) einbauen, besser geht es hier mit einer vorgelagerten WAF. Bei .NET-Anwendungen lässt sich XSS-Filter aktivieren.	3 (.NET: 2)
		T3	Bei .NET-Anwendungen XSS-Filter aktivieren.	- (.NET: 2)
		WAF	WAF ermöglicht in diesem Fall keine Ausgabevalidierung, da sie den Kontext der Daten nicht kennt. Die Validierung muss schon bei der Eingabe erfolgen und kann eventuell mit der Ausgabe korreliert werden	2
		P		-
A2	Injection Flaws	T1	Kann durch Verwendung eines OR-Mappers (z. B. Hibernate) oder konsequente Parametrisierung aller Eingaben (z. B. Stored Procedures oder besser: Prepared Statements) vermieden werden. Andere Injection Flaws (z. B. bei XML) lassen sich ggf. nur	1

1: Aufwand gering 2: mittlerer Aufwand 3: hoher Aufwand -: nicht umsetzbar

Kriterien zur Einsatz-Entscheidung (I)

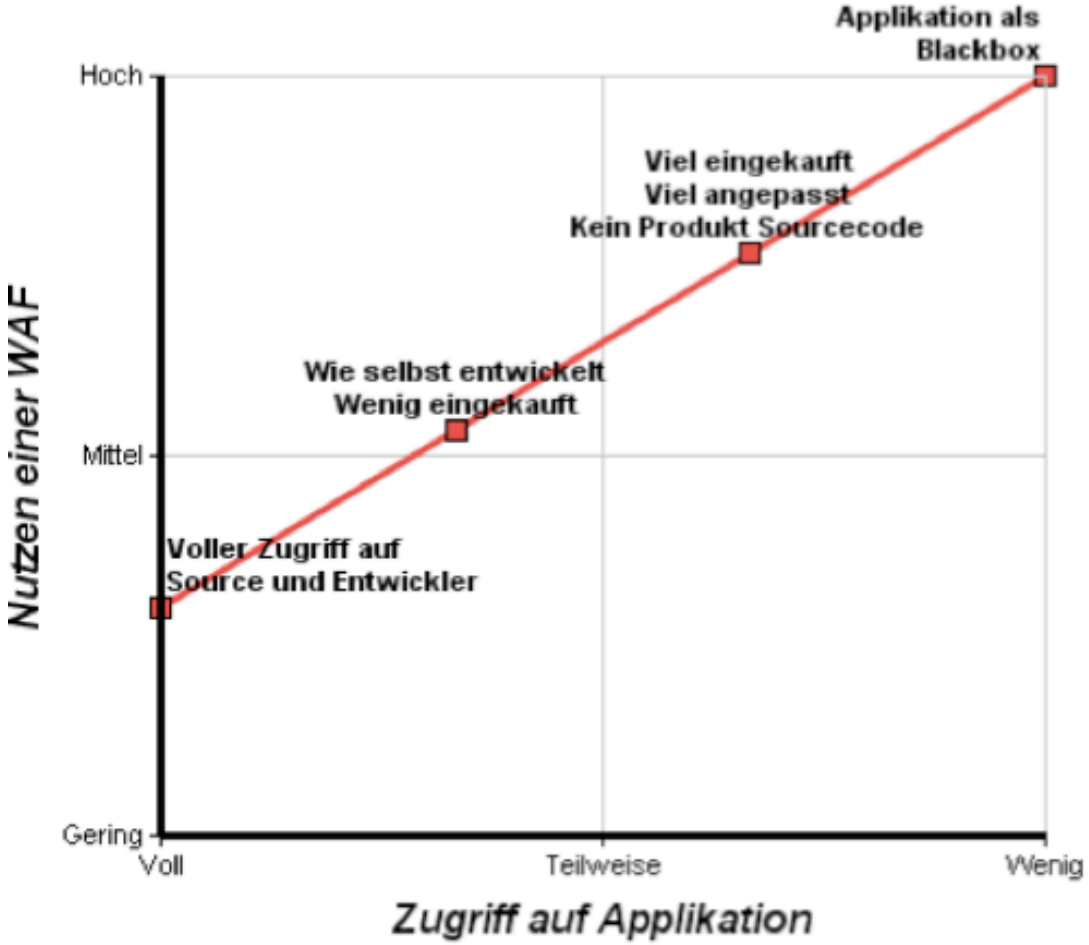
- Unternehmensweite Kriterien
  - Bedeutung der Webapplikation für den Unternehmenserfolg
  - Anzahl der Webapplikationen
  - Komplexität
  - Betriebsaufwände
  - Performance
  - Skalierbarkeit

Kriterien zur Einsatz-Entscheidung (II)

- Kriterien hinsichtlich einer Webapplikation
  - Zugriff auf Webapplikation
  - Dokumentation
  - Wartungsverträge
  - kurze Fehlerbehebungszeiten
- Bewertung und Zusammenfassung
  - Benutzen der Checkliste (im Anhang des Dokuments)
- Wirtschaftlichkeit
  - P: Vermeidung von wirtschaftlichen Schäden
  - P: Geringere Kosten durch frühzeitig behobene Probleme
  - Einsparungen durch Nutzung zentraler Dienste

Kriterien zur Einsatz-Entscheidung (II)

- Leitfaden zur Entscheidungsfindung



Best Practices bei Einführung und Betrieb (I)

- Aspekte der vorhandenen Web-Infrastruktur
  - Zentrale oder dezentrale Infrastruktur
    - Zentrale Proxy-Anwendung
    - Host basierte Installation
    - Virtuell !!???!!!
  - Performance
    - GBits/Second Datendurchsatz ist NICHT entscheident
    - HTTP Anfragen pro Sekunde verarbeiten
    - Gleichzeitige Clients (Benutzer) auf Webanwendung
    - Hochlastphasen (Weihnachten steht vor der Tür)

Best Practices bei Einführung und Betrieb (II)

- Organisatorische Aspekte
  - Einhaltung bestehender Security Policies
    - Policies sollten (wenn möglich) nicht verändert werden
      - Beispiel: SSL Terminierung
  - Neues Rollenmodel
    - Anwendungsverantwortlicher WAF
      - Einkauf und Evaluierung einer WAF
      - Verständnis von WAF Fähigkeiten
      - Alarm und Fehler Management
      - Regelwerkänderungen
      - Bester Freund der Entwicklungsabteilungen!

Best Practices bei Einführung und Betrieb (III)

- Iteratives Vorgehen bei der Implementierung - vom Grundschutz zur Vollversiegelung
  - Schritt 1: Festlegung der Verantwortlichen
    - Idealerweise mit Hilfe des vorgestellten Rollenkonzepts
  - Schritt 2: Grundschutz für alle Webapplikationen
    - Anfangs Black-Listing mit Hersteller Signaturen
    - Überwachung und Eliminierung von False-Positives
  - Schritt 3: Priorisierte Liste von Anwendungen mit erhöhtem Sicherheitsbedarf
    - Benutzung der Checkliste (Anhang am Paper)
  - Weitere Schritte: Vollversiegelung laut Liste
    - Lernmodi, Code Review, Pentests

Anhänge

- Checkliste: Zugriff auf eine Webapplikation unter Security Gesichtspunkten
  - Je mehr Punkte pro Applikation gesammelt werden desto höher ist der Zugriff auf die Anwendung
- Beschreibung neues Rollenmodell
  - Plattformverantwortlicher WAF
    - Konzernweites Management
  - Anwendungsverantwortlicher WAF
    - Implementierung der Regeln
    - Überwachung und Betreuung der WAF
  - Anwendungsverantwortlicher
    - Betrieb oder Entwicklung der fachlichen, zu schützenden Applikation

Wo findet man das Paper?

- OWASP Wiki
  - https://www.owasp.org/index.php/Best_Practices:_Web_Application_Firewalls

Danke!

Antworten!!!

Diskussion???

Mitmachen!!!!

Alexander Meisel
alexander.meisel@artofdefence.com