Was eine WAF (nicht) kann

Mirko Dziadzka

OWASP Stammtisch München 24.11.2009

Inhalt

- Meine (subjektive) Meinung was eine WAF können sollte und was nicht
- Offen für andere Meinungen und Diskussion
- Disclaimer:
  - Ich arbeite für einen WAF Hersteller
  - Ich gebe hier meine Meinung wieder

Wer bin ich?

- Studium Mathe/Informatik
- 93-2003 Vorlesungen Unix und Security
- Unix-lastiger Techniker
- Seit 10 Jahren im Umfeld
  - Beratung - Entwicklung - Betrieb
  - Security - Scalability

Was ist eine WAF?

- OWASP: "eine Schutzlösung auf Webanwendungsebene, die technisch nicht von der Anwendung selbst abhängig ist."
- Eine Komponente, die den Datenverkehr zwischen Browser und Webapplikation untersucht, ggf. ändert oder blockiert.
- Unabhängig von der Applikation
- Kein B2B SOAP Filter

Funktionalitäten

- Request-Filterung
  - Blacklist - Bekannte Angriffe (ähnlich einem Virenscanner)
- Whitelist - was ist ungefährlich
  - Allgemein
  - Applikations-spezifisch

Funktionalitäten

- Response Filterung
  - Data Leakage (keine KK-Nummern)
  - Fehlermeldungen (Stackdump, SQL Error Meldungen)
  - Nachträgliches erkennen von Angriffen oder Angriffsversuchen
  - Malware Detection

Funktionalitäten

- Sessions
  - Sichere Session zw. Browser und WAF
  - Setzt mehrere Requests in Verbindung
  - Bestimmung gültiger Daten
    - auto-whitelist
    - URLs, Formularfelder, Cookies, ...

Funktionalitäten

- Authentisierung, Autorisierung
  - von Benutzern
    - zentrale Authentisierung, SSO
    - Applikationsunabhängig
  - von IP Addressen

Funktionalitäten

- Normalisierung der HTTP Requests
  - IMHO: Nein, WAF soll Entscheidungen fällen und nicht kaputtes HTTP eines Clients reparieren
- Aber: Sinnvoll, damit WAF und Applikation dieselbe Sicht auf die Daten haben (zum Beispiel bei der Interpretation von Multipart-Mime Headern)

Warum braucht es eine WAF?

- Das kann man doch alles in der Applikation lösen?

- Ja, aber.

  - Applikation nicht änderbar, hot patching, PCI compliance, Zweites Sicherheitsnetz

  - Neue Funktionalität

Administration

- Ziele: Einfachheit und Nachvollziehbarkeit
- Versionierung des Regelwerks, Rollback, Audit-Log (Nachvollziehbarkeit)
- Optional: Granulares User und Rechtekonzept
- Optional: Clustermanagement
- Logview, Statistiken, Reports, Alerting, ...

Regelwerkerstellung

- Mitgelieferte Blacklists a.k.a. Grundschutz
- Hilfe bei der Erstellung von Whitelists passend zu den Applikationen
- Einstellung anderer Funktionalitäten:
  - Cookie Protection, Formular Protection, ...
- Regelwerk testen (wie mache ich das?)

Deployment

- Mode: Detection, Blocking, Modifying
  - Je nach Anforderung sind andere Szenarien denkbar, grade im Zusammenhang mit SSL!
- Reverse Proxy: terminiert TCP + SSL
- Transparente Bridge: Kann evt. in SSL reinschauen, aber nicht modifizieren

Deployment

- Softwareplugin im Webserver
- Softwareplugin in der Servlet-Engine
- Plugin in Firewall, Load-Balancer, ...
- WAF in der Cloud?

Probleme

- Encoding / Charset
  - Der Browser deklariert nicht(!) in welchem Zeichensatz die reinkommenden Bytes zu interpretieren sind, der Web-App und die WAF müssen raten.
- Die WAF muss an die Applikation angepasst werden.

Probleme

- Interpretation von Standards
  - Multipart-Mime (siehe Beispiel)
  - Web ist von Anfang an "schwammig" definiert, die Parser für HTTP sind "programming by example."
  - Es gibt eine Teilmenge des Standards, die von allen gleich verstanden wird.

Probleme

- AJAX / Web 2.0
  - Das klassische Request-Response Modell gibt es nicht mehr.
  - URL-Encryption, Formfield Protection, etc. müssen pro Applikation angepasst werden.
  - Code wird dynamisch generiert.

Probleme

- Die WAF braucht ein Modell der Applikation
  - Applikationen können WAF und WAF-Admin freundlich sein oder nicht
  - korrektes HTML
  - Variablennamen, die etwas aussagen und konsistenz über die ganze Applikation sind
  - URLs die unterscheidbar sind vs. index.php

Probleme

- Das Thema Performance hängt immer vom Regelwerk mit ab.
- Eine WAF ist kein "Ein Klick und ich bin sicher" Gerät, die Konfiguration ist deutlich komplexer als die einer klassischen Firewall.

Performance

- Traue keiner Statistik …
- Mbit/sec ist nicht sehr aussagekräftig, Request/sec ist schon besser.
- Aber was für Requests? “GET / HTTP/1.0”.
- Und welches Regelwerk?
- Muss die WAF nur den Request oder auch den Response parsen? HTML verstehen?

Performance

- WAF Hersteller brauchen Standardisierte Benchmarks!
- OWASP Projekt???
- Was messen? Anzahl gleichzeitiger Benutzer bei definiertem Antwortverhalten.
- Req/seq vs. Latenz

Performance

- Selber messen! Je nach Deployment, braucht man andere Werte.
- Wie kann ich skalieren wenn meine App um den Faktor 10 mehr Benutzer bekommt? Faktor 100?

Aktuelle Vorfälle

- libri.de 29.10.2009
  - http://www.heise.de/security/meldung/Libri-laesst-Kundenrechnungen-offen-im-Netz-liegen-845001.html
- Sparkassenverlag 3.11.2009
  - http://www.heise.de/newsticker/meldung/Zugriff-auf-Rechnungen-im-Sparkassen-Shop-moeglich-Update-848538.html

Auswahlkriterien WAF

- Welches Ziel will ich erreichen?
- Wie "dynamisch" ist meine Applikation
- Welche Architektur will ich? RP vs. Plugin
- Welche Betriebsprozesse muss ich beachten
- Welche Ressourcen habe ich

Fragen?

Links

- http://www.owasp.org/index.php/Best_Practices:_Einsatz_von_Web_Applicatio n_Firewalls
- http://www.suspekt.org/downloads/POC2009-ShockingNewsInPHPExploitation.pdf
- http://mirko.dziadzka.de/