OWASP AppSec Germany 2009 Conference
http://www.owasp.org/index.php/Germany

Pentestvorbereitung: Sitemapping

Achim Hoffmann
OWASP Member
SecureNet GmbH, München
achim@owasp.org ah@securenet.de
+49 89 32133 631

OWASP
Nürnberg, 13.9.2009

Pentestvorbereitung: Sitemap ... Probleme ...

- “Anwendungsprofil” erstellen ➔ Sitemap
- Quantität: wieviele URLs, Parameter (Formulare)
- Qualität: Technik, Logik der Anwendung
- ➔ Aufwandsabschätzung
  - Kunde
  - Dienstleister/Pentester

Es werden nicht alle Schuhe über den gleichen Leisten gemacht.

Über

- Achim Hoffmann
- Senior Security Consultant bei SecureNet GmbH
  - Webapplikationen bei SecureNet seit 1998
  - Web Application Security seit 2001
  - BSI: Sicherheit von Webanwendungen Maßnahmenkatalog und Best Practices
  - OWASP: Best Practice Web Application Firewalls (WAF)
  - OWASP: Best Practice Projektierung der Sicherheitsprüfung von Webanwendungen
  - WASC: Web Application Threat Classification
  - WASC: WAF-EC WAF Evaluation Criteria

Ein Experte weiß über immer weniger immer mehr, bis er am Ende über nichts alles weiß.

Zielgruppe

- Anwendungseigner
- Anwendungsbetreiber
- Penetrationstester

Jeder weiß selbst am besten, wo ihn der Schuh drückt.

Schritt für Schritt

- Aufwandsabschätzung
- Anwendungseigner, Anwendungsbetreiber, die einen Penetrationstest beauftragen wollen, müssen den Aufwand (Links, Parameter) angeben
- Penetrationstester brauchen einen Überblick über die zu testende Anwendung

Aufbruch

➔Sitemap

Schritt für Schritt: Testmethodik

Penetrationstester orientieren sich an einer definierten Testmethodik:

- OSSTMM

  http://www.osstmm.org/

- OWASP Testing Guide

  http://www.owasp.org/index.php/Category:OWASP_Testing_Project

Allen Methodiken gemeinsam: 1. Schritt

Ziel (1. Schritt):

Informationsbeschaffung (Discovery)

Auch die längste Reise beginnt mit dem ersten Schritt.

Schuhgröße

Fragestellung:

1. Wieviele Anwendungen?
2. Wieviele Seiten/Formulare pro Anwendung?
3. Wieviele Parameter haben diese?
4. Was ist zu Testen (XSS, SQLI, RFI, usw., ...)?

➔Wieviele Testfälle ergeben sich daraus?

Ein enger Schuh drückt, ein großer stolpert.

Schuhgröße: Zahlenspiele

1. Anwendungen: 40
2. Seiten/Formulare: 400000
3. Parameter: 5000
4. Testfälle: XSS, SQLI, ... 20

---

5. 1 Request+Response pro Sekunde

➔ URL einmal aufrufen:
400000 / 3600 = 110 Stunden

➔ pro URL ein Parameter testen:
400000 * 20 / 3600 = 2200 Stunden

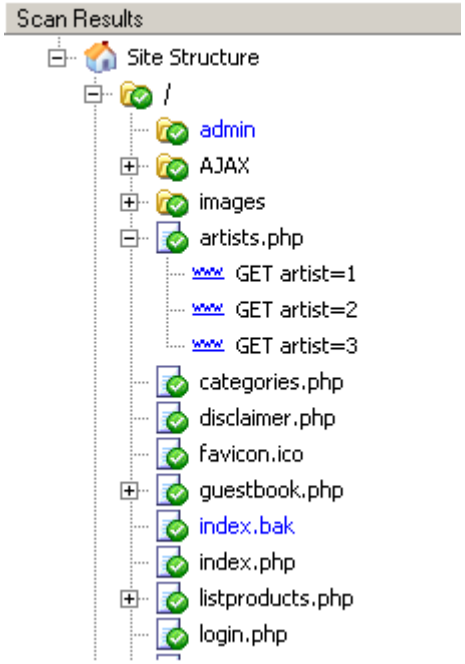
(reine Rechenzeit, **keine** Auswertung!)

Wenn man das Ziel nicht kennt, ist kein Weg der richtige.

Sitemap: Was ist das?

Vollständige, hierarchische Struktur, also alle Einzelselseiten und Einzeldokumente des Internetauftritts

Idealerweise mit allen Verknüpfungen



Sitemap: Wozu?

- Überblick über die Anwendung (Website)
- Erkennen der “Einstiegspunkte”
- Erkennen der Business-Logik / Workflow
- Benutzereingaben

Schuhe, Boots, Mokassins, ...

Für das (nachträgliche) Erstellen einer Sitemap gibt es unterschiedliche Bezeichnungen:

- Spider
- Crawler

Pantoffel oder High Heels?

Wie funktioniert eine Webanwendung?

- “klassisch”: eine URL pro Formular
- Dispatcher/Controler: nur eine URL
  (mit z.B. `aktion=suche` Parameter)
- Datengetrieben, multi-step-Anwendung
  (Form1 -> Form2 -> Form3)
- einmalige Aktionen (Login, Registrierung)
- Mischung aus obigen

Schrittfolge

HTTP ist zustandslos, darum:

Session-IDs in der Webanwendung

- HTTP Authentication (Basic, Digest, NTLM)
- HTTP-Cookie-Header (kurz Cookies genannt)
- URL-Parameter (manchmal auch URL-Rewriting genannt)
- Form-Parameter

Stolpersteine

Die Art der Webanwendung (“klassisch”, Dispatcher, usw.) ist entscheidend für Auswahl eines Tools.

Weitere Probleme für Erstellung der Sitemap:

- nur “klassische” Anwendungen problemlos
- Dispatcher fast unmöglich
- datengetrieben unmöglich
- multi-step nur begrenzt (meist kommerzielle Tools)
- HTTP-Header (z. B.: JSON, XmlHTTPRequest)
- Session-IDs (Cookie, URL-Parameter, usw.)
- Login

Ich weinte, weil ich keine Schuhe hatte, bis ich einen traf, der keine Füße hatte.

Schuhwerk: Tools

- curl, wget
- HTTrack

  http://www.httrack.com/
- SiteScope

  http://www.foundstone.com/us/resources/proddesc/sitescope.htm
- Site Map Builder

  http://www.sitebapbuilder.net/
- ntoinsight

  http://www.ntobjectives.com/freeware/index.php
- burp, paros, WebProxy

  http://portswigger.net/

  http://parosproxy.org/
- Grendel, w3af

  http://grendel-scan.com/

  http://w3af.sourceforge.net/

Man muß Schuhe suchen, die den Füßen gerecht sind.

Am Ziel?

- False Negatives: alle Links gefunden?
- False Positives: Link mehrfach getestet?
- Daten der Sitemap exportierbar?
- Sitemap graphisch darstellbar?
- Weiterverwendung der Daten möglich (Scanner)?

Fragen?