webtech CONFERENCE | 2010

Carsten Eilers | www.ceilers-it.de

Pentesters Toolbox

Vorstellung

- Berater für IT-Sicherheit
- Autor
  - PHP Magazin, Entwickler Magazin
  - Buch „Ajax Security“
  - www.ceilers-news.de
  - und anderes ...
- Schwachstellen-Datenbank

Agenda

- **Pentest - Was ist das, wie geht das?**
- Tools zum Erkunden
- Tools zum Testen
- Tools zum automatischen Test
- Firefox-Erweiterungen
- Bekannte finden

Pentest - Was ist das, wie geht das? (1)

- Suche nach Schwachstellen
  (Hier: In Webanwendungen)

- Manuell

- Schwachstellenscanner

Pentest - Was ist das, wie geht das? (2)

- Vorgehensweise:
  1. Erkunden
     „Was haben wir denn da?“
  2. Testen
     „Geht da vielleicht dieses oder jenes?“
     „Was passiert, wenn...?“
  3. Überprüfen
     „Lässt sich das Gefundene ausnutzen?“

Erkunden - Zuerst manuell

- Webbrowser nach Wahl
- Einmal durch die Anwendung surfen
- HTML-Seiten sichern
- nachgeladene Dateien sichern

Später oder parallel für jede Seite alle vorhandenen Parameter notieren

Erkunden ist einfach, oder?

Demo

Erkunden kann mühsam sein

Was bei kleinen Websites problemlos von Hand geht, ist bei großen mühsam...

... aber wir haben ja einen Computer zur Hand...

Die Lösung: Ein Tool (oder zwei oder ...)

Agenda

- Pentest - Was ist das, wie geht das?
- **Tools zum Erkunden**
- Tools zum Testen
- Tools zum automatischen Test
- Firefox-Erweiterungen
- Bekannte finden

Paros Proxy

- Proxy
- speziell für Pentests
- in Java geschrieben
- u.a. auch Spider (gleich)
  und (primitiver) Scanner (später)

Paros Proxy

Demo

Paros Proxy Spider

Statt selbst zu surfen, surfen lassen

Problem: Doppelte Seiten im Ergebnis

Der Spider kennt keine Kataloge, Foren,...

Paros Proxy Spider

Demo

Paros Proxy Nachteile

- „etwas“ älter
  Letzte Version: 10. November 2004, 3.2.0Alpha

- Forks:
  - andiparos
    Client-Zertifikate auf Smartcards
  - Zed Attack Proxy (ZAP)
    seit kurzem OWASP-Projekt

Testen - zuerst manuell

- Testwerte eingeben
- Ergebnis auswerten

Werte:

- <script>alert(1)</script>
- '
- http://www.ein.example/?

Testen ist einfach, oder?

Demo

Testen kann kompliziert sein

- Viele Parameter unzugänglich:
  - Cookies
  - POST-Formulare mit versteckten Feldern
- Clientseitige Prüfungen stören
  - <script>alert(1)</script> ist weder E-Mail-Adresse noch BLZ

Lösung: Ein Tool... (oder ...)

Agenda

- Pentest - Was ist das, wie geht das?
- Tools zum Erkunden
- **Tools zum Testen**
- Tools zum automatischen Test
- Firefox-Erweiterungen
- Bekannte finden

OWASP WebScarab

- Framework zur HTTP/HTTPS-Analyse
- Proxy
- speziell für Pentests
- in Java geschrieben
- Plugins, u.a. Spider, Fuzzer und XSS/CSRF-Analyse

OWASP WebScarab

Demo

OWASP WebScarab Nachteile

- eigentlich keine
- Nachfolger (in Entwicklung): WebScarab NG
  - über Java Webstart erhältlich
  - vieles fehlt noch

Eine Alternative: Burp Suite

- Plattform zum Testen von Webanwendungen
- Proxy, Spider und mehr
- Scanner in kommerzieller Version
- in Java geschrieben

Burp Suite

Demo

Agenda

- Pentest - Was ist das, wie geht das?
- Tools zum Erkunden
- Tools zum Testen
- **Tools zum automatischen Test**
- Firefox-Erweiterungen
- Bekannte finden

Scanner

Bisher war Handarbeit nötig:

- Testwert eingeben
- Ergebnis mit erwarteter Ausgabe vergleichen

Kann der Computer das nicht selbst?

Scanner in Parox Proxy

Ziemlich eingeschränkt:

- HTTP PUT
- indexierbare Verzeichnisse
- obsolete Dateien
- reflektiertes XSS
- Default-Dateien auf Websphere-Servern

w3af

- Web Application Attack and Audit Framework
- in Python geschrieben
- vielfältig konfigurierbar

w3af

Demo

skipfish

- „Web Application Security Reconnaissance Tool“
- in C geschrieben
- sehr schnell
- „Fuzzing“ für Dateinamen & Endungen mittels Wörterbücher
- PHP: Keine RFI, keine eval()-Injection

skipfish

Demo

Wapiti

- „Web Application Security Auditor“
- in Python geschrieben
- umfangreich (u.a. RFI, eval()-Injection)

Wapiti

Demo

Spezialisten

Allgemeine Scanner liefern allgemeine Ergebnisse

Wie sieht es mit Spezialisten aus?

Mini MySqlat0r

- Sucht SQL-Injection-Schwachstellen
- in Java geschrieben

Mini MySqlat0r

Demo

XSSploit

- Sucht XSS-Schwachstellen
- in Python geschrieben

XSSploit

Demo

fimap

- Sucht File-Inclusion-Schwachstellen
- in Python geschrieben
- kann auch exploiten
- Nachteile:
  - keine Cookies
  - „dynamic RFI“ ist überflüssig (feste Suffix kann durch ?foobar= neutralisiert werden)

fimap

Demo

JavaScript dekodieren

„obfuscated“ JavaScript - Freund oder Feind?

```
eval(function(p,a,c,k,e,d)
{e=function(c){return(c...
```

JavaScript dekodieren

Demo

JavaScript dekodieren

manuelle Lösung:

`eval(` durch

`document.write(` ersetzen

Onlinetool: Wepawet

Agenda

- Pentest - Was ist das, wie geht das?
- Tools zum Erkunden
- Tools zum Testen
- Tools zum automatischen Test
- **Firefox-Erweiterungen**
- Bekannte finden

Die Firefox-Variante (1)

- Firebug und Firecookie

- XSS Me
- SQL Inject Me
- Access Me
  (Zugriffskontrolle, z.B. fehlende Session-ID)

Die Firefox-Variante (2)

- HackBar

- Modify Headers
- Tamper Data

- JavaScript Deobfuscator
  (greift ausgeführtes Skript ab =>
  GEFAHR!)

Die Firefox-Variante (3)

- Wappalyzer
  erkennt Webanwendungen

Agenda

- Pentest - Was ist das, wie geht das?
- Tools zum Erkunden
- Tools zum Testen
- Tools zum automatischen Test
- Firefox-Erweiterungen
- **Bekannte finden**

Bekannte (Schwachstellen) finden

- Nikto
  - Webserver-Scanner
  - sucht nach bekannten Schwachstellen
  - in Perl geschrieben

Webanwendungen erkennen

- Blind Elephant
  - vergleicht typische Dateien mit deren gespeicherten Hashwerten
  - in Python geschrieben

- WhatWeb
  - Plugins zur Erkennung z.B. typischer Strings
  - in Ruby geschrieben

Fragen?

Vielen Dank...

... für Ihre Aufmerksamkeit

Material und Links auf
www.ceilers-it.de/konferenzen/