Cross-Site Scripting Filter Evasion

Alexios Fakos
Senior Security Consultant
n.runs AG
alexios.fakos@nruns.com

OWASP
25.11.2008

The OWASP Foundation
http://www.owasp.org

Bevor es losgeht …

Voraussetzungen

- Sie sind vertraut mit den Buzzwords
  - Cross-Site Scripting
  - Filter

Der Vortrag behandelt nicht

- Wie man einen (sicheren) Filter erstellt

- Bespricht keine neuen (bahnbrechende) XSS Filter Umgehungen

Kaffeepause ?

Ziel des Vortrags

- Welche Filtermechanismen existieren
- Überblick verschaffen
- Auf dem aktuellen Stand bleiben

Danksagung

Neben dem RSnake XSS Cheat Sheet …

http://ha.ckers.org/xss.html

Generell jede Person, die sich mit Beiträgen beteiligt hat:

http://sla.ckers.org/

Agenda

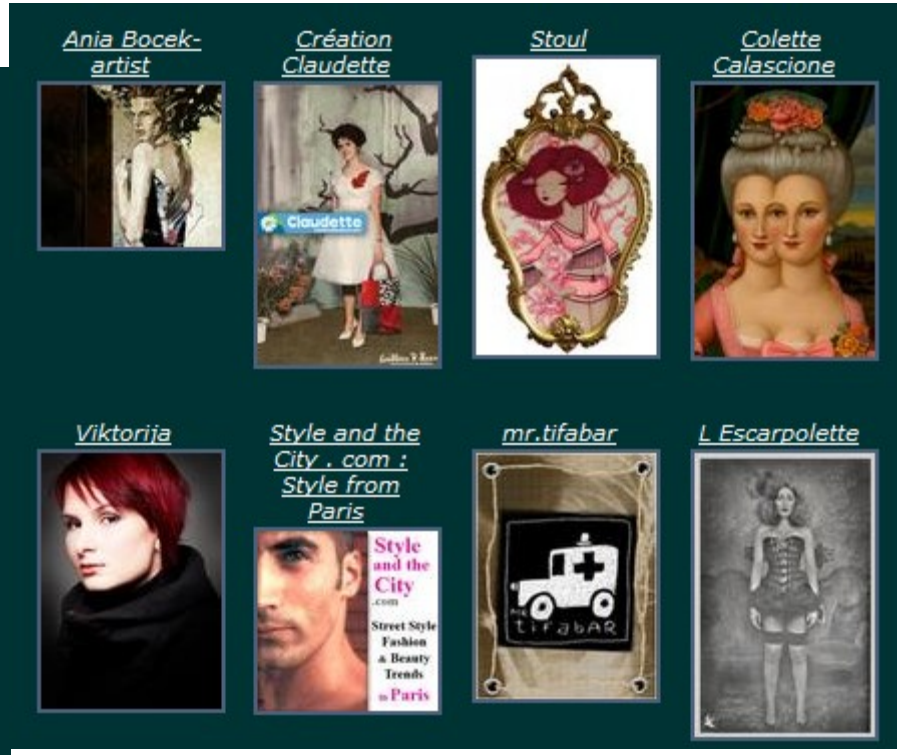
- Einführung
- Problemdefinition
- Problemlösung
- XSS Filterung
- Umgehungsvarianten
- Beispiele
- Fazit

Agenda

- **Einführung**
- Problemdefinition
- Problemlösung
- XSS Filterung
- Umgehungsvarianten
- Beispiele
- Fazit

Einführung

- Benutzerdefinierte Seiten
  - Business gestützte Anforderung ➔ krit. Erfolgsfaktor
  - “Express yourself” oder Social Networking



Agenda

- Einführung
- **Problemdefinition**
- Problemlösung
- XSS Filterung
- Umgehungsvarianten
- Beispiele
- Fazit

Problemdefinition

- Wie man mit Benutzereingaben umgehen soll
  - Grundsätzlich:
    Ein- und Ausgabevalidierung als zweiseitiges Schutzschild
  - Whitelist Ansatz favorisieren
  - Ausgaben(daten) entsprechend kodieren

Problemdefinition

- Umgang mit Benutzereingaben
  - Realität oder
    Die Erfolgsgeschichte des
    sozialen Netzwerks

    - Whitelist funktioniert nicht
      - “Express yourself” Daten sind in keiner Weise berechenbar
    - Kodierung ist unangemessen
      - “Express yourself” Daten müssen exakt dargestellt werden

Agenda

- Einführung
- Problemdefinition
- **Problemlösung**
- XSS Filterung
- Umgehungsvarianten
- Beispiele
- Fazit

Problemlösung

- Konsequenz

Blacklisting von Benutzereingaben

Agenda

- Introduction
- Problem definition
- Mitigation
- **XSS Filterung**
- Umgehungsvarianten
- Beispiele
- Fazit

XSS Filterung

- Generelle Ansätze
  - String search and deny input
  - Substitution and continue with the data flow
  - Regular expressions

XSS Filterung

- String search and deny input
  - Eingabe enthält “böses” Zeichen/Wort ?
    - Benutzereingaben werden verworfen
    - Fehlermeldung

  - Nicht oft gesehen
    - Vielleicht nicht benutzerfreundlich in dem Web 2.0 Zeitalter
    - Oder WAF läuft im passiven Modus …

XSS Filterung

- Substitution and continue
  - Eingabe mit definierten “bösen” Werten ?
    - Ersetze „böses“ mit “harmlosen” Daten
    - Weiterverarbeitung mit “neuen validierten” Benutzerdaten
  - Oft eingesetztes Verfahren
    - Keine Fehlermeldung
    - Benutzerfreundliches Verhalten

XSS Filterung

- Regular expressions (seek and destroy)
  - Eingabe mit definierten “bösen” Werten ?
    - Substitution and continue or
    - String search and deny
  - De-facto Standardlösung
    - Ansatz: Substitution and continue
    - RegExp als Schweitzer Messer

XSS Filterung

- Fehlerstricke ?
  - Definition von “bösen” Daten

  - Jeder vorgestellte Filteransatz hat Vor- und Nachteile

XSS Filterung

- Vorteile – Fazit

  - "Verboten" als fachliche Anforderung klingt immer gut
  - Alle Ansätze sind relativ einfach umzusetzen
  - Alle Ansätze sind „benutzerfreundlich/-orientiert" Ausname: string search and deny input

XSS Filterung

- Nachteile – Fazit
  - Welche Daten sind “böse”
  - Schon laut Definition nicht vollständig
  - Berücksichtigen Groß-/Kleinschreibung und Kodierung nicht
  - Substitution kann „harmlose” Daten beeinträchtigen

XSS Filterung

- Nachteile – Fazit
  - Datenkorrektur kann sicherheits-relevante Probleme provozieren
  - Datenkorrektur kann logische Fehler verursachen
  - RegExp kann zur einer Herausforderung werden
  - Regulärer Ausdruck kann logische Fehler verursachen

XSS Filterung – Reguläre Ausdrücke

- Wer kennt den
  „match only at end of string" modifier ?

XSS Filterung – Reguläre Ausdrücke

- Beispiel

```
$> perl -e "print (\"joe\n\" =~ m/^[a-z]+$/);“
```

XSS Filterung – Reguläre Ausdrücke

- Wer hätte das gewusst ?

```
$> perl -e "print (\"joe\n\" =~ m/^[a-z]+$/);“
⇨ 1
```

XSS Filterung – Reguläre Ausdrücke

- Der „match only at end of string” modifier

$> perl -e "print int((\"joe\n\" =~ m/[a-z]+**\z**/));“

⇨ 0 ⇦

Agenda

- Introduction
- Problem definition
- Mitigation
- Filtering XSS
- **Umgehungsvarianten**
- Beispiele
- Fazit

Umgehungsvarianten

- Technik
  - Logische Fehler
  - Third-Party Probleme

- Mensch
  - Falsche Annahmen

Umgehungsvarianten – Technik

- Logische Fehler
  - „Böses" Zeichen fehlt
  - Falscher oder fehlender modifier in der RegExp
  - Auswahl der RegExp Bibliothek
  - Falscher Evaluierungsausdruck
    - Vergleichs- oder Zuweisungsoperator ?
    - Logisches UND/ODER oder Bit Vergleich ?
  - Falsche Annahmen von der "sicheren" API
  - Endlosschleife aufgrund von Parsingfehler

Umgehungsvarianten – Technik

- Third-Party Probleme
  - Webserver
  - Browser
  - Browser Plug-Ins / Erweiterungen
  - Betriebssystem
  - API Misbrauch

Umgehungsvarianten – Third-Party

- Websserver
  - Fehlerseiten
    - Apache – CVE-2002-0840, CVE-2007-6203, CVE-2008-2168
    - IIS 4.0, 5.0, 5.1 - CVE-2002-0148
  - Module / Erweiterungen
    - mod_negotiation - CVE-2008-0455
    - mod_imap - CVE-2007-5000
  - Nicht gehärtete Systeme – Samples
    - IIS CSS – CVE-2000-0746
    - Tomcat hello XSS – CVE-2007-1355

Umgehungsvarianten – Third-Party

- Browser
  - HTTP Response Handling (UTF-7 Universal XSS)
    - IE – CVE-2007-1114
    - Opera – CVE-2007-1115
    - FF - CVE-2007-5415, CVE-2007-0996
  - Parsing / Kanonisierung
    - Anhängen von Zeichen in HTML-Tag Attribute (onload<u>..</u>="doEvil(); ") - CVE-2007-0995
    - ASCII Zeichen / 7 versus 8 bit - CVE-2006-3227
    - 0x00 Bytes

Umgehungsvarianten – Third-Party

- Browser
  - Parsing / Kanonisierung
    - Unicode
      - Variable width encoding
      - Whitespace / linefeed characters
      - Best fit mapping
      - Swallowing
    - Unusual JavaScript
    - Invalid HTML
    - Fragmentation
    - MIME-Type Handling
    - RSS Feeds
    - E4X und weiteren (“neuen”) HTML Erweiterungen

Umgehungsvarianten – Third-Party

- Browser Plug-Ins / Erweiterungen
  - Flash / Silverlight
    - Einbetten von verwundbare Flash Dateien
      main.swf?baseurl=asfunction:getURL,javascript:alert(1)// CVE 2007-6244, CVE 2007-6637
    - Facebook Advisory von Jouko Pynnonen
  - Adobe Universal XSS – CVE-2007-0045
  - Firebug, Sarge RSS Reader – CVE-2007-1878, CVE-2007-1947, CVE-2006-4712
  - Infiltrierung von Dateien
    - MOV, MP3, PDF, ASF, Bilder etc.

Umgehungsvarianten – Mensch

- Falsche Annahmen
  - Umgang mit Daten
    - Webserver
    - Browser
  - XSS Filter besteht XSS cheat sheet ➔ Filter ist sicher
  - API fehlinterpretiert
  - Vertrauenswürdige Third-Party Dienste
  - Vertrauenswürdige interne Daten Dienste

➔ Oft die Ursache von logischen Fehlern

Agenda

- Introduction
- Problem definition
- Mitigation
- Filtering XSS
- Types of Evasion
- **Beispiele**
- Fazit

Beispiele – Technik

- Logische Fehler
  - „Böses" Zeichen/Wort fehlt ;-)

Horde Framework forgot
<frame> and <frameset> in the XSS Filter

CVE-2007-6018

Beispiele – Technik

- Logische Fehler
  - „Böses" Zeichen oder Wort (ver)fehlt ;-)

Eval() functions are re-written by the FirePass engine but double eval() functions are ignored

CVE-2007-0188

Beispiele – Technik

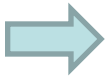
- Logische Fehler
  - Falscher oder fehlender modifier in der RegExp

```
perl -e"$v='<SCRI<script>PT>ddd';
        $v =~s/<script>//g; print $v;"
```

Beispiele – Technik

- Logische Fehler
  - Falsche oder fehlender modifier in der RegExp

```
perl -e"$v='<SCRI<script>PT>ddd';
        $v =~s/<script>//g; print $v;"
```



<SCRIPT>ddd

Beispiele – Technik

- Logische Fehler
  - Auswahl der RegExp Bibliothek
  - POSIX Bibliothek in PHP ist NICHT binary safe !!!
    - ereg* Funktionen behandeln das NULL Byte als das Ende einer Zeichenkette
  - Schwerwiegende Schwachstellen wurden in der Perl-Compatible Regular Expression (PCRE) Bibliothek gefunden

Beispiele – Technik

- Third-Party
  - Browser

Wer kennt den Notepad Bug

➔ “this app can break” ?

Beispiele – Technik

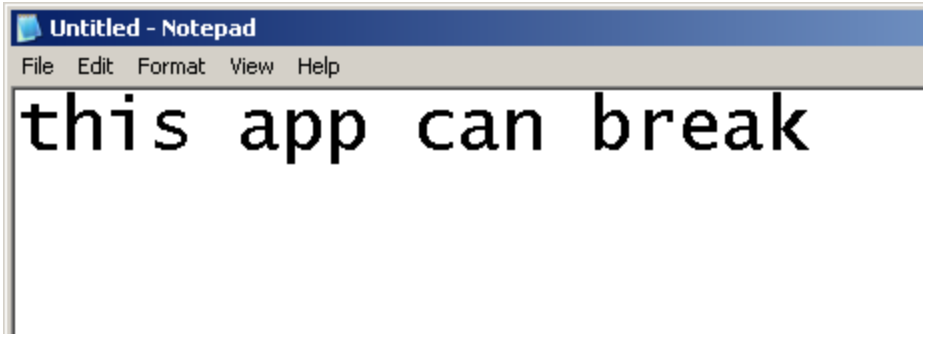
- Third-Party
  - Browser

notepad.exe unter Windows XP öffnen
und folgenden Text eingeben

“this app can break”

Beispiele – Technik

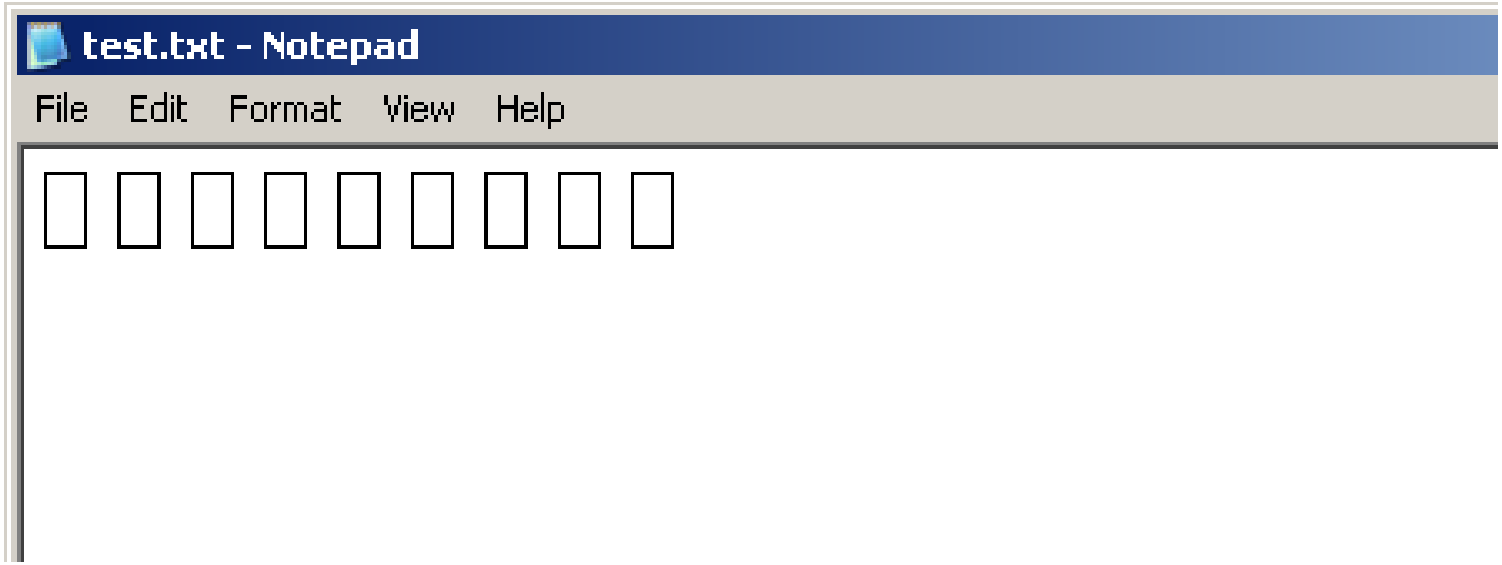
- Third-Party
  - Browser



  - Datei speichern, z.B. “test.txt”
  - Notepad schließen und öffnen Sie die Datei erneut !

Beispiele – Technik

- Third-Party
  - Browser



Beispiele – Technik

- Third-Party
  - Browser

Beispiele – Technik

- Third-Party
  - Browser

Notepad muss mit unterschiedlichen Zeichenkodierungen umgehen !

Beispiele – Technik

- Third-Party
  - Browser

Notepad muss mit unterschiedlichen Zeichenkodierungen umgehen !

…manchmal wird geraten…

Beispiele – Technik

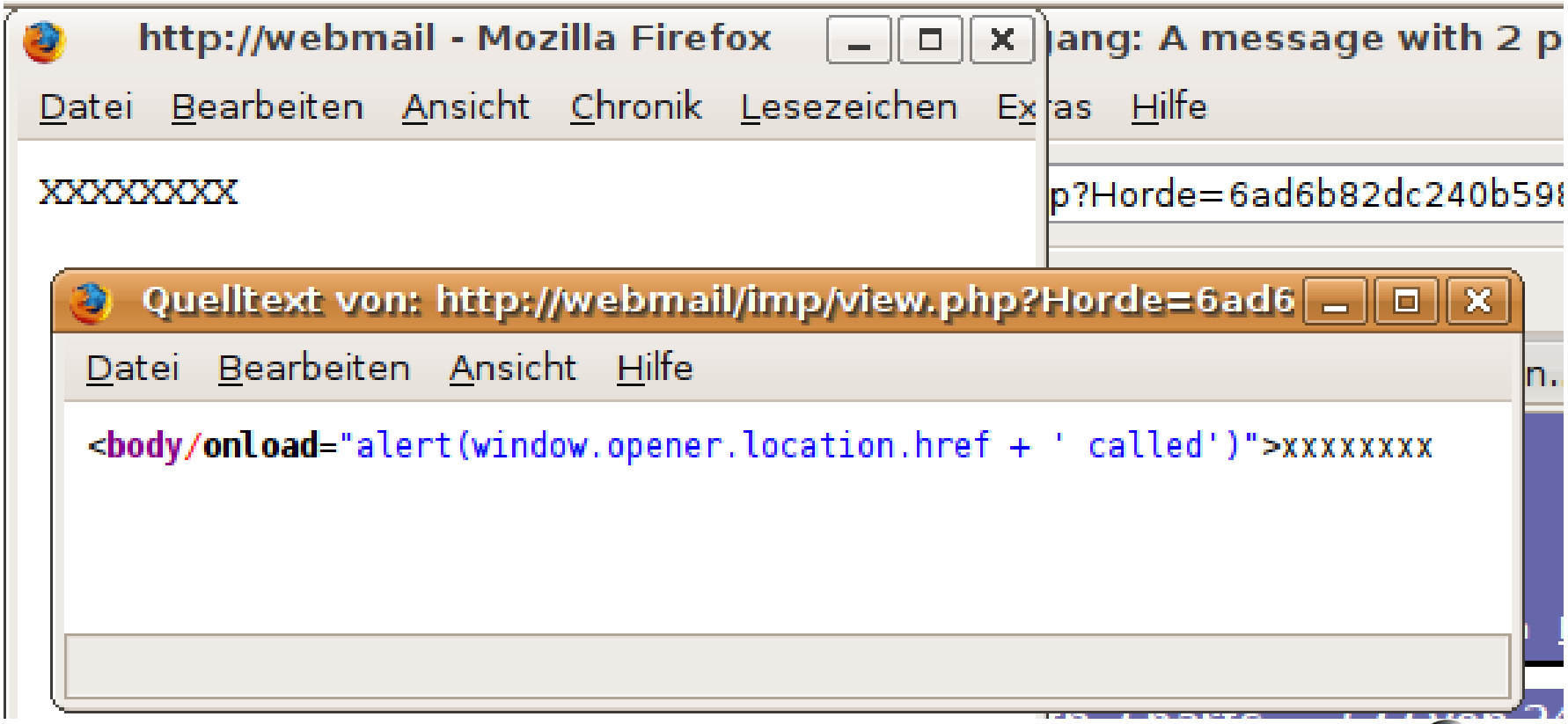
- Third-Party
  - Browser – Whitespace / linefeed characters
  - HTML 4 Spezifikation
    http://www.w3.org/TR/REC-html40/struct/text.html
    - ASCII space ( )
    - ASCII tab (	)
    - ASCII form feed ()
    - Zero-width space () (Thai)
    - U+000A LINE FEED (LF)
    - U+000D CARRIAGE RETURN (CR)

Beispiele – Technik

- Third-Party
  - Browser – Whitespace / linefeed characters
  - Aber auch weitere Zeichen werden als Leerzeichen interpretiert !!!

Beispiele – Technik

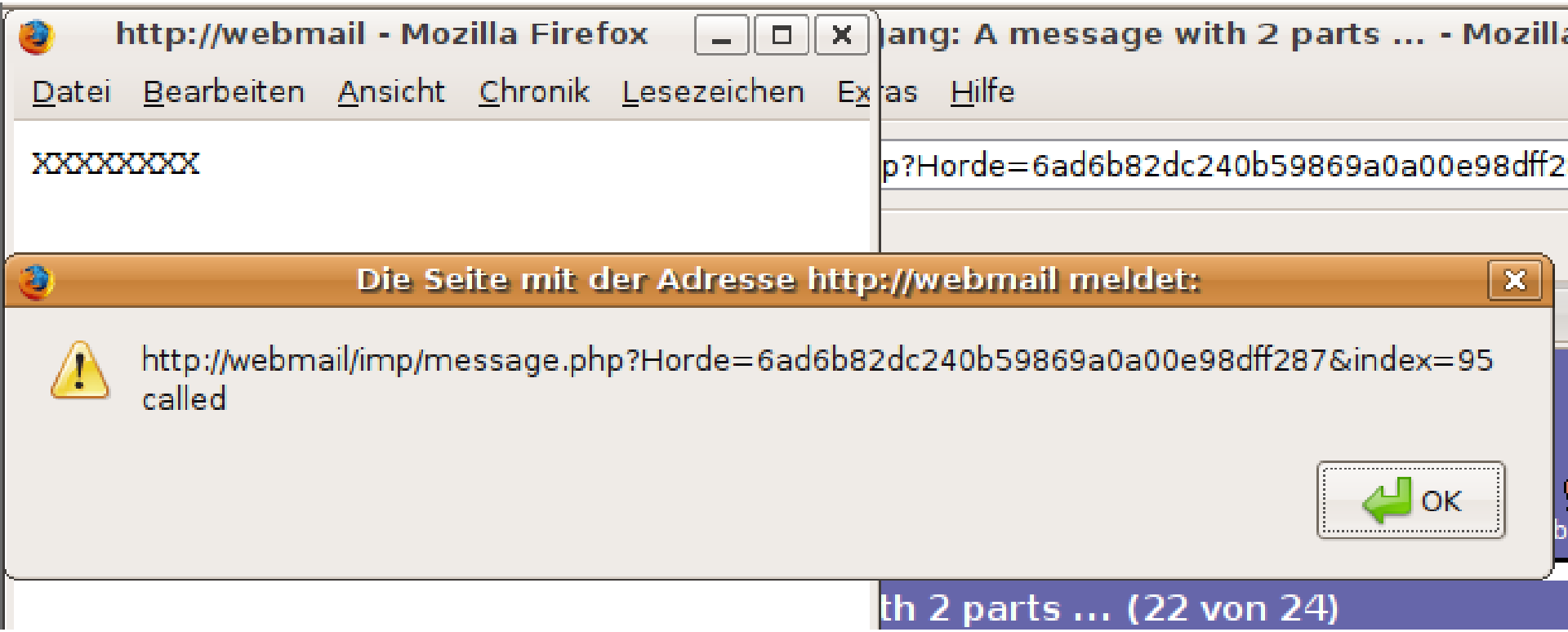
- Third-Party
  - Browser – Whitespace / linefeed characters



CVE-2008-3824

Beispiele – Technik

- Third-Party
  - Browser – Whitespace / linefeed characters



CVE-2008-3824

Beispiele – Technik

- Third-Party
  - Browser – HTML parsing

```
$ echo -en
"\x89\x50\x4E\x47\x0D\x0A\x00\x00\x00\0DPHCK\x00\x00\x00\x01
\x00\x00\x00\x01" >XSS.png
```

```
$ echo -n
'<html><body>foooooooo<script>alert("OOOoooOOoOOoOoOOOo
o")</script></body></html>' >> XSS.png
```

Beispiele – Technik

- Third-Party
  - Browser – HTML parsing

‰PNG DPHCK□□fooooooooo

Beispiele – Technik

- Third-Party
  - Browser / OS – Bestimmung des MIME Typs ?
    - **FindMimeFromData** enthält Abfragen für aktuell 26 separate MIME Typen
    - Gelesen werden die ersten 256 Bytes und dann abgefragt
    - Randnotiz von der MSDN:

      "Internet Explorer 8 and later. FindMimeFromData will not promote image types to "text/html" even if the data is lacking magic numbers (signature bytes)."

Agenda

- Einführung
- Problemdefinition
- Problemlösung
- XSS Filterung
- Umgehungsvarianten
- Beispiele
- **Fazit**

Fazit

- Sicherheit ist wie eine Zeitgeschichte
  - Alte Verwundbarkeiten in neuen Kleidern
- IPS/IDS Umgehnungstechniken sind auch auf Layer 7 anwendbar

- Empfehlung für XSS Filter Entwickler
  - 1ster Schritt: Kanonisierung der Daten (DOM, tidy)
  - Vertraue keinem Browser und den Erweiterungen !!!
  - Kontinuierliches Update der Blacklist

Fazit

Noch eine wichtige Anmerkung:

Selbst wenn Ihr XSS Filter das XSS Sheet Cheat ohne Fehler durchläuft, heißt es noch lange nicht, dass Ihr Filter “sicher” ist !

Vielen Dank für Ihre Aufmerksamkeit !

Alexios Fakos
alexios.fakos@nruns.com