Using Guided Missiles in Drive-bys

Automatic browser fingerprinting and exploitation with the Metasploit Framework

James Lee

whoami

- James Lee
- egypt
- Developer, Metasploit Project
- Co-Founder, Teardrop Security
- Member, Attack Research

The Metasploit Framework

- Created by HD Moore in 2003
  - ncurses based game
  - Later became a real exploit framework in perl
- Rewritten in ruby in 2005

My Involvement in MSF

- Started submitting patches and bug reports in 2007
- HD gave me commit access in April 2008
  - Broke the repo with my first commit

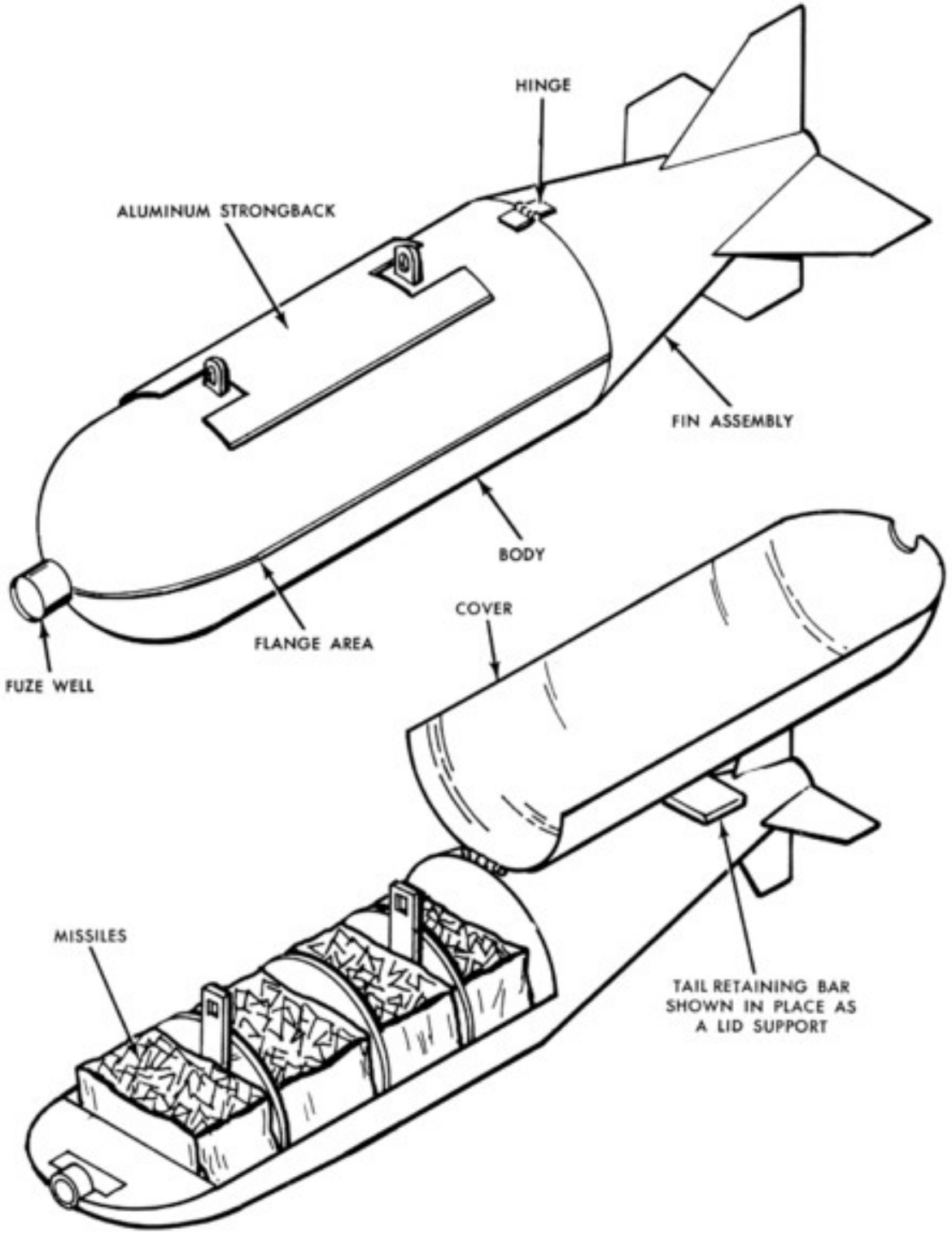
Why clientsides

- Karmetasploit
- Weakest link, blah, blah, blah
- See Chris Gates

client exploits in msf

- Extensive HTTP support
  - Heapspray in two lines of code
  - Sotirov's .NET DLL, heap feng shui
- Wide range of protocol-level IDS evasion
- Simple exploit in ~10 lines of code
- Or arbitrarily complex
- As of June 28, MSF has 85 browser exploit modules

Problem

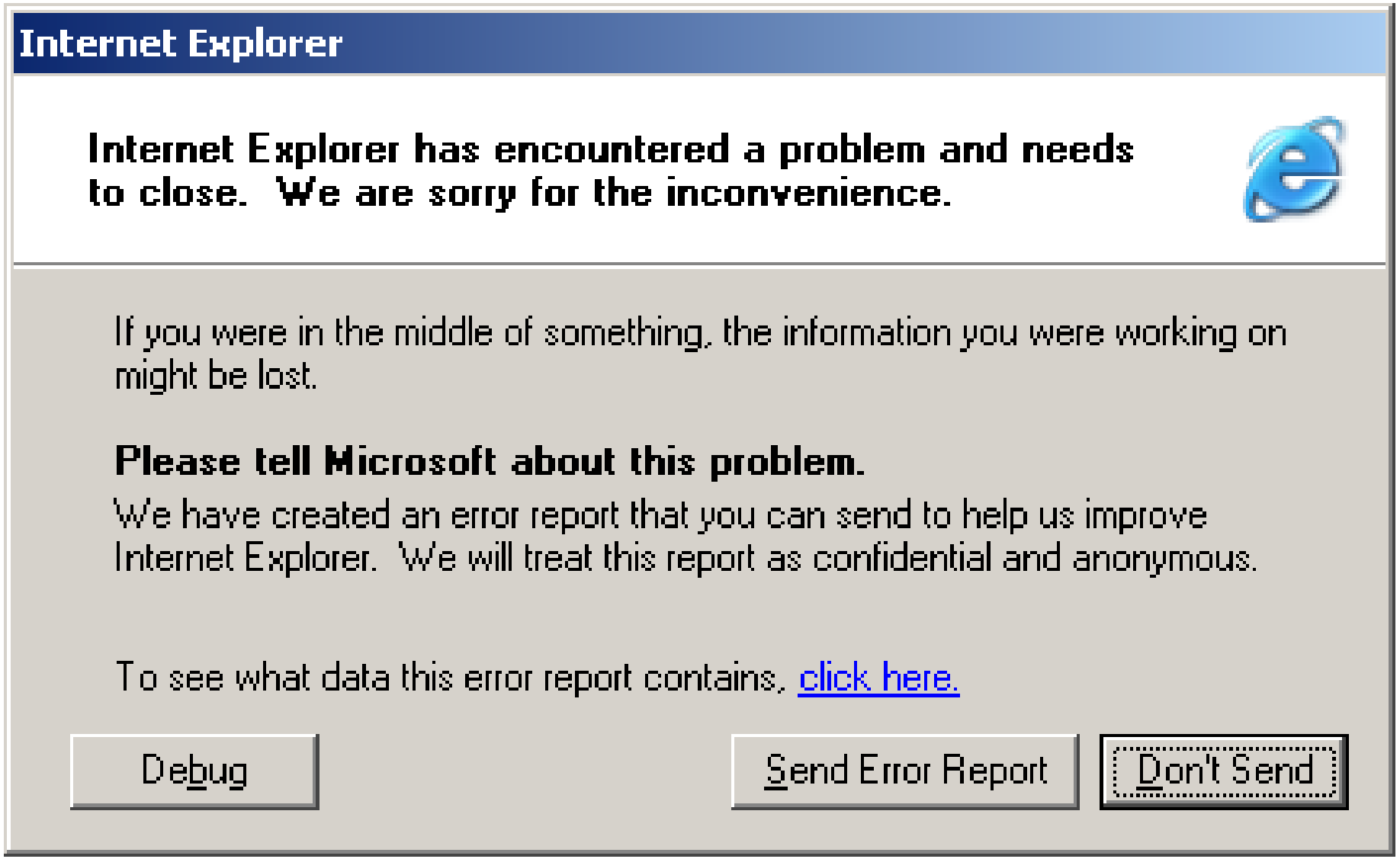

HINGE
ALUMINUM STRONGBACK
FIN ASSEMBLY
BODY
COVER
FLANGE AREA
FUZE WELL
MISSILES
TAIL RETAINING BAR
SHOWN IN PLACE AS
A LID SUPPORT

Solution

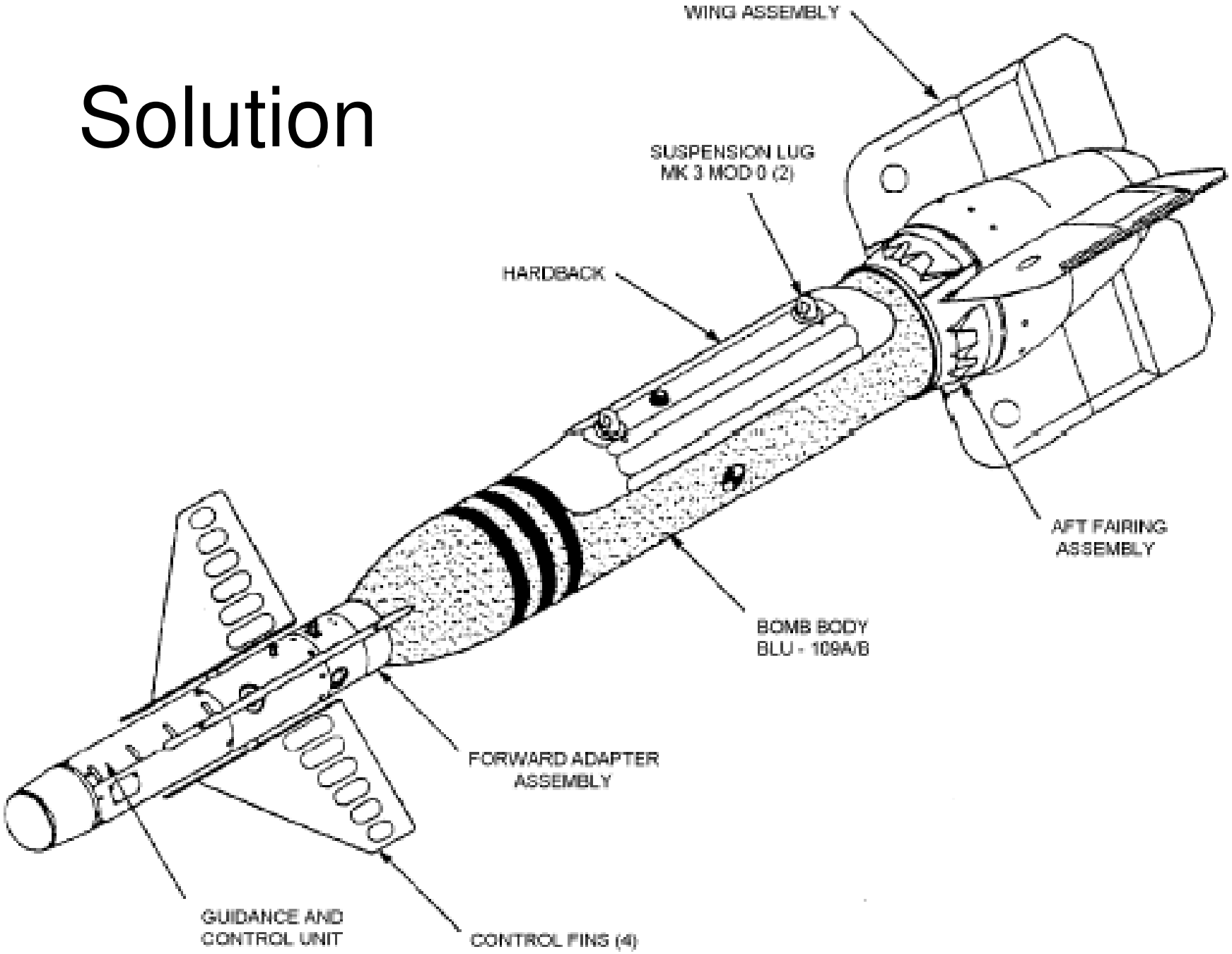
Cluster Bomb Approach

- Is it IE? Send all the IE sploits
- Is it FF? Send all the FF sploits
- Ad-hoc exploits
  - Pain in the ass when new sploits come out

Problem


Internet Explorer
Internet Explorer has encountered a problem and needs to close. We are sorry for the inconvenience.
If you were in the middle of something, the information you were working on might be lost.
Please tell Microsoft about this problem.
We have created an error report that you can send to help us improve Internet Explorer. We will treat this report as confidential and anonymous.
To see what data this error report contains, click here.
Debug
Send Error Report
Don't Send

Solution


WING ASSEMBLY
SUSPENSION LUG
MK 3 MOD 0 (2)
HARDBACK
AFT FAIRING
ASSEMBLY
BOMB BODY
BLU - 109A/B
FORWARD ADAPTER
ASSEMBLY
GUIDANCE AND
CONTROL UNIT
CONTROL FINS (4)

Guided Missile Approach

- Only send exploits likely to succeed
  - Browser is IE7? Don't send IE6 sploits, etc.
- Added better client and OS fingerprinting
  - less likely to crash or hang the browser
- Still ad-hoc, still a pain in the ass

Shiny New Hotness

- Fingerprinting is more complete
  - More on this shortly
- Sort exploits by reliability
- Exploits contain their own tests
- Javascript sends a report, stored in a DB

Fingerprinting the Client

- User agent
  - Easy to spoof
  - Easy to change in a proxy
  - A tiny bit harder to change in JS

Fingerprinting the Client

- Various JS objects only exist in one browser
  - window.opera, Array.every
- Some only exist in certain versions
  - window.createPopup, Array.every, window.Iterator
- Rendering differences and parser bugs
  - IE's conditional comments

Hybrid

- Existence of document.getElementsByClassName means FF 3.0
- If UA says IE6, go with FF 3.0
- If UA says FF 3.0.8, it's probably not lying, so use the more specific value

Fingerprinting the OS

- Useragent
- From the server side, that's about it
- What about client-side?

Internet Explorer

- ScriptEngine*Version()
- Almost unique across all combinations of client and OS
- Brought to my attention by Jerome Athias

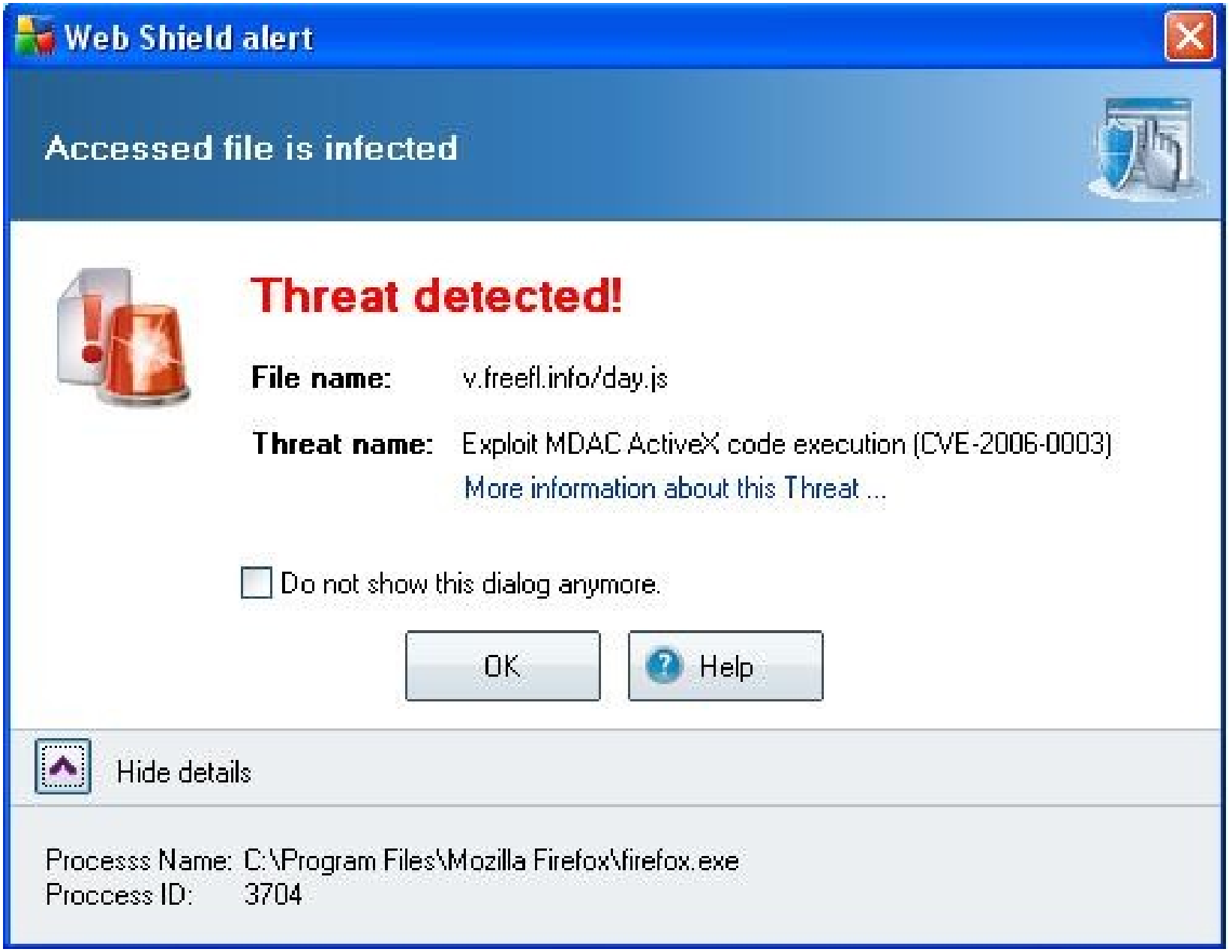
Opera

- window.opera.version()
  - Includes minor version, e.g. “9.01”
- window.opera.buildNumber()
  - Different on each platform for a given version
  - e.g.: “8501” == Windows

Others

- Really all we're left with is the User agent
- That's okay, most people don't lie
  - And those that do are likely to be patched anyway
- Generic, works everywhere that UA is not spoofed

Problem


Web Shield alert
Accessed file is infected
Threat detected!
File name: v.freefl.info/day.js
Threat name: Exploit MDAC ActiveX code execution (CVE-2006-0003)
More information about this Threat ...
Do not show this dialog anymore.
OK
Help
Hide details
Processs Name: C:\Program Files\Mozilla Firefox\firefox.exe
Proccess ID: 3704

Solution

- JS obfuscation
- Encryption?

Obfuscation

- Randomize identifiers
- Build strings from other things
- JSON / AJAX
- Obfuscation is not crypto

Writing Exploits

- Add autopwn_info() to top of exploit class
- :vuln_test should be some javascript to test for the vulnerability
  - Unless it's ActiveX
  - Usually comes directly from the exploit anyway

Example

- mozilla_navigatorjava

```
include Msf::Exploit::Remote::BrowserAutopwn
autopwn_info({
    :browser_name => HttpClients::FF,
    :javascript => true,
    :rank       => NormalRanking,#reliable memory corruption
    :vuln_test  => %Q|
        is_vuln = false;
        if (
           window.navigator.javaEnabled &&
           window.navigator.javaEnabled()
        ){
            is_vuln = true;
        }
        |,
})
```

Writing ActiveX Exploits

- IE doesn't seem to have a generic way to tell if an ActiveX object got created correctly
  - document.write(“<object ...>”) works sometimes
  - document.createElement(“object”) works sometimes
  - new ActiveXObject() only works if you have the class name, not the clsid

Solution

- typeof(obj.method)
  - 'undefined' if the object failed to initialize
  - 'unknown' or possibly a real type if it worked

Example

- ms06_067_keyframe

```
include Msf::Exploit::Remote::BrowserAutopwn
autopwn_info({
    :ua_name    => HttpClients::IE,
    :javascript => true,
    :os_name    => OperatingSystems::WINDOWS,
    :vuln_test  => 'KeyFrame',
    :classid    => 'DirectAnimation.PathControl',
    :rank       => NormalRanking #reliable memory corruption
})
```

Example

- winzip_fileview

```
include Msf::Exploit::Remote::BrowserAutopwn
autopwn_info({
    :ua_name    => HttpClients::IE,
    :javascript => true,
    :os_name    => OperatingSystems::WINDOWS,
    :vuln_test  => 'CreateFolderFromName',
    :classid    => '{A09AE68F-B14D-43ED-B713-BA413F034904}',
    :rank       => NormalRanking #reliable memory corruption
})
```

Commercial Comparison

- Firepack
- mpack
- Luckysploit

Mpack, Firepack

- Hard to acquire
- Old exploits
- Detection is only server-side
- Hard to change or update exploits
- Obfuscation + XOR

Luckysploit

- Real crypto (RSA, RC4)
- Harder to acquire

Browser Autopwn

- Easy to write new exploits or take out old ones
- Free (three-clause BSD license)
- Easy to acquire (http://metasploit.com)
- Not written in PHP
- OS and client detection is client-side, more reliable

Demonstrations

Thanks

- hdm, valsmith, tebo, mc, cg, Dean de Beer, pragmatk
- Everybody who helped with testing
- Whoever created ActiveX