Showcase Showdown
Browser Security Edition

Actionable Metrics for Web Browser Security

Shawn Moyer
Practice Manager

Ryan Smith
Chief Scientist

Hi, BlackHat.

- Quick overview of browser security research
  - Released in late 2011
  - Evaluated security of Internet Explorer 9, Chrome 12 & 13 , Firefox 5, on Windows 7 (32-bit)
- Collaborative effort by the entire Labs R&D team:
  - Drake, Mehta, Miller, Moyer, Smith, Valasek
- Some key points and a nickel tour.
- Paper, etc: http://www.accuvantlabs.com

We've come a long way...

- The browser is the most critical application we use today
  - In some cases it may be the only application we use
  - Especially true as we move to SaaS / cloud / etc
- Most common entry point for viruses, malware, client-side exploitation

No maps for these territories

- Metrics / bakeoffs thus far have been narrowband
  - Focused on some single, easy-to-measure test case
  - Bar charts are not the end goal of security "research"
- We took a more holistic view.
  - Defined shared attack surface on 3 major browsers
    - Specific focus on exploitation/persistence defense
  - Our goal was to create measurable, agnostic criteria
  - Public release of all test data and tool chains to foster an open dialogue

Browser Security Ecosystem

- We defined the browser security ecosystem as:
  - Browser Process Security Architecture
  - Add-On Security (Plugins, Extensions)
  - Exploit Mitigation and Sandboxing
  - Malware Detection / Blacklisting
  - Historical Vulnerability Metrics
- Again, our focus was on commonalities.

Process Security Architecture

- Common across all modern browsers:
  - Multi-process / multi-threaded architecture
  - Security barriers, trust zones, integrity models
- Integrity models in Windows 7:
  - System
  - High
  - Medium
  - Low

IE Process Architecture

- "Loosely Coupled" model
  - UI frame, tabs (low integrity) largely independent
- Medium integrity broker process
  - Creates low integrity tabs:
    - General Browsing and Rendering
    - ActiveX controls and other plugins
    - GPU acceleration
    - Tab-independent: downloads, toolbars, etc

iexplore.exe	3240	Medium	"C:\Program Files (x86)\
iexplore.exe	4588	Low	"C:\Program Files (x86)\
iexplore.exe	5580	Low	"C:\Program Files (x86)\
iexplore.exe	3960	Low	"C:\Program Files (x86)\

Chrome Process Architecture

- Uses a medium integrity broker process
  - Manages the UI
  - Creates separate low integrity processes for:
    - Rendering tabs
    - Out-of-process hosting for plugins, extensions
    - GPU acceleration
    - Named pipes created by broker for IPC

chrome.exe	1036	Medium	
chrome.exe	3052	Low	--type=extension --lan
chrome.exe	3848	Low	--type=plugin --plugin-
chrome.exe	5288	Low	--type=renderer --lang
chrome.exe	5680	Low	--type=gpu-process --

Firefox Process Architecture

- Single, medium integrity browser process
  - Contains entire browsing session in a single address space
    - All tabs
    - All add-ons
    - GPU acceleration
    - etc.
  - One exception: Flash and Silverlight plugins
    - Hosted out-of-process at medium integrity

firefox.exe	3912	Firefox	Medium
plugin-container.exe	6104	Plugin Container f...	Medium

Why Architecture Matters

- Process architecture determines if an exploit will
  - Succeed or fail
  - Attain persistence
  - Have access to other in-browser data
  - Communicate with other processes / plugins
- Along with sandboxing, key criteria for true exploitability

Process Name	Pid	Integrity Level	Limited Token	Description
chrome.exe	5880	Medium	No	Chrome Main Broker
chrome.exe	2072	Low	Yes	Chrome Renderer
chrome.exe	3956	Low	Yes	Sandboxed Flash plug-in
iexplore.exe	5732	Medium	No	IE UI Frame
iexplore.exe	4476	Low	No	IE Low Integrity Browser
firefox.exe	360	Medium	No	Firefox browser
plug-in-container.exe	3064	Medium	No	Plug-in container for Firefox

Sandboxing

- Why is sandboxing important?
  - There will always be bugs (until Skynet takes over)
  - Assume attackers will find a method for exploitation
  - Limit what damage can be done
- We've accepted compromise, hence emphasis on limitations post-mortem
- Ultimately if a sandbox bypass is required to land a payload, attacker complexity is increased

Sandboxing (cont.)

- General effectiveness of sandboxes

Sandbox Result	Chrome	Internet Explorer	Firefox
Read Files	✓	✗	✗
Write Files	✓	●	●
Read Registry Keys	✓	●	●
Write Registry Keys	✓	✓	●
Network Access	✓	✗	✗
Resource Monitoring	✓	●	●
Thread Access	✓	●	●
Process Access	✓	●	●
Process Creation	✓	✗	✗
Clipboard Access	✓	✓	✗
System Parameters	●	●	✗
Broadcast Messages	✓	✗	✗
Desktop & Windows Station Access	✓	✗	✗
Windows Hooks	✗*	✗	✗
Named Pipes Access	✓	●	✗

*Isolated Desktop and Window Station

✓ Action was blocked

● Action was partially blocked

✗ Action was allowed

Sandboxing (cont.)

- Google Chrome prevents processes in the sandbox from doing much of anything
  - Even if permission is granted, it is limited to the alternate desktop
- Microsoft Internet Explorer allows read access to most objects on the operating system
  - Deters a handful of system modifications
- Mozilla Firefox, on the other hand, is only limited by standard medium integrity
  - Permitting read, write and system change capabilities associated with regular, non-administrator users
  - If current user can do it, so can FF

JavaScript JIT Hardening

- JIT engines emit native code that can weaken security
- ASLR and DEP already exist for compiled binaries, but are not effective protections for JIT engines because
  - JIT compilation bridges the distinction between data and code
  - Predictable executable memory can turn a previously un-exploitable bug into a trivial exploit
- JIT hardening prevents the abuse of the JIT engine itself

JIT Hardening Comparison

Browser Comparison

JIT Hardening Techniques	Chrome	Internet Explorer	Firefox
Codebase Alignment Randomization	✗	✓	✗
Instruction Alignment Randomization	✗	✓	✗
Constant Folding	✓	✓	✗
Constant Blinding	✓	✓	✗
Resource Constraints	✓	✓	✗
Memory Page Protection	✗	✓	✗
Additional Randomization	✓	●	✗
Guard Pages	✓ *	●	✗

✓ Technique was implemented

● Technique was not necessary

✗ Technique was not implemented

* Chrome 14

ACCUVANT LABS

URL Blacklisting Services

- Intent: Early warning system for fast-flux malware
  - IE: MS Phishing filter -> MS URS / SmartScreen Filter
  - Google SBL, used by Chrome, FF, Safari
- Similar goals, some implementation differences
  - SBL: Sourced from crawl data, public submissions
  - MS URS: Numerous private feeds, public submissions
- We tested both services against public malware URL feeds
  - BLADE, MalwareBlacklist, MalwareDomains, MalwarePatrol
  - We wanted to use public, attributable sources

Blacklisting Services (cont.)

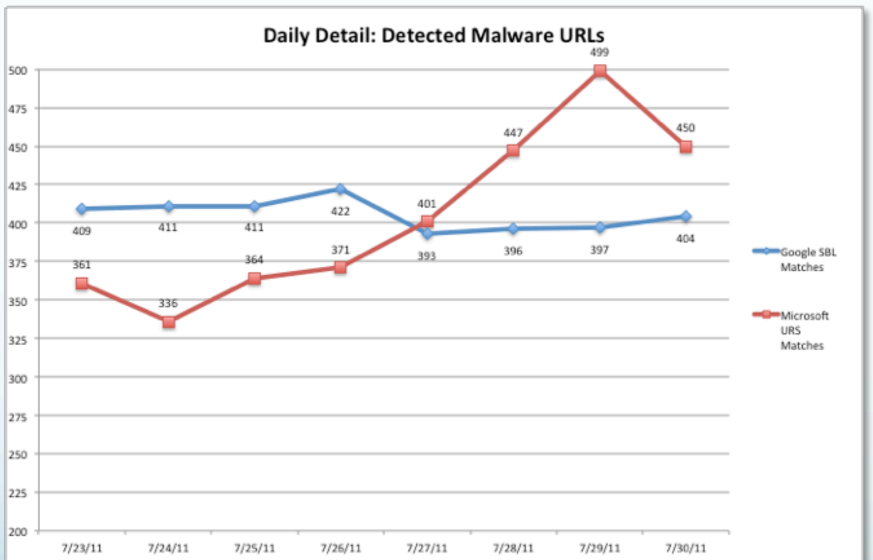
- 3086 average unique live URLs per day
  - 404 vs 405 matches for SBL vs URS
  - Interestingly, 42 SBL URLs also in URS
  - No URS URLs in SBS

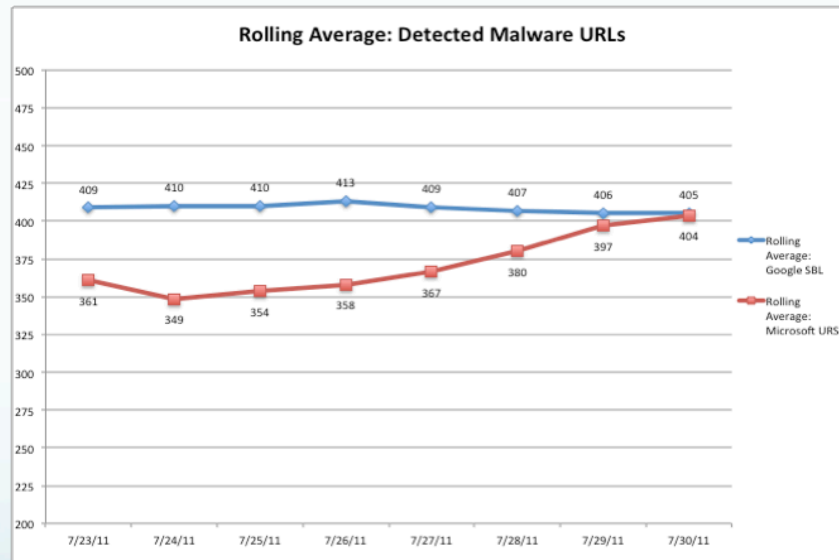
Date	7/23	7/24	7/25	7/26	7/27	7/28	7/29	7/30	Average
Google SBL Matches	409	411	411	422	393	396	397	404	405
Microsoft URS Matches	361	336	364	371	401	447	499	450	404
Total URLS	5684	5724	5738	6128	6145	6089	6149	6025	5960
Live URLS	2993	2948	3040	3416	3128	3043	3115	3003	3086

ACCUVANT LABS

Blacklisting Services (cont.)

- Both only ID a fraction of our sample set. What gives?
  - Apparently, malware SIGINT is really hard
  - Sharing info / collaboration could help
  - Still, it's clear neither of these services is a panacea



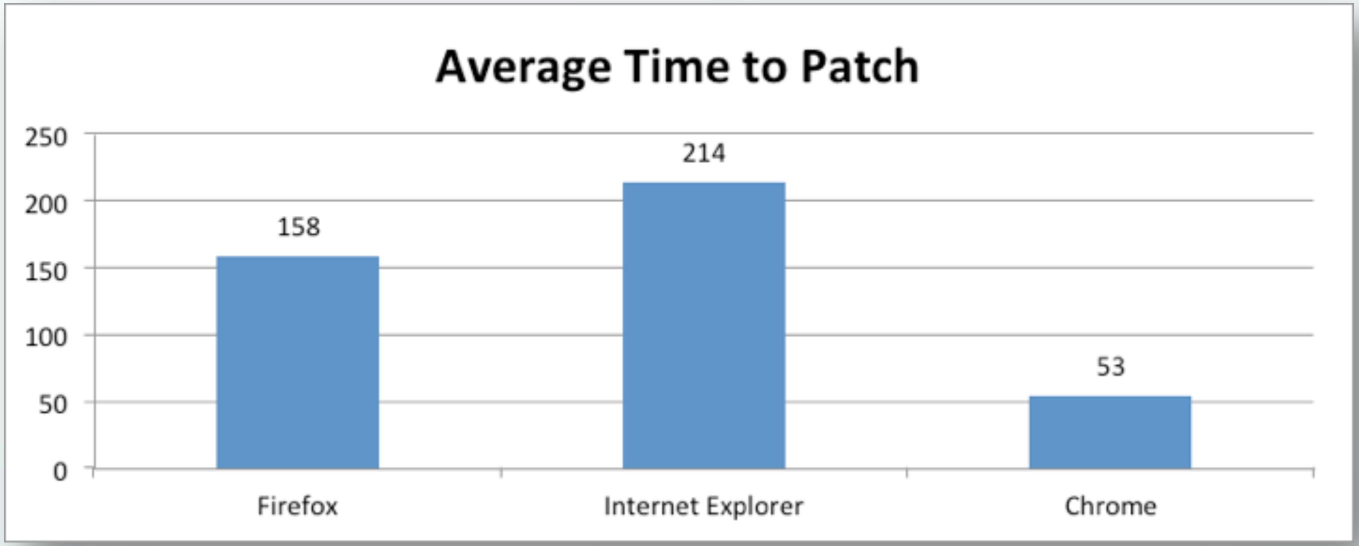


Vulnerability Statistics

- Difficult if not impossible to make clear comparisons here
  - Privately disclosed bugs, rollups, internal discoveries
  - Timelines and vagaries, severity metrics
  - We discarded what wasn't clearly measurable, normalized the data

Vuln Stats (cont.)

- One fairly reliable and interesting metric is time to patch
  - Again, based only on what we could normalize



ACCUVANT LABS

Conclusions

- Every browser has improved over the last 4 years
  - Diversity and the browser wars have benefited end users
- Most of the yardsticks are broken
  - Security models are hard to make charts from
- We believe, that the best defended browser is the most payload-hostile one

Conclusions (cont.)

- In the long run, no disinfectant like sunlight
  - Without transparency, there's no real debate on this topic
  - We shared our tools and data, anyone is welcome to debate the merit of our work, regardless of funding
- We're proud of the dialogue and conversation we created
  - We hope we've set a precedent in publishing our test data
  - Please expand our research! We might even help!

ACCUVANT LABS
Security Assessment and Research