Picture from deviantart.com

Turla - development & operations

THE BIG PICTURE

ANDRZEJ DERESZOWSKI

Agenda

- Introduction
- Part I – Development
- Part II – Operations
- How to protect yourself
- Attribution ?

A bit of history from my perspective

- 2008 – Agent.BTZ – threat that hit Pentagon
- 2009 – Some Agent.BTZ incidents here and there
- 2011 – me, tecamac and other researchers get together to analyze certain complex threat
- Beginning 2013 – we started distributing our report and help others handle infections
- Beginning 2014 – a series of discoveries started by G-Data and BAE Systems

What has been published so far ?

- ThreatExpert – "Agent.btz - A Threat That Hit Pentagon" – Nov 2008
- Trend Mirco - Windows XP/Server 2003 Zero-Day Payload Uses Multiple Anti-Analysis Techniques – Dec 2013
- G-DATA – "Uroburos – Highly complex espionage software with Russian roots" – Feb 2014
- BAE Systems – "Snake Campaign & Cyber Espionage Toolkit" – Mar 2014
- Deresz&tecamac – "Uroburos – The Snake Rootkit" – Mar 2014
- Sourcefire VRT – "Snake Campaign: A few words about the Uroburos Rootkit" – Apr 2014
- F-Secure – "Anatomy of Turla Exploits" – May 2014
- Kernelmode.info threads – Jun 2014
- CIRCL – "TR-25 Analysis - Turla / Pfinet / Snake/ Uroburos" – Jul 2014
- Symantec – "Turla: Spying tool targets governments and diplomats" – Aug 2014
- Kaspersky – "The Epic Turla Operation – Aug 2014

Many publications – many names

- Currently there is a lot of confusion in naming scheme
- Final stage:
  Agent.BTZ/Snake/Turla/Uroburos/Carbon/Pfinet/Snark/Sengoku
- Reconaissance stage:
  Epic/Tavdig/WipBot/WorldCupSec/TadjMakhal
- NOT all of them decribe the same « product »

PART I
Development

What is Turla ?

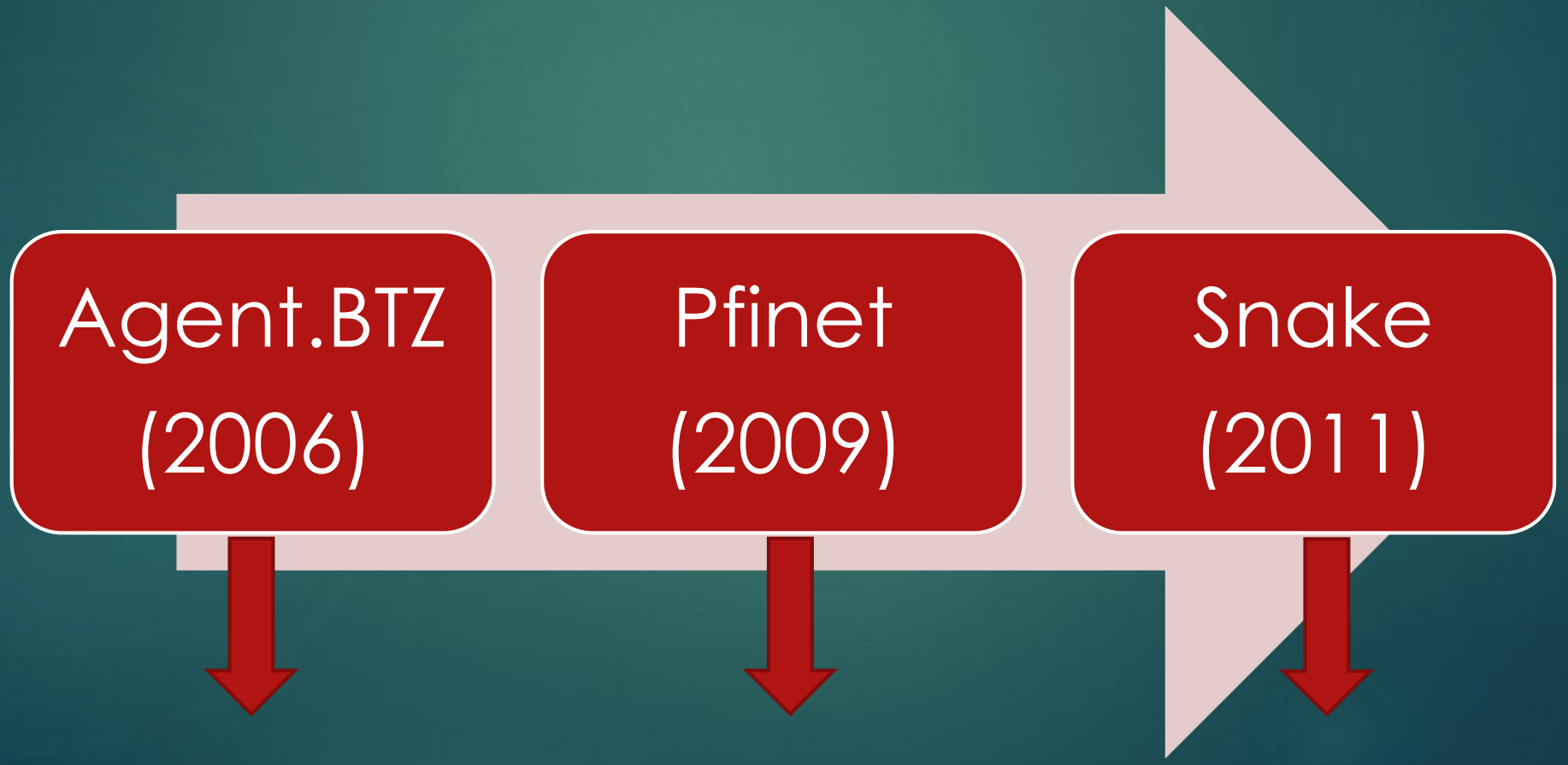
- Family of related sophisticated backdoor software
- Name comes from Microsoft detection signature – anagram of Ultra (Ultra3 was a name of the fake driver)
- All related by shared code

Code history

Agent.BTZ:
- Sengoku

Pfinet:
- Carbon
- Usermorde-centric Snake

Snake:
- Urouros
- Snark
- Kernelmode-centric Snake



Features: summary

Feature	Agent.BTZ	Pfinet	Snake
Storage	Hidden folder	Encrypted VFS	Encrypted VFS
Configuration	Hardcoded	Text file	Key-value store (queue)
Networking	Separate exes	Userland payloads	Kernel+userland
Incomming transports	No	No	Yes

VirtualBox exploit to load the driver

Pfinet
Snake

Uses a vulnerability in old (yet still signed!) VirtualBox driver to load its own driver

Source: F-Secure

VirtualBox driver EoP vulnerability – disabling driver signature enforcement

Turla also targets the *Oracle VirtualBox* software for exploitation. The EoP vulnerability Turla exploits only exists on *VirtualBox* versions 1.6.2 and 1.6.0, and was first disclosed by *CoreSecurity* in 2008; the vendor patched the vulnerability within a month [6].

Turla takes advantage of a vulnerable *VirtualBox* device driver (VBoxDrv.sys) in order to bypass a very important *Windows* security feature called Driver Signature Enforcement (DSE), which was first introduced in *Windows Vista*. Starting with the 64-bit version of Windows Vista, the driver code signing policy for the *Windows* OS requires all driver code to have a digital signature, to increase the platform's safety and stability [7]. This means

r device drivers if they want to load their malicious driver code on a
ıre, they must get rid of DSE in order for their malicious products

by *innotek*. Turla's author discovered an interesting way to utilize
uld then allow Turla's own unsigned rootkit driver to be run.
a five-step exploitation process.

proof-of-concept code presented by *CoreSecurity* [6] against this
that the exploit sample attempts to get rid of DSE and then
ails of the exploit sample in the next

Source: Sourcefire VRT

The VBoxdrv module

The DLL starts by calling 2 fu
"*sbhub.sys*. Once the Virtual
"VirtualBox exploit") is extra

Source: kernelmode.info

DSEFix - Defeating x64 Driver Signature Enforcement

POSTREPLY | Search this topic... | Search

DSEFix - Defeating x64 Driver Signature Enforcement

by **EP_X0FF** » Sun Jun 08, 2014 8:50 am

What is Driver Signature Enforcement? It is a security feature added to the NT6 which main purpose is to disallow loading drivers without digital signing, see http://msdn.microsoft.com/en-us/library/windows/hardware/dn653559(v=vs.85).aspx for more info. In reality this is yet another marketing bullshit from MS which ruined many freeware programs, and didn't fixed anything in antimalware field - if malware authors really want to load their driver - they will do this. Mainstream crapware like ssdt hooking trash were dying even without this "improvements"

Udis86: on-the-fly manipulation of dissassembled code in live kernel

Agent.BTZ
Pfinet
Snake

Source: deresz & tecamac

1.6 Force kernel mode

The rootkit will interact between user-mode and kernel-mode. Such interaction is delicate; in particular several system calls behave slightly differently when called from each mode. On such calls, the system traps the caller and uses the routine *ExGetPreviousMode* to determine whether the parameters are from a user-mode or kernel-mode source.

For example on Windows 2003, the code of *ExGetPreviousMode* is quite simple, it just gets and returns the PreviousMode value

```
.text:0044086C          mov     eax, large fs:124h
.text:00440872          mov     al, [eax+0D7h]
.text:00440878          retn
```

The tricky part is that the code may change across Windows versions. The current rootkit implements an easy solution for this issue: it disassembles *ExGetPreviousMode*, copies each instruction to the generated directive until the last return, and replaces `mov al, [eax+0D7h]` by `mov [eax+0D7h] MODE` where the D7 is dynamically computed according to the disassembly process and MODE is the requested mode (see Figure 1).

The disassembly is achieved with the support of *Udis86*[4]

Hooking engine – Udis86 reused

Agent.BTZ
Pfinet
Snake

Source: deresz & tecamac

2.1 Hooking engine

The hooking engine is very pragmatic, it relies on a custom interrupt (C3h). When a location is to be hooked, the instructions that cover the first four bytes are relocated to a handler structure. A callback is also set in this structure and the structure is inserted in a table *handler_table (9A75C)* where it is associated to an ID.

The target location is hooked replacing the first four following bytes with `push h; int C3h` where `h` is a handler identifier.

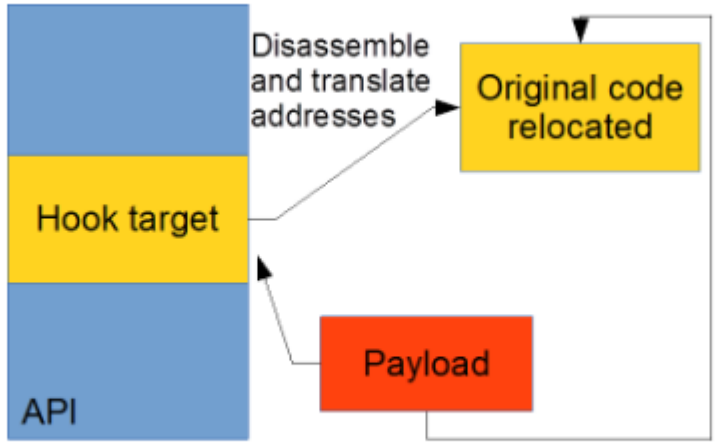
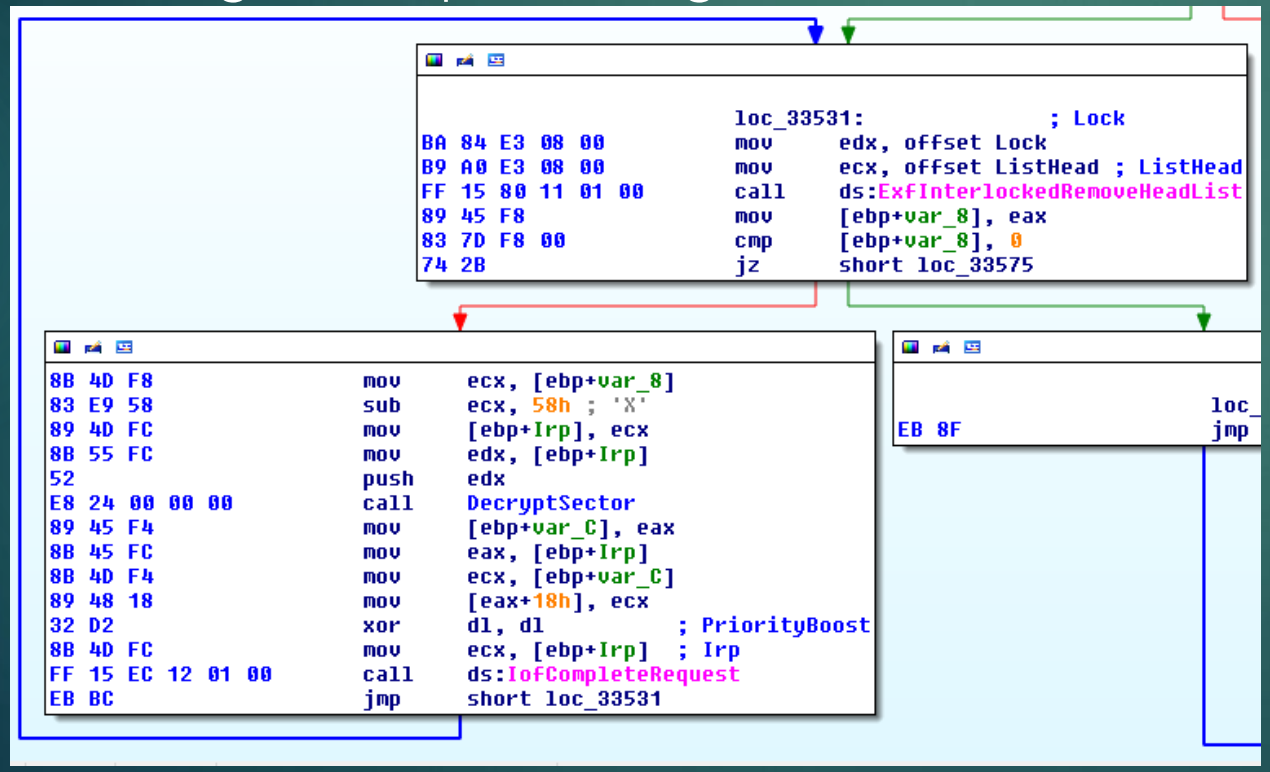


Figure 1: Snake advanced assembly manipulation when relocating the code.

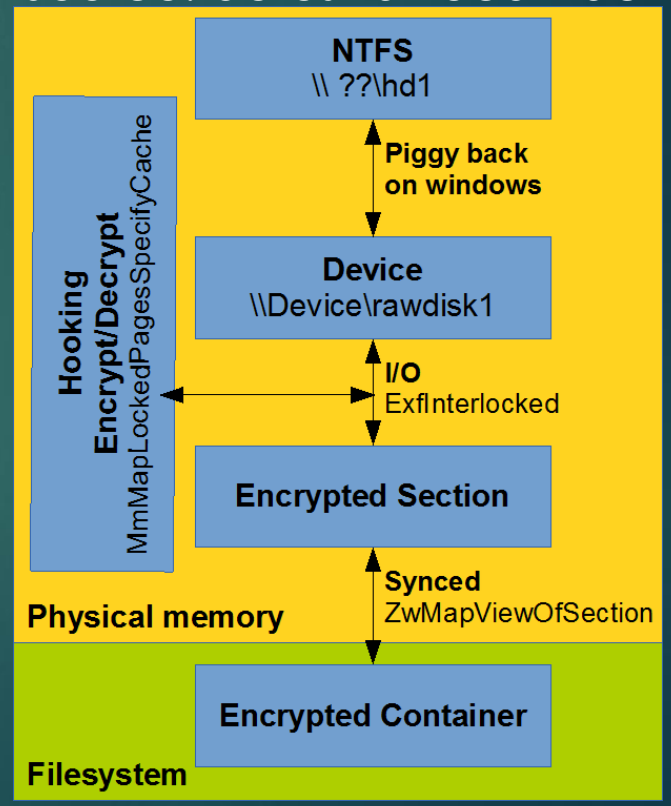
Encrypted VFS

Pfinet
Snake

- Implemented in Carbon and Snake
- CAST 128 encryption used
- Decryption/encryption implemented on low level by hooking sector processing mechanisms:



Source: deresz & tecamac



Encryped VFS

Pfinet
Snake

Two volumes: permanent (mapped to a file on a real file system) and volatile storage

Encrypted container located in `%windows%\$NtUninstallQ817473$\hotfix.dat`

Source: deresz & tecamac

```
C:\Documents and Settings\Administrator>dir \\.\Hd1\\
 Volume in drive \\.\Hd1 has no label.
 Volume Serial Number is 0000-1D1E

 Directory of \\.\Hd1

[output redacted]
10/18/2011  06:23 AM                 0 dump
05/04/2013  06:10 AM             8,334 klog
10/18/2011  03:50 AM           294,912 pscp.exe
10/21/2011  08:17 AM         1,089,536 queue
10/06/2011  07:33 AM         1,089,536 queue.sav
10/20/2011  05:29 AM           275,968 rar.exe
              16 File(s)      3,475,618 bytes
               0 Dir(s)               0 bytes free

C:\Documents and Settings\Administrator>
```

```
C:\Documents and Settings\Administrator>dir \\.\Hd2\\
 Volume in drive \\.\Hd2 has no label.
 Volume Serial Number is BA9B-99E8

 Directory of \\.\Hd2

File Not Found
```

Configuration mechanism

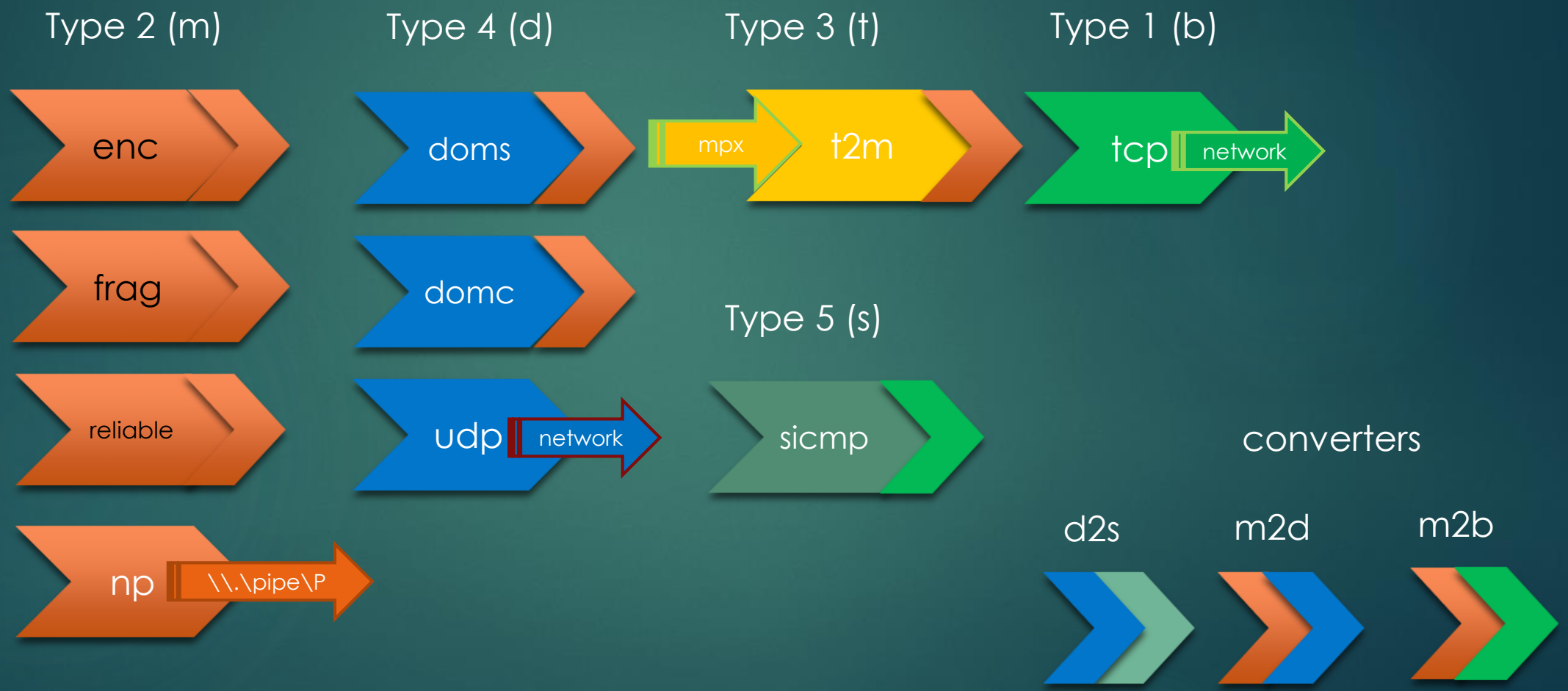
- Agent.BTZ:
  - Config hardcoded in the user mode executables
- Pfinet:
  - Configuration file stored on the VFS in a flat file: *config.txt*
  - Transports implemented in user mode
  - Usermode payloads hardcoded in the rootkit body:
    - cryptoapi.dll
    - inetpub.dll

Configuration mechanism

- SNAKE:
  - Uses « queue » file that contains configuration parameters in the form of key/value pairs
  - « queue » file located on the VFS
  - Queue contains:
    - Transports configuration
    - Userland payloads:
      - inj_snake_Win32.dll – a counterpart of a rootkit for userland
      - inj_services_Win32.dll
      - rkctl_Win32.dll

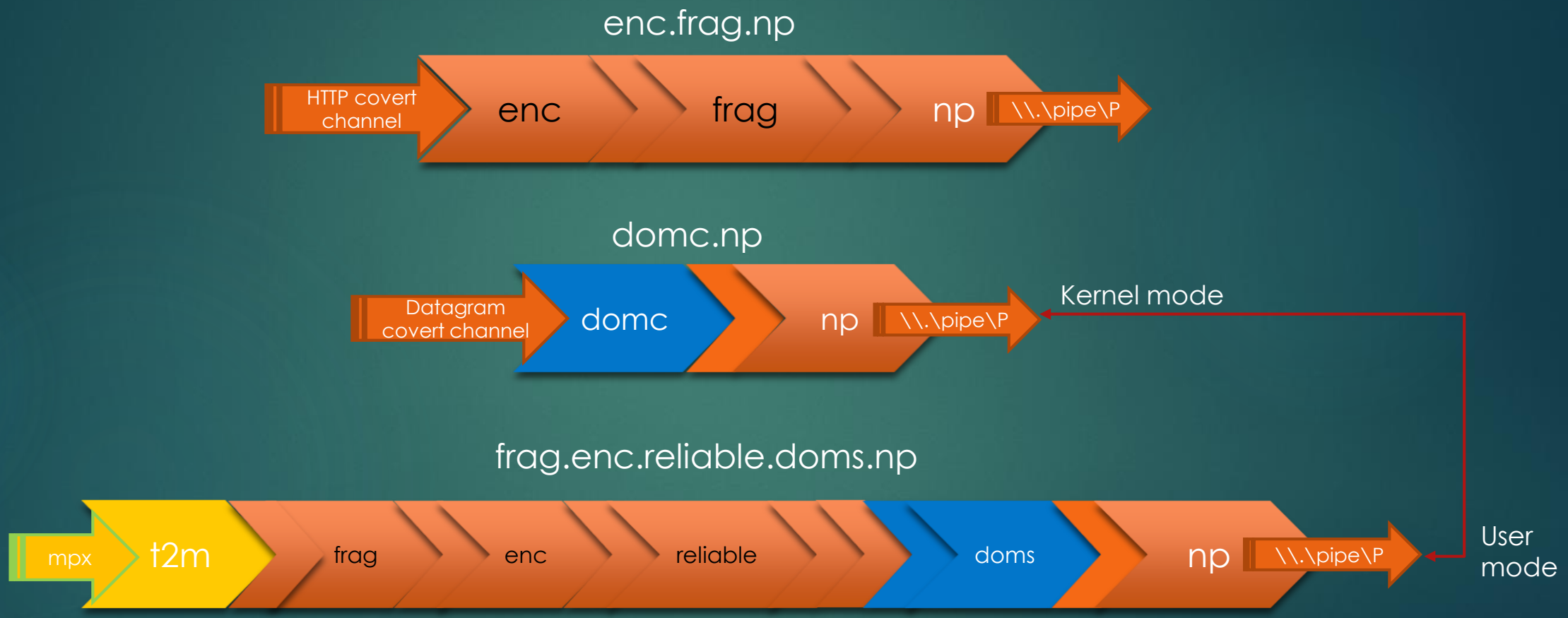
Modular transports – channel elements

Snake



Modular transports – combined together

Snake



Protocols to choose

- Datagram covert channels:
  - Raw layer 2 (Ethernet type 0x7FF)
  - Raw ICMP
  - Raw UDP - DNS
  - Raw IP

- Stream covert channels and activation triggers:
  - Raw TCP
  - HTTP: URL parameters of an HTTP request
  - HTTP: Hidden in HTTP headers
  - HTTP: Hidden in local part of the URL
  - SMTP: triggered by a recipient e-mail address

Examples of *incoming transports* – covert channels

SMTP covert channel – rootkit resides on the mail server of pwned-prg.com

```
HELO whatever.com
250 Hello whatever.com, I am glad to meet you
MAIL FROM: <you.bet@you.are.not>
250 OK
RCPT TO: <trueburger@pwned-org.com>
250 OK
354 End data with <CR><LF>.<CR><LF>
<commands>
.
```

Recipient user name must be 10 characters
Last two characters (in red) are the checksum calculated on the first 8:

```
username[9] == sum / 26 + 65
username[10] == 122 - sum % 26
```

Examples of *incoming transports*

HTTP covert channel – rootkit resides on the web server of pwned-prg.com

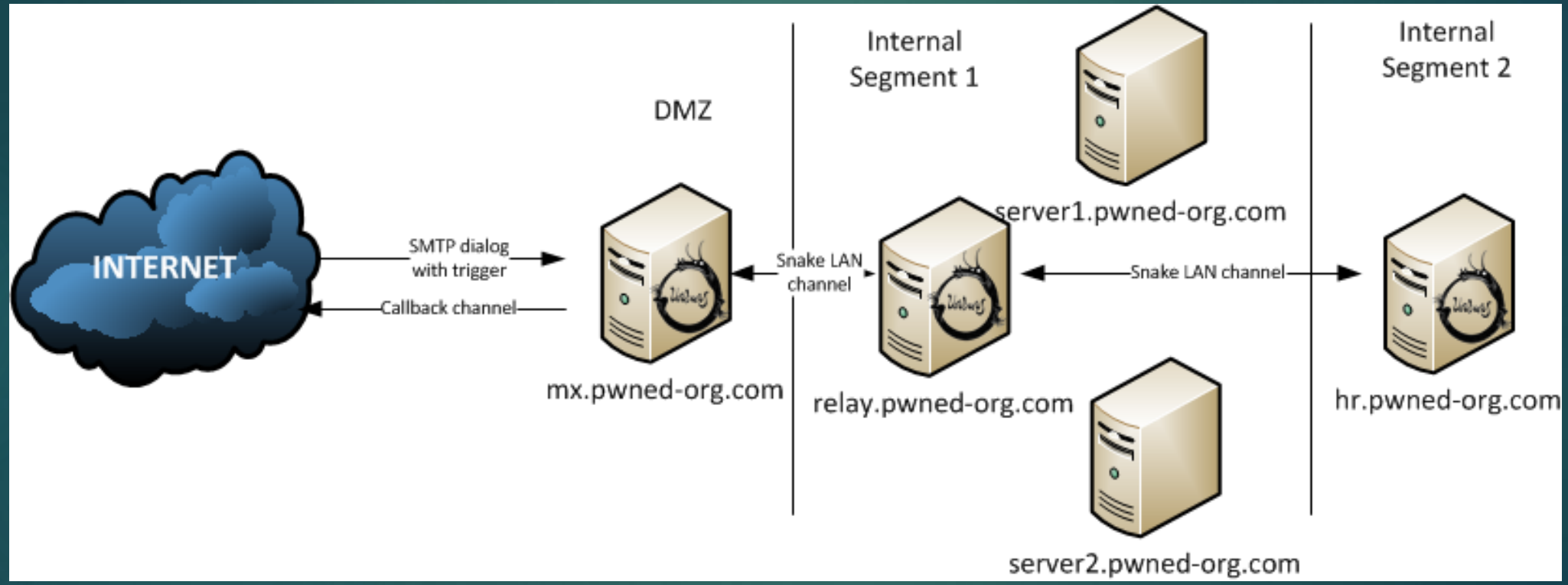
```
GET / HTTP/1.1
SomeHeader: trueburgerYmFzZTY0ZW5jb2R1ZCBzdHJpbmcKYmFz
…
```

- Same signature calculated on the first 10 bytes of the header value
- Base 64 content that follows is decoded and XOR-ed back with raw buffer starting at offset 0
- First four bytes of the resulting content is a magic value, by default set to 0xDEADBEEF but changed by the initialization queue

SNORT signatures – difficult to create !

Possible to create Surricata sigs with the use of LUA

Big picture view of compromised network



Developers

- Vlad, gilg, urik
- Version control info present in some of the samples:

$Id: snake_config.c 5204 2007-01-04 10:28:19Z vlad $
$Id: mime64.c 12892 2010-06-24 14:31:59Z vlad $
$Id: event.c 14097 2010-11-01 14:46:27Z gilg $
$Id: named_mutex.c 15594 2011-03-18 08:04:09Z gilg $
$Id: nt.c 20719 2012-12-05 12:31:20Z gilg $
$Id: ntsystem.c 19662 2012-07-09 13:17:17Z gilg $
$Id: rw_lock.c 14516 2010-11-29 12:27:33Z gilg $
$Id: rk_bpf.c 14518 2010-11-29 12:28:30Z gilg $
$Id: t_status.c 14478 2010-11-27 12:41:22Z gilg $
$Id: l1_check.c 4477 2006-08-28 15:58:21Z vlad $
$Id: m2_to_b2_stub.c 4477 2006-08-28 15:58:21Z vlad $
$Id: m_frag.c 8715 2007-11-29 16:04:46Z urik $

Who are they ?

phpBB creating communities

KernelMode.info
A forum for kernel-mode exploration.

Search... Search
Advanced searc

Board index ‹ Forums ‹ Tools/Software

FAQ Register Logi

DSEFix - Defeating x64 Driver Signature Enforcement

POSTREPLY Search this topic... Search

18 posts • Page 1 of 2 • 1

DSEFix - Defeating x64 Driver Signature Enforcement

by **EP_X0FF** » Sun Jun 08, 2014 8:50 am

We are so happy that most of "rootkit" code inside Turla was inspired by our program and features (this level of awareness is never seen anywhere in ITW malware since Rustock), so we decided to create something inspired by Turla in sort of exchange.

This proclaimed to be goverment sponsored lolkit in a reality is just a result, a compilation of several freelancers work (from both UA and RU) to create and support toolkit they sell for various kinds of espionage. For idiots from BAE Systems who are painting fake malware distribution diagrams in the Excel - No KGB or Kremlin here, guys, take a pill and relax with your prepaid propaganda.

EP_X0FF
Global Moderator

Posts: 3879
Joined: Sun Mar 07, 2010 5:35 am
Location: Russian Federation
Reputation point: 465

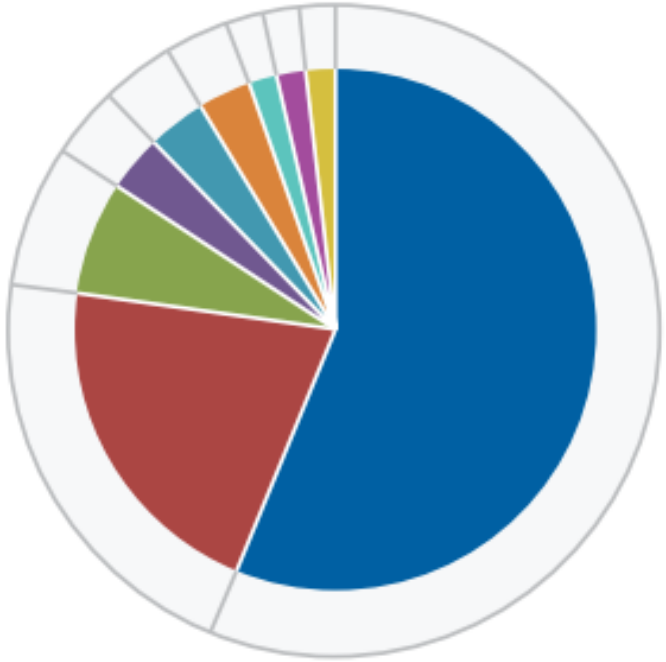
PART II
Operations

Hmm, which group are we talking about ?

- Not sure we can speak about one « Turla group »
- Turla is just one of the tools « Turla group(s) » uses
- There is however a lot of common things …
- While the tool itself is quite impressive, operators that are using it are sloppy …

Countries of interest

Source: BAE Systems



#Samples	Submission Year					
Source country	2010	2011	2012	2013	2014	Total
Ukraine	1	3	6	8	14	32
Lithuania				9	2	11
Great Britain				4		4
Belgium				2		2
Georgia					2	2
United States		1	1			2
Romania				1		1
Hungary					1	1
Italy					1	1
Total	1	4	7	24	20	56

Publicly known victims

Pentagon computer networks attacked

The cyber-strike on key sites is thought to be from inside Russia.

November 28, 2008 | Julian E. Barnes | Barnes is a writer in our Washington bureau.

Finland admits it's suffered a massive cyber-attack. Is the same thing happening across Europe?

Sweden's National Defense Radio Establishment – its version of the U.S. National Security Agency signals-intelligence agency – told Reuters it had detected a number of attacks by Turla/Snake /Uroburos; officials in Finland also acknowledged having been attacked, but didn't confirm whether the culprit was related to Turla or Agent.BTZ. None of the investigations have turned up proof the malware is Russian, or that it is connected to official Russian intelligence services.

Print this article

Share 141

Uroburos rootkit: Belgian Foreign Ministry stricken

First information published on spy attack involving a high profile victim

Belgique : un ministère ciblé par un piratage informatique en lien avec l'Ukraine

Des dossiers et documents liés à la crise ukrainienne ont été piratés au sein du ministère belge des Affaires étrangères, ces derniers jours.

Turla: Staged operation

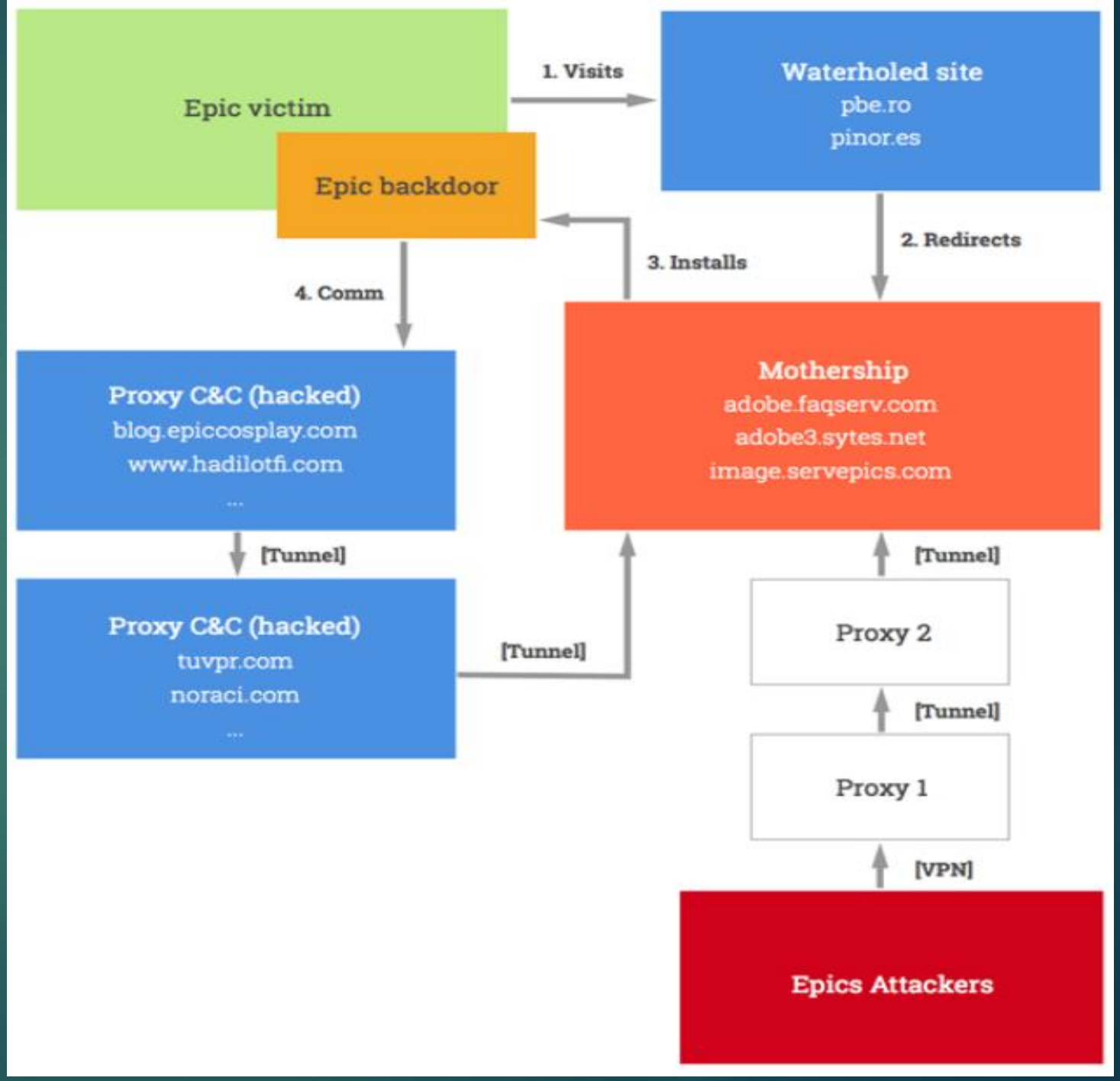
- Stage 0 – attack stage - infection vector
- Stage 1 – reconaissance stage - initial backdoor
- Stage 2 – lateral movements
- Stage 3 – « access established » stage – **TURLA deployed**
- On each stage they can quit if it turns out that the « non-interesting » victim has been encountered

Stage 0: infection vector

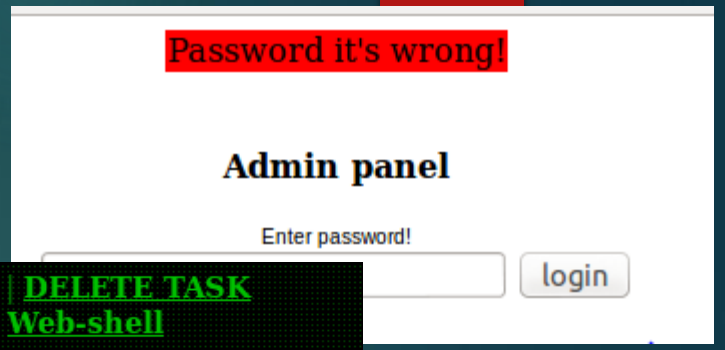
- Traditional infection vector – spear phishing: exploits (CVE-2013-3346 + CVE-2013-5065)
- Watering holes (strategic web compromise)
  - "Adobe update" social engineering trick
  - Java exploits (CVE-2012-1723), Adobe Flash exploits (unknown) or Internet Explorer 6, 7, 8 exploits (unknown)
- Third party suppliers compromise
- No use of 0-day exploits (almost no)

Stage 0: Watering holes mechanism

Source: Kaspersky Lab



Stage 0: Watering hole panel



Stats | View Rule | View Exception | Black List | Clear Log+Exception | DELETE TASK
| TASK EDITOR | CONFIG EDITOR | DELETE successful count | Web-shell

Down

Total - 0

Interesting IP - 0

IE 6.0	IE 7.0	IE 8.0	Opera	Firefox	Safari	Chrome	Unknown

ID	Date	IP	Mode	OS	Client	Country	Referer	User-agent	Lang	Cl IP	Page	JavaScript	Adobe Shockwave	Adobe Flash	Adobe Reader	Win Media Player	Quick Time	MS Word	Java

Total - 0

Interesting IP - 0

Current Time: 2014-09-24 11:42:53

Up

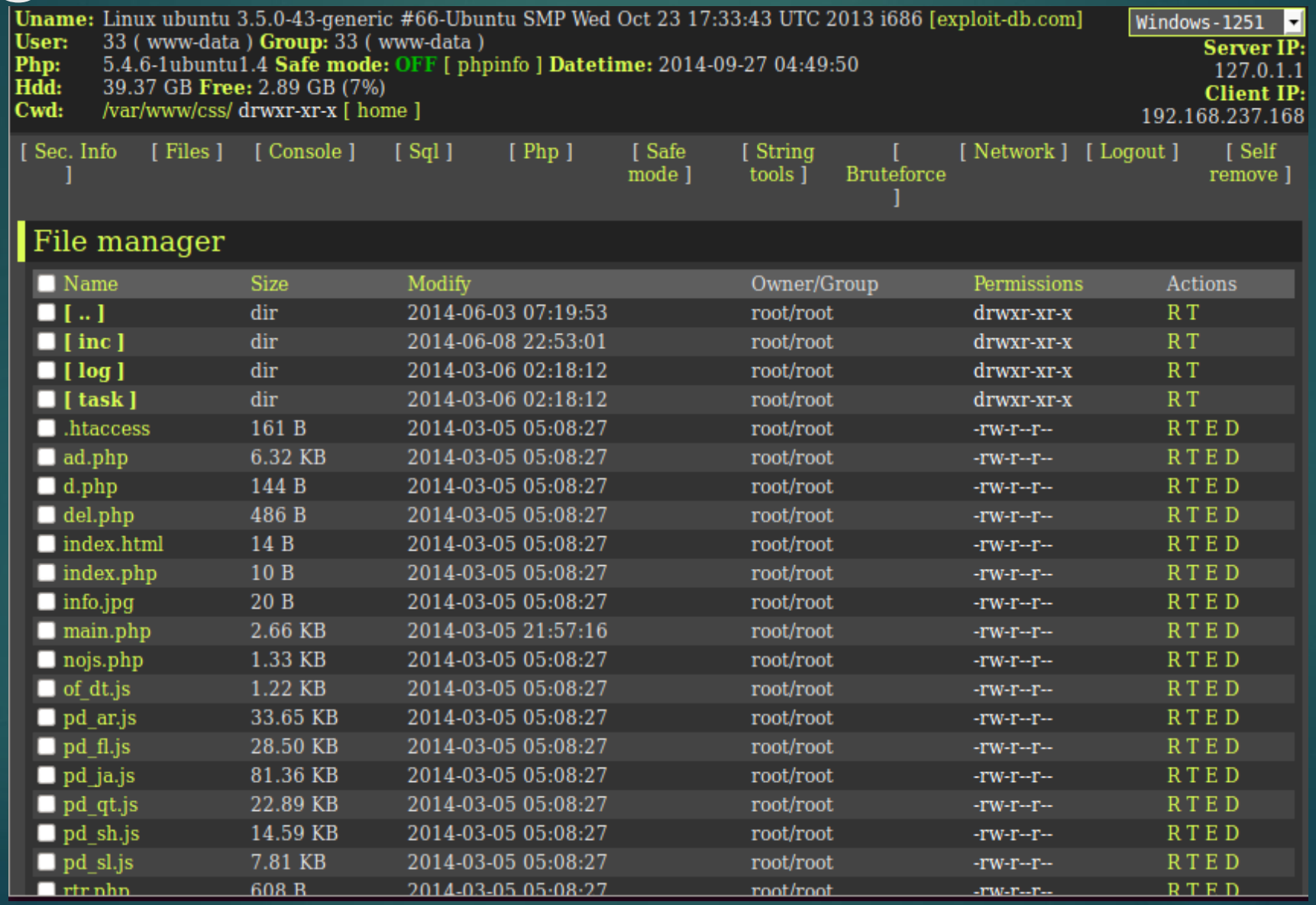
{*} File is NOT writable -> /var/www/css/log/maxsuc.css
{*} File is NOT writable -> /var/www/css/task/task.css
{*} File is NOT writable -> /var/www/css/task/time.css
{*} File exists -> /var/www/css/task/deny.css
{*} Check time -> 180

**********TASK**********

{*} Total - 0

****

Stage 0: Web shell



Uname: Linux ubuntu 3.5.0-43-generic #66-Ubuntu SMP Wed Oct 23 17:33:43 UTC 2013 i686 [exploit-db.com]
User: 33 (www-data) **Group:** 33 (www-data)
Php: 5.4.6-1ubuntu1.4 **Safe mode:** OFF [phpinfo] **Datetime:** 2014-09-27 04:49:50
Hdd: 39.37 GB **Free:** 2.89 GB (7%)
Cwd: /var/www/css/ drwxr-xr-x [home]

Windows-1251

Server IP: 127.0.1.1
Client IP: 192.168.237.168

[Sec. Info] [Files] [Console] [Sql] [Php] [Safe mode] [String tools] [Bruteforce] [Network] [Logout] [Self remove]

File manager

Name	Size	Modify	Owner/Group	Permissions	Actions
[..]	dir	2014-06-03 07:19:53	root/root	drwxr-xr-x	R T
[inc]	dir	2014-06-08 22:53:01	root/root	drwxr-xr-x	R T
[log]	dir	2014-03-06 02:18:12	root/root	drwxr-xr-x	R T
[task]	dir	2014-03-06 02:18:12	root/root	drwxr-xr-x	R T
.htaccess	161 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
ad.php	6.32 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
d.php	144 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
del.php	486 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
index.html	14 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
index.php	10 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
info.jpg	20 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
main.php	2.66 KB	2014-03-05 21:57:16	root/root	-rw-r--r--	R T E D
nojs.php	1.33 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
of_dt.js	1.22 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
pd_ar.js	33.65 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
pd_fl.js	28.50 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
pd_ja.js	81.36 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
pd_qt.js	22.89 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
pd_sh.js	14.59 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
pd_sl.js	7.81 KB	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D
rtr.php	608 B	2014-03-05 05:08:27	root/root	-rw-r--r--	R T E D

Stage 1: reconaissance stage

- Initial backdoor dropped – WipBot/Epic/TavDig
- Simple backdoor with a handful of commands
- Has no code in common with any variant of Turla but exports functions with the same names: ModuleStart and ModuleStop
- Well desribed in Kaspersky Lab report:

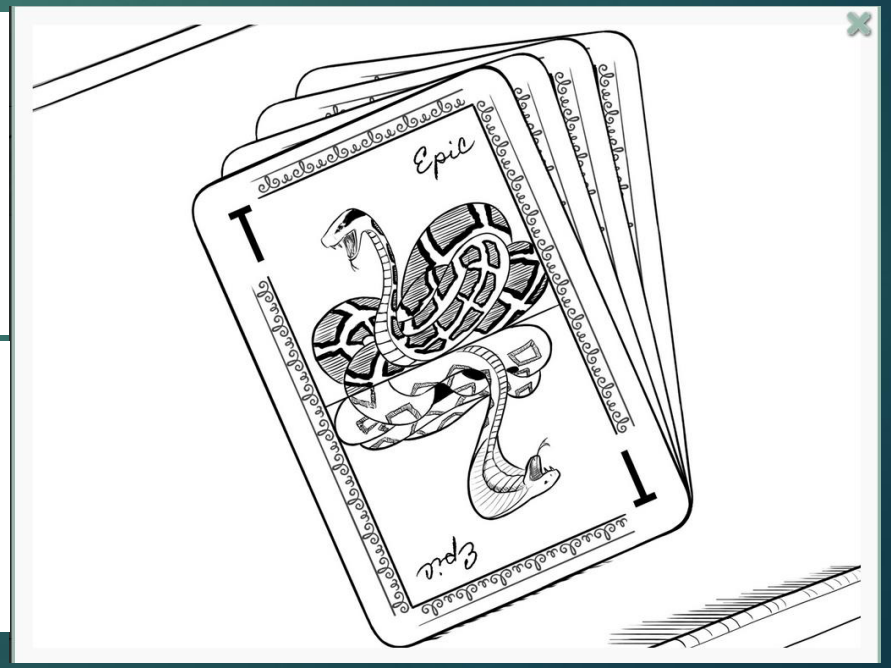
The Epic Turla Operation

Solving some of the mysteries of Snake/Uroburos

By GReAT on August 7, 2014. 1:55 pm

Executive Summary

Over the last 10 months, Kaspersky Lab researchers have analyzed a massive cyber-espionage operation which we call "Epic Turla". The attackers behind Epic Turla have infected several hundred computers in more than 45 countries, including government institutions, embassies, military, education, research and pharmaceutical companies.



Stage 1: Some interesting tricks used in WipBot

Source: Trend Micro

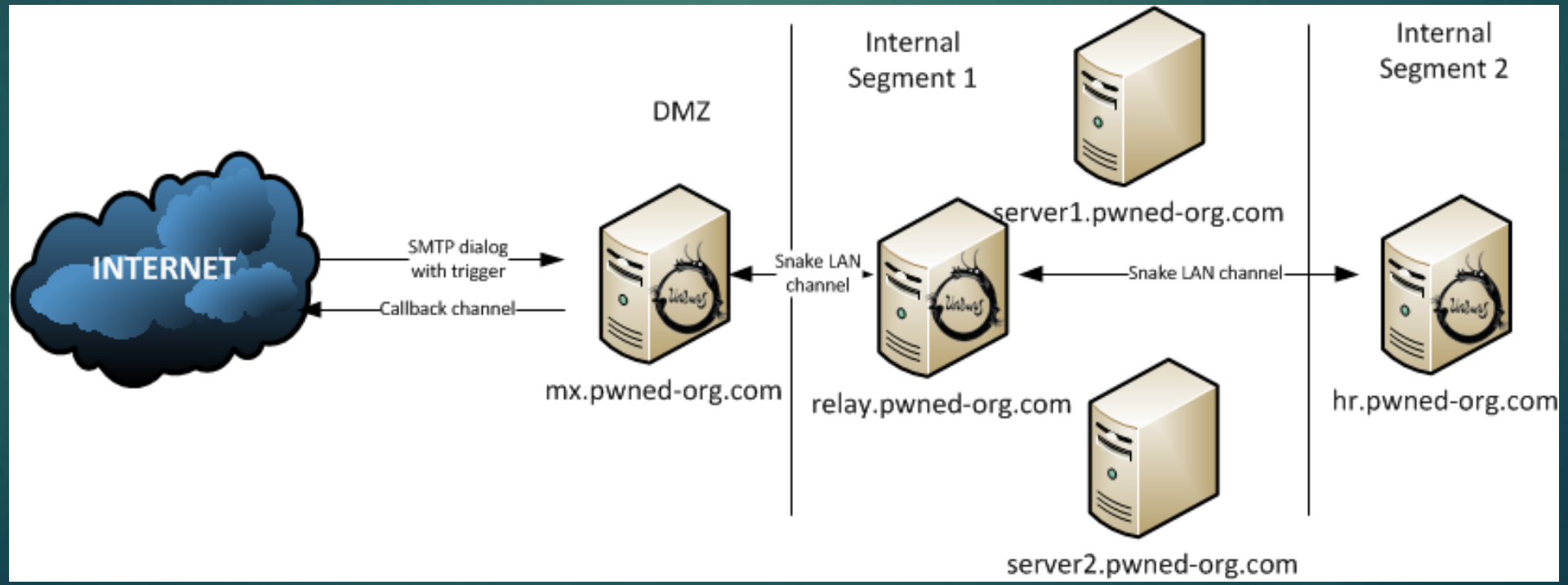
- CVE-2013-3346 - zero day used together with a known exploit
- Sets ThreadHideFromDebugger – breaks debugging
- Creates a new process in suspended mode and maps the same section of memory twice, in two different processes
- SetWindowsLong API call to start a thread in the newly created process – breaks most malware sandboxes
- Jumps several times from one process to another
- Wipes out the PE section so that it is harder to rebuild the unpacked executable

Stage 2: lateral movements

- Stage 1 C&C servers are easy targets – for example, they can be caught in spear phishing e-mails and sandbox
- Stage 2 backdoor: So let's replace this by a less known backdoor
- Go after Domain Admin credentials
- Further explore and compromise the network

Stage 3: Turla

- Network has been found interesting to explore long-term and exfiltrate
- Is fully compromised
- Turla dropped on chosen machines
- Usage of many other tools
- Some networks owned for years…



How to detect Turla ?

- Not very easy task …
- One fun story to tell ☺

- Do not only rely on vendors - talk to your partner organizations
- Establish relationships and share information !
- IOCs exchange is good but not enough these days:
  - They are easy to change by the intruders
  - Separate samples and infras used for different victims
- Good Yara sigs and custom detection tools can help
- Check your third party suppliers – for intruders it's a perfect way to get in

Divagations on attribution

- Development:
  - Vlad, gilg, urik
  - “Transmittion”, “Password it’s wrong” etc.
  - Zagruzchik.dll
- Operations:
  - Geographic distribution of infections
  - Virustotal submission countries
  - `$default_charset = 'Windows-1251';`

Questions ?

deresz@gmail.com
@deresz666