F-SECURE.COM

News from the Lab

F-Secure.

Search

Go

MAIN INDEX
ARCHIVES
ABOUT US

F-SECURE LABS
SUBMIT SAMPLES

SAFE AND SAVVY
TWITTER/F-SECURE
@FSLABS
@5EAN5ULLIVAN
@MIKKO
FACEBOOK
YOUTUBE/FSLABS
YOUTUBE/FSNEWS

TRY F-SECURE

<<< >>>

Monday, June 23, 2014

## Havex Hunts For ICS/SCADA Systems

Posted by Daavid @ 14:46 GMT

During the past year, we've been keeping a close eye on the Havex malware family and the group behind it. Havex is known to be used in targeted attacks against different industry sectors, and it was earlier reported to have specific interest in the energy sector.

The main components of Havex are a general purpose Remote Access Trojan (RAT) and a server written in PHP. The name "Havex" is clearly visible in the server source code:

```
define("PATH_BLOCKFILE", "block.tmp");
define("PATH_LOGFILE", "testlog.php");
define("DATATAG_START", "<html><head><mega http-equiv='CACHE-CONTROL' "
       "content='NO-CACHE'></head><body>No data!<!--havex");
define("DATATAG_END", "havex--></body></head>");
define("NODATA", "<html><head><mega http-equiv='CACHE-CONTROL' "
       "content='NO-CACHE'></head><body>Sorry, no data corresponding your request."
       "<!--havexhavex--></body></html>");
define("ANSWERTAG_START", "<xdata d='%s' u='%s'>");
define("ANSWERTAG_END", "</xdata>\n");
define("FILE_OUTPUT_BLOCK_SIZE", 16384);
```

During the spring of 2014, we noticed that Havex took a specific interest in Industrial Control Systems (ICS) and the group behind it uses an innovative trojan horse approach to compromise victims. The attackers have trojanized software available for download from ICS/SCADA manufacturer websites in an attempt to infect the computers where the software is installed to.

We gathered and analyzed 88 variants of the Havex RAT used to gain access to, and harvest data from, networks and machines of interest. This analysis included investigation of 146 command and control (C&C) servers contacted by the variants, which in turn involved tracing around 1500 IP addresses in an attempt to identify victims.

The attackers use compromised websites, mainly blogs, as C&C servers. Here are some examples of command and control servers used:

```
abainternationaltoursandtravel.com
adultfriendgermany.com
africancranesafaris.com
alexvernigor.com
al-mashkoor.com
alpikaclub.com
antibioticsdrugstore.com
arsch-anus.com
artem.sataev.com
artsepid.com
ask.az
atampy.com
aziaone.com
```

We also identified an additional component used by the attackers that includes code to harvest data from infected machines used in ICS/SCADA systems. This indicates that the attackers are not just interested in compromising the networks of companies they are interested in, but are also motivated in having control of the ICS/SCADA systems in those organizations. The source of this motivation is unclear to us.

### Trojanized Software as an Infection Vector

The Havex RAT is distributed at least through following channels:

- Spam email
- Exploit kits
- Trojanized installers planted on compromised vendor sites

The spam and exploit kit channels are fairly straightforward distribution mechanisms and we won't analyze them in more detail here.

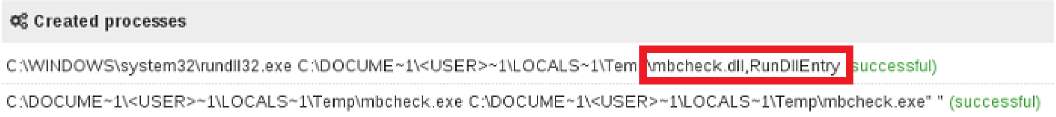
Of more interest is the third channel, which could be considered a form of "watering-hole attack", as the attackers chose to compromise an intermediary target - the ICS vendor site - in order to gain access to the actual targets.

It appears the attackers abuse vulnerabilities in the software used to run the websites to break in and replace legitimate software installers available for download to customers.

Our research uncovered three software vendor sites that were compromised in this manner. The software installers available on the sites were trojanized to include the Havex RAT. We suspect more similar cases exist but have not been identified yet.

Based on the content of their websites, all three companies are involved in development of applications and appliances for use in industrial applications. These organizations are based in Germany, Switzerland and Belgium. Two of them are suppliers of remote management software for ICS systems and the third develops high-precision industrial cameras and related software.

As an example, we can see the partial results of dynamic analysis for one of the trojanized installers:



Created processes

```
C:\WINDOWS\system32\rundll32.exe C:\DOCUME~1\<USER>~1\LOCALS~1\Tem \mbcheck.dll,RunDllEntry successful)
C:\DOCUME~1\<USER>~1\LOCALS~1\Temp\mbcheck.exe C:\DOCUME~1\<USER>~1\LOCALS~1\Temp\mbcheck.exe" " (successful)
```

The normal, clean installer does not include a file called "mbcheck.dll". This file is actually the Havex malware. The trojanized software installer will drop and execute this file as a part of the normal installation. The user is left with a working system, but the attacker now has a backdoor to access and control the computer.

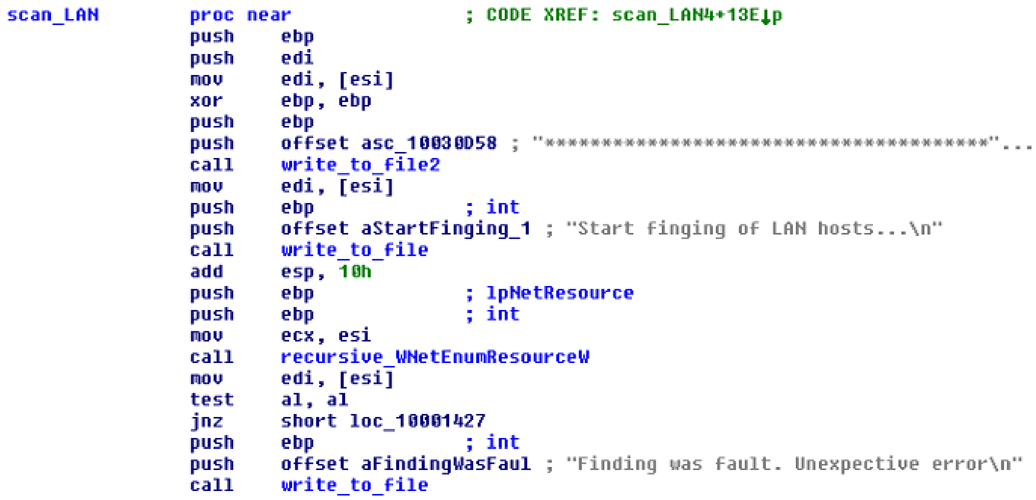
### Target Organizations

We were able to locate some of the infected systems and identify the organization affected by the samples analyzed in this report by tracing the IP addresses communicating to the C&C servers used by the Havex RAT.

All of these entities are associated in some way with the development or use of industrial applications or machines. The majority of the victims are located in Europe, though at the time of writing at least one company in California was also observed sending data to the C&C servers. Of the European-based organizations, two are major educational institutions in France that are known for technology-related research; two are German industrial application or machine producers; one is a French industrial machine producer; and one is a Russian construction company that appears to specialize in structural engineering.

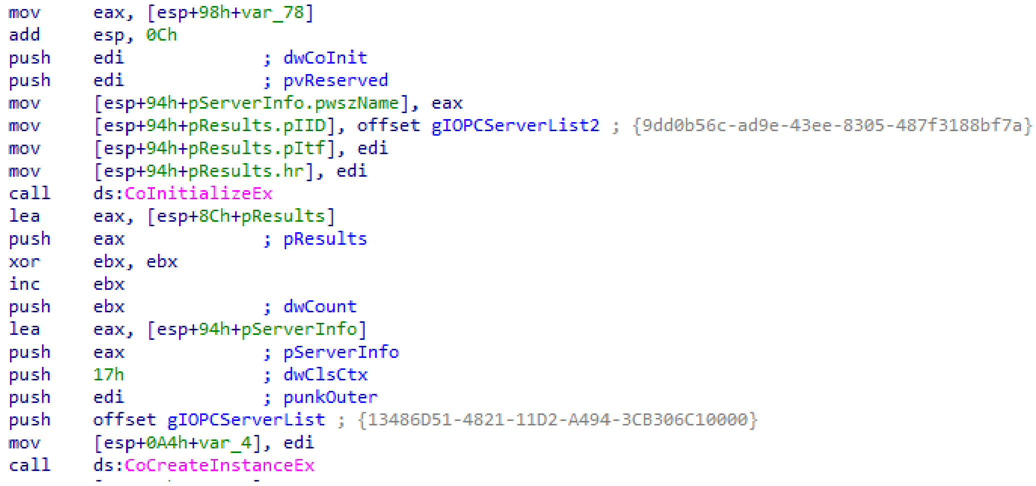
### ICS/SCADA Sniffer

Our analysis of Havex sample codes also uncovered its "ICS/SCADA sniffing" behavior. The C&C server will instruct infected computers to download and execute further components, and one of these components appeared very interesting. While analyzing this component, we noticed that it enumerates the local area network and looks for connected resources and servers:



```
scan_LAN        proc near               ; CODE XREF: scan_LAN4+13E↓p
                push    ebp
                push    edi
                mov     edi, [esi]
                xor     ebp, ebp
                push    ebp
                push    offset asc_10030D58 ; "****************************************"...
                call    write_to_file2
                mov     edi, [esi]
                push    ebp             ; int
                push    offset aStartFinging_1 ; "Start finging of LAN hosts...\n"
                call    write_to_file
                add     esp, 10h
                push    ebp             ; lpNetResource
                push    ebp             ; int
                mov     ecx, esi
                call    recursive_WNetEnumResourceW
                mov     edi, [esi]
                test    al, al
                jnz     short loc_10001427
                push    ebp             ; int
                push    offset aFindingWasFaul ; "Finding was fault. Unexpective error\n"
                call    write_to_file
```

We then noticed that it uses Microsoft Component Object Model (COM) interfaces (CoInitializeEx, CoCreateInstanceEx) to connect to specific services:

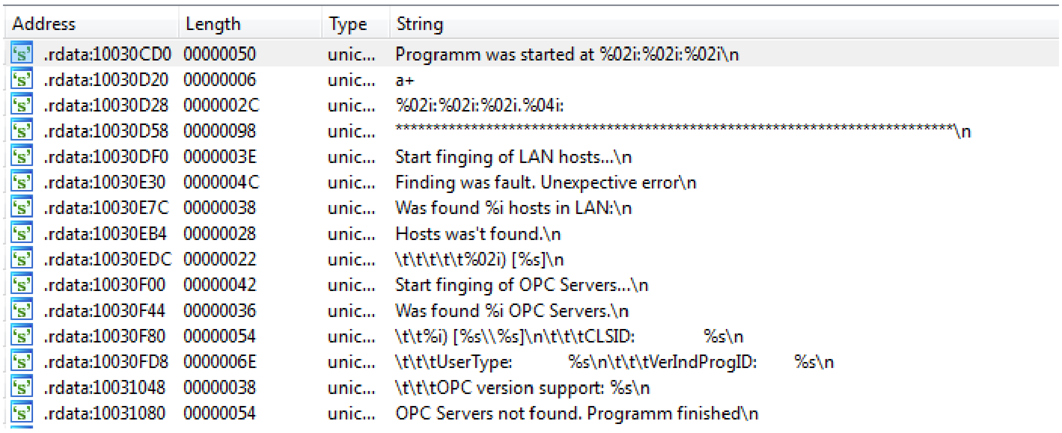


```
mov     eax, [esp+98h+var_78]
add     esp, 0Ch
push    edi             ; dwCoInit
push    edi             ; pvReserved
mov     [esp+94h+pServerInfo.pwszName], eax
mov     [esp+94h+pResults.pIID], offset gIOPCServerList2 ; {9dd0b56c-ad9e-43ee-8305-487f3188bf7a}
mov     [esp+94h+pResults.pItf], edi
mov     [esp+94h+pResults.hr], edi
call    ds:CoInitializeEx
lea     eax, [esp+8Ch+pResults]
push    eax             ; pResults
xor     ebx, ebx
inc     ebx
push    ebx             ; dwCount
lea     eax, [esp+94h+pServerInfo]
push    eax             ; pServerInfo
push    17h             ; dwClsCtx
push    edi             ; punkOuter
push    offset gIOPCServerList ; {13486D51-4821-11D2-A494-3CB306C10000}
mov     [esp+0A4h+var_4], edi
call    ds:CoCreateInstanceEx
```

To identify which services the sample is interested in, we can simply search for the identifiers seen above, which tell us what kind of interfaces are being used. A bit of googling gives us these names:

- **9DD0B56C-AD9E-43EE-8305-487F3188BF7A** = IID_IOPCServerList2
- **13486D51-4821-11D2-A494-3CB306C10000** = CLSID_OPCServerList

Note the mention of "OPCServer" in the names. There are more hints pointing in the same direction -- the strings found in the executable also make several references to “OPC”:



| Address | Length | Type | String |
|---|---|---|---|
| .rdata:10030CD0 | 00000050 | unic... | Programm was started at %02i:%02i:%02i\n |
| .rdata:10030D20 | 00000006 | unic... | a+ |
| .rdata:10030D28 | 0000002C | unic... | %02i:%02i:%02i.%04i: |
| .rdata:10030D58 | 00000098 | unic... | **************************************************************************\n |
| .rdata:10030DF0 | 0000003E | unic... | Start finging of LAN hosts...\n |
| .rdata:10030E30 | 0000004C | unic... | Finding was fault. Unexpective error\n |
| .rdata:10030E7C | 00000038 | unic... | Was found %i hosts in LAN:\n |
| .rdata:10030EB4 | 00000028 | unic... | Hosts was't found.\n |
| .rdata:10030EDC | 00000022 | unic... | \t\t\t\t\t%02i) [%s]\n |
| .rdata:10030F00 | 00000042 | unic... | Start finging of OPC Servers...\n |
| .rdata:10030F44 | 00000036 | unic... | Was found %i OPC Servers.\n |
| .rdata:10030F80 | 00000054 | unic... | \t\t%i) [%s\\%s]\n\t\t\tCLSID:      %s\n |
| .rdata:10030FD8 | 0000006E | unic... | \t\t\tUserType:      %s\n\t\t\tVerIndProgID:      %s\n |
| .rdata:10031048 | 00000038 | unic... | \t\t\tOPC version support: %s\n |
| .rdata:10031080 | 00000054 | unic... | OPC Servers not found. Programm finished\n |

It turns out that OPC stands for OLE for Process Control, and it's a standard way for Windows applications to interact with process control hardware. Using OPC, the malware component gathers any details about connected devices and sends them back to the C&C for the attackers to analyze. It appears that this component is used as a tool for intelligence gathering. So far, we have not seen any payloads that attempt to control the connected hardware.

**Summary**

The attackers behind Havex are conducting industrial espionage using a clever method. Trojanizing ICS/SCADA software installers is an effective method in gaining access to target systems, potentially even including critical infrastructure.

The method of using compromised servers as C&C's is typical for this group. The group doesn't always manage the C&C's in a professional manner, revealing lack of experience in operations. We managed to monitor infected computers connecting to the servers and identify victims from several industry sectors.

The additional payload used to gather details about ICS/SCADA hardware connected to infected devices shows the attackers have direct interest in controlling such environments. This is a pattern that is not commonly observed today.

SHA-1 hashes of the samples discussed:

7f249736efc0c31c44e96fb72c1efcc028857ac7
1c90ecf995a70af8f1d15e9c355b075b4800b4de
db8ed2922ba5f81a4d25edb7331ea8c0f0f349ae
efe9462bfa3564fe031b5ff0f2e4f8db8ef22882

F-Secure detects this threat as **Backdoor:W32/Havex.A**.

Post by — Daavid and Antti

<<< Necurs - Rootkit For Hire | You Are Responsible For Your Security And Privacy >>>

- RSS
- **Contact Us** |
- Copyright & Privacy |