“Winnti”

More than just a game

April 2013
Kaspersky Lab Global Research and Analysis Team

Contents

Executive Summary

This research, which started in autumn of 2011 by Kaspersky Lab, is still ongoing. The subject of this research project is a series of targeted attacks against private companies around the world.

In the research, we reveal activity of one of the hacking groups which has Chinese origins. This group was named "Winnti".

According to our estimates, the Winnti group has been active for several years and specializes in cyber-attacks against the online video game industry. The main objective of the group is to steal source code of online game projects as well as digital certificates of legitimate software vendors. Besides that, they are deeply interested in the set-up of network infrastructure (including production gaming servers) and new developments such as conceptual ideas, design and more.

We aren't the first to investigate the attacks attributed to the Winnti group.. It is known that, at least in 2010, the U.S.- based company HBGary investigated information security incidents related to the Winnti group at one of HBGary's customers – an American video game production company.

In the beginning …

In the autumn of 2011, a Trojan was detected on a large number of computers – all of them linked by the fact that they were used by players of a popular online game. It emerged that the piece of malware landed on users' computers as part of a regular update from the game's official update server. Some even suspected that the publisher itself was spying on its customers. However, it later became clear that the malicious program ended up on the users' computers by mistake: the cybercriminals were in fact targeting the companies that develop and release computer games.

The computer game publisher whose servers spread the Trojan asked Kaspersky Lab to analyze the malicious program that was found on its update server. It turned out to be a DLL library compiled for a 64-bit Windows environment and even used a properly signed malicious driver.

The malicious DLL infected gamers' computers running under either 32-bit or 64-bit operating systems. It could not start in 32-bit environments, but it could, under certain conditions, launch without the user's knowledge or consent in 64-bit environments, though no such accidental launches have been detected.

The DLL contained a backdoor payload, or, to be exact, the functionality of a fully-fledged Remote Administration Tool (RAT), which gave the cyber-criminals the ability to control the victim computer without the user's knowledge.

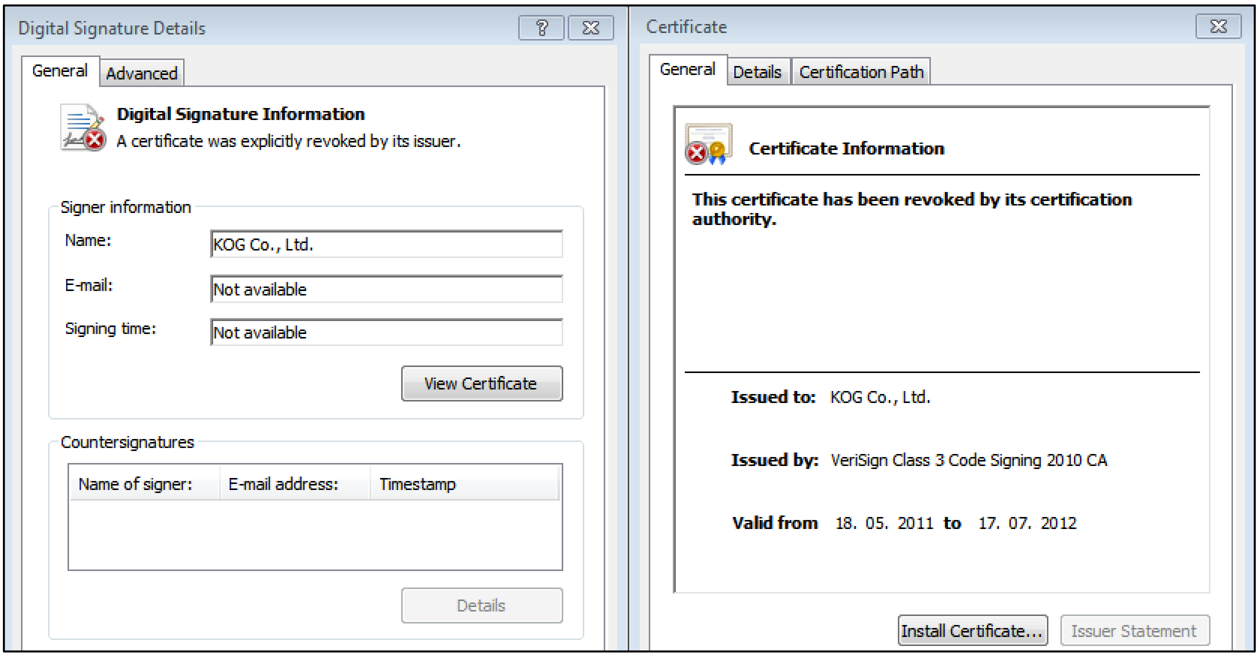
The malicious module turned out to be the first Trojan for the 64-bit version of Microsoft Windows with valid digital signature that we have seen. We used to see similar cases before, but in all previous incidents we have seen digital signature abuse, there were only 32-bit applications.

At an early stage of our research, we identified a number of similar backdoors, both 32-bit and 64-bit, in our collection of malware samples. Thesewere detected under various verdicts. We grouped them together into a separate family. Symantec appears to be the first to name these malicious programs; we kept Symantec's name –

Winnti – in the name of the malware family we created: Backdoor.Win32(Win64).Winnti. As for the people behind these attacks involving this remote administration tool, we ended up calling them “the Winnti group”.

Interestingly, the digital signature belonged to another video game vendor - a private company known as KOG, based in South Korea. The main business of this company was MMRPG (massively multi player online role-playing games) games, which was identical to the business area of the first victim.

We contacted KOG, whose certificate was used to sign malicious software and notified Verisign, which issued the certificate for KOG. As a result, the certificate was revoked.



Digital Signature Information
A certificate was explicitly revoked by its issuer.

Signer information

Name: KOG Co., Ltd.
E-mail: Not available
Signing time: Not available

Countersignatures

Name of signer:	E-mail address:	Timestamp

Certificate Information

This certificate has been revoked by its certification authority.

Issued to: KOG Co., Ltd.

Issued by: VeriSign Class 3 Code Signing 2010 CA

Valid from 18. 05. 2011 **to** 17. 07. 2012

Digital Certificates

When we discovered the first stolen digital certificate, we didn't realize that stealing the certificates and signing malware for upcoming attacks against other victims was the modus operandi of that group. In eighteen months, we manage to discover more than a dozen compromised digital certificates.

Moreover, we found that those digital certificates seemed to have been used in attacks organized by other hacking groups, presumably coming from China.

For example, an attack against South Korean social networks Cyworld and Nate in 2011 (http://www.bbc.co.uk/news/technology-14323787) - the attackers used a Trojan that was digitally signed using the certificate of YNK Japan Inc gaming company.)

A digital certificate of the same company was used recently (March 2013) in Trojans targeting Tibetan and Uyghur activists (https://www.securelist.com/en/blog/208194165/New_Uyghur_and_Tibetan_Themed_Attacks_Using_PDF_Exploits).
In fact, this story has long roots dating back to 2011. We highly recommend reading this Norman blog post of a similar incident here: http://blogs.norman.com/2011/security-research/invisible-ynk-a-code-signing-conundrum.

At the same time, in March 2013, Uyghur activists were targeted by another malware which was digitally signed by another gaming company called MGAME Corp according to http://www.f-secure.com/weblog/archives/00002524.html

We believe that the source of all these stolen certificates is same group which we call Winnti. This group either has close contacts with other Chinese hacker groups or sells the certificates on the black market in China.

Below is the list of known compromised companies and digital certificates used by the Winnti group in different campaigns:

Company	Serial number	Country
ESTsoft Corp	30 d3 fe 26 59 1d 8e ac 8c 30 66 7a c4 99 9b d7	South Korea
Kog Co., Ltd.	66 e3 f0 b4 45 9f 15 ac 7f 2a 2b 44 99 0d d7 09	South Korea
LivePlex Corp	1c aa 0d 0d ad f3 2a 24 04 a7 51 95 ae 47 82 0a	South Korea/ Philippines
MGAME Corp	4e eb 08 05 55 f1 ab f7 09 bb a9 ca e3 2f 13 cd	South Korea
Rosso Index KK	01 00 00 00 00 01 29 7d ba 69 dd	Japan
Sesisoft	61 3e 2f a1 4e 32 3c 69 ee 3e 72 0c 27 af e4 ce	South Korea
Wemade	61 00 39 d6 34 9e e5 31 e4 ca a3 a6 5d 10 0c 7d	Japan/South Korea/US
YNK Japan	67 24 34 0d db c7 25 2f 7f b7 14 b8 12 a5 c0 4d	Japan
Guangzhou YuanLuo	0b 72 79 06 8b eb 15 ff e8 06 0d 2c 56 15 3c 35	China
Fantasy Technology Corp	75 82 f3 34 85 aa 26 4d e0 3b 2b df 74 e0 bf 32	China
Neowiz	5c 2f 97 a3 1a bc 32 b0 8c ac 01 00 59 8f 32 f6	South Korea

Victims

It’s tempting to assume that Advanced Persistent Threats (APTs) primarily target high-level institutions: government agencies, ministries, military and political organizations, power stations, chemical plants, critical infrastructure networks, and so on. In this context, it seems unlikely that a commercial company would be at risk unless it was operating on the scale of Google, Adobe or The New York Times, which was recently targeted by a cyber-attack, and this perception is reinforced by the publicity that attacks on corporations and government organizations usually receive. However, any company with data that can be effectively monetized is at risk from APTs. This is exactly what we encountered here: it was not a governmental, political, military, or industrial organization. The target was specifically gaming companies.

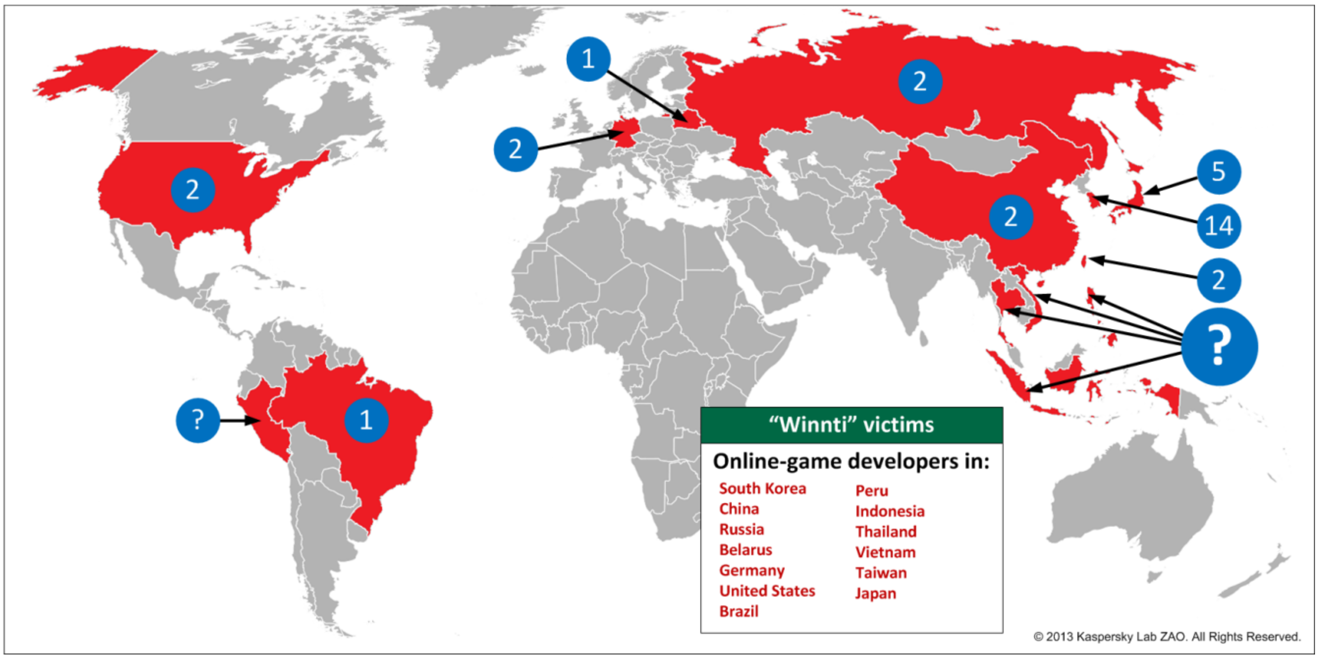
Analyzing the Winnti samples helped to identify who and what were the targets. We found that we were dealing with targeted attacks: the Winnti team infects companies that develop and release computer games. It appears the team has been active for quite a while – since 2009.

It’s difficult to name all the victims of the Winnti team. Judging by the information that we have at our disposal – namely the tags within malicious programs, the names of the C&C domains, the companies whose digital certificates were stolen to sign malware, and the countries where detection notifications came from – we can say that at least 35 companies were infected by the Winnti malware at some time.

Countries where the affected companies are located:

Asia	Europe	South America	North America
China India Indonesia Japan Philippines S. Korea Taiwan Thailand Vietnam	Belarus Germany Russia	Brazil Peru	USA

This data demonstrates that the Winnti team targets gaming companies located in various parts of the world, albeit with a strong focus on South East Asia.



© 2013 Kaspersky Lab ZAO. All Rights Reserved.

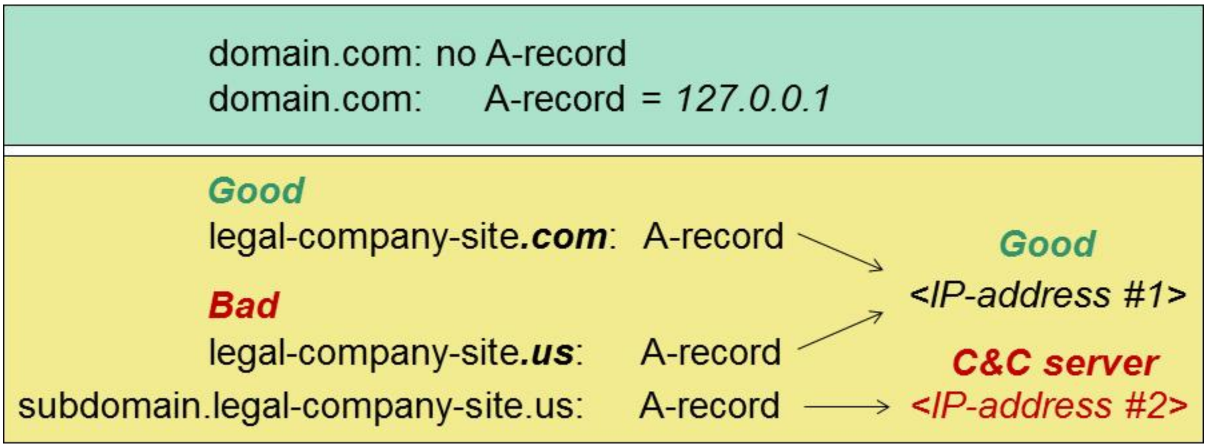
Countries where gaming companies have been affected

This geographic diversity is hardly surprising. Often, gaming companies (both publishers and developers) are international, having representatives and offices worldwide. Also, it is common practice for gaming companies from various regions to cooperate. The developers of a game may be located in a different country from its publisher. When a game eventually reaches markets in regions away from its initial ‘home’, it is often localized and released by other publishers. In the course of this cooperation, the partner companies often grant each other access to network resources to exchange data associated with the gaming content, including distribution kits, gaming resources, resource assembly kits, etc. If one company in the network gets infected, it’s easy for the cybercriminals to spread the infection throughout the partnership chain.

Winnti C&C Structure

During the investigation, we identified more than a hundred malicious programs, each individually compiled to attack a particular company. Typically, separate command-and-control (C&C) domains were assigned to each targeted company. Virtually all the C&C domains were arranged as follows: a second-level domain was created without a DNS A-record, i.e., there was no IP address assigned to it.

In cases where there was an A-record, the assigned IP address was typically 127.0.0.1. It is also noteworthy that some of the second-level domains that the cybercriminals created for their C&C had very similar names to the domain hosting the site of a certain real gaming company. And the malicious users’ domain was resolved to the same IP address which the site of the real gaming company used. In any case, the third-level domains resolved to IP addresses assigned to the attackers’ actual C&C servers.


domain.com: no A-record
domain.com: A-record = 127.0.0.1
Good
legal-company-site.com: A-record
Bad
legal-company-site.us: A-record
subdomain.legal-company-site.us: A-record
Good
<IP-address #1>
C&C server
<IP-address #2>

C&C domain naming and resolution

Sometimes the Winnti team registered their C&C units with public hosts. Judging by the samples identified, these C&C centers were subdomains of such domains as 6600.org, 8866.org, 9966.org or ddns.net.

From the names of the C&C domains or subdomains, the attack targets or countries of residence could be guessed, as in:

ru.gcgame.info
kr.zzsoft.info
jp.xxoo.co
us.nhntech.com
fs.nhntech.com
as.cjinternet.us

The subdomains “ru”, “kr”, ”jp” and “us” most probably mean that these C&C servers manage bots hosted on the computers of companies located in Russia, South Korea, Japan and the U.S. respectively, while “fs” and “as” are acronyms for the names of the companies being attacked.

Sometimes Winnti’s malicious programs had a local IP address, such as 192.168.1.136, specified in the settings for the C&C. This could mean that, at some point in time, there was an infected computer that did not have a connection to the Internet, but the cybercriminals needed control over it (it may have been infected while malware was spread via a corporate network). In this case, the cybercriminals deployed a dedicated local C&C server on another compromised computer within the same local network which did have an Internet connection; via that C&C, the first victim computer could be controlled. System administrators often try to isolate critical computers from the outside world. This decreases the probability of haphazard infection, but, apparently, does not always help in a targeted attack.

In the Winnti samples that were detected and analyzed, we found 36 unique C&C domains. Most probably, this is only a small portion of all existing Winnti C&C domains, as we only managed to obtain some of the samples from this malware family. This is hardly surprising since these malicious programs are used to execute targeted attacks, so no information is available about many instances of infection; for this reason, we have no way of obtaining samples of the malware used in these undisclosed attacks.

Domain names used in the attacks we discovered
newpic.dyndns.tv
update.ddns.net
nd.jcrsoft.com
cc.nexoncorp.us
kr.zzsoft.info
as.cjinternet.us
ca.zzsoft.info
sn.jcrsoft.com
lp.apanku.com
sshd.8866.org
ftpd.6600.org
tcpiah.googleclick.net
rss.6600.org
lp.zzsoft.info
lp.gasoft.us
eya.jcrsoft.com
ftpd.9966.org
kr.xxoo.co
wi.gcgame.info
tcp.nhntech.com
ka.jcrsoft.com
my.zzsoft.info
jp.jcrsoft.com
su.cjinternet.us
vn.gcgame.info
ap.nhntech.com
ru.gcgame.info
kr.jcrsoft.com
wm.ibm-support.net
fs.nhntech.com
docs.nhnclass.com
rh.jcrsoft.com
wm.nhntech.com

```
wm.myxxoo.com
ka.zzsoft.info
ad.jcrsoft.com
my.gasoft.us
```

Knowing the 2nd level domains used by Winnti, we brute forced through all third level sub-domains up to 4 symbols long, and identified those that have the IP addresses of real servers assigned to them. Having searched through subdomains for a total of 12 second level domains, we identified 227 “live” third level domains. Many of them are C&C servers for Winnti-class malware that have hitherto remained unidentified.

Analyzing the WHOIS data for the 12 second level domains, we found the following list of email addresses used for registration:

evilsex@gmail.com
jslee.jcr@gmail.com
whoismydns@gmail.com
googl3@live.com
wzcc@cnkker.com
apanku2009@gmail.com

For some of these domains, registration data proved to be the same as those for the domain google.com:

Registrant: Google Inc.
1600 Amphitheatre Parkway
Mountain View, California 94043
United States
+1.6503300100

Judging by the domain registration data, the Winnti team started their criminal activities as far back as 2007. The early domains were involved in spreading rogue anti-virus programs (FakeAV). From 2009 onwards, domains began to emerge hosting C&C servers for bots used to infect gaming companies. Apparently, the cybercriminals graduated to relatively large-scale penetrations into the corporate networks of gaming companies starting from 2010.

Known Malware

The favorite tool of the attackers is a malicious program we call "Winnti". It has evolved since the first use, but we divide all variants into two generations: 1.x and 2.x. Our publication describes both variants of this tool. The second generation (2.x) was used in one of the attacks that we investigated in the active stage and helped the victim to interrupt data transfer and isolate infections in a corporate network.

In addition to that, we observed usage of a popular backdoor known as PlugX, which is believed to have Chinese origins, however used only previously in attacks against Tibetan activists.

The Commercial Interest

As has been stated above, APTs can target any commercial company if cyber-criminals find a way to financially profit from the attack.

So what methods do cyber-criminals use to generate illicit earnings from attacks on gaming companies?

Based on the available information, we have singled out three main monetization schemes that could be used by the Winnti team.

- **The unfair accumulation of in-game currency/“gold” in online games and the conversion of virtual funds into real money.**
- **Theft of source code from the online games server to search for vulnerabilities in games – often linked to point 1.**
- **Theft of source code from the server part of popular online games to further deploy pirate servers.**

Let’s look at an example. During our investigation of an infection at a computer gaming company, we found that malware had been created for a particular service on the company’s server. The malicious program was looking for a specific process running on the server, injected code into it, and then sought out two places in the process code where it could conceal call commands for its function interceptors. Using these function interceptors, the malicious programs modified process data which was processed in those two places, and returned control back. Thus, the attackers change the normal execution of the server processes. Unfortunately, the company was not able to share its targeted application with us, and we cannot say exactly how this malicious interference affected gaming processes. The company concerned told us that the attackers’ aim was to acquire gaming “gold” illegally.

Malicious activity like this has an adverse impact on the game itself, tilting the balance in favor of cheats. But any changes the Winnti team introduces into the game experience are unlikely to be very noticeable. After all, maintaining a skillful balance is the main attribute of online games. Users will simply stop playing if they feel that other players are using non-standard methods to create an advantage beyond normal gameplay or if the game loses its intrinsic competitiveness due to resources or artifacts appearing in the game without the developers’ knowledge. At the same time, the attackers are keen for the game to remain popular; otherwise, they would be unable to effectively turn all the time and effort of infecting a gaming company into financial gain.

Members of the Winnti team are patient and cautious. Cyber-criminals have affected the processes of the online games from the infected companies and stolen money from them for years, but they have found ways of doing this without attracting attention to themselves.

Winnti 1.0 Technical Analysis

The Initial DLL

Everything starts with a DLL. The DLL mimics one of the standard Windows libraries, winmm.dll or apphelp.dll. Since, in the vast majority of cases the samples that we detected disguised themselves as winmm.dll, we would like to fix this name for this malicious library at the end of this document.

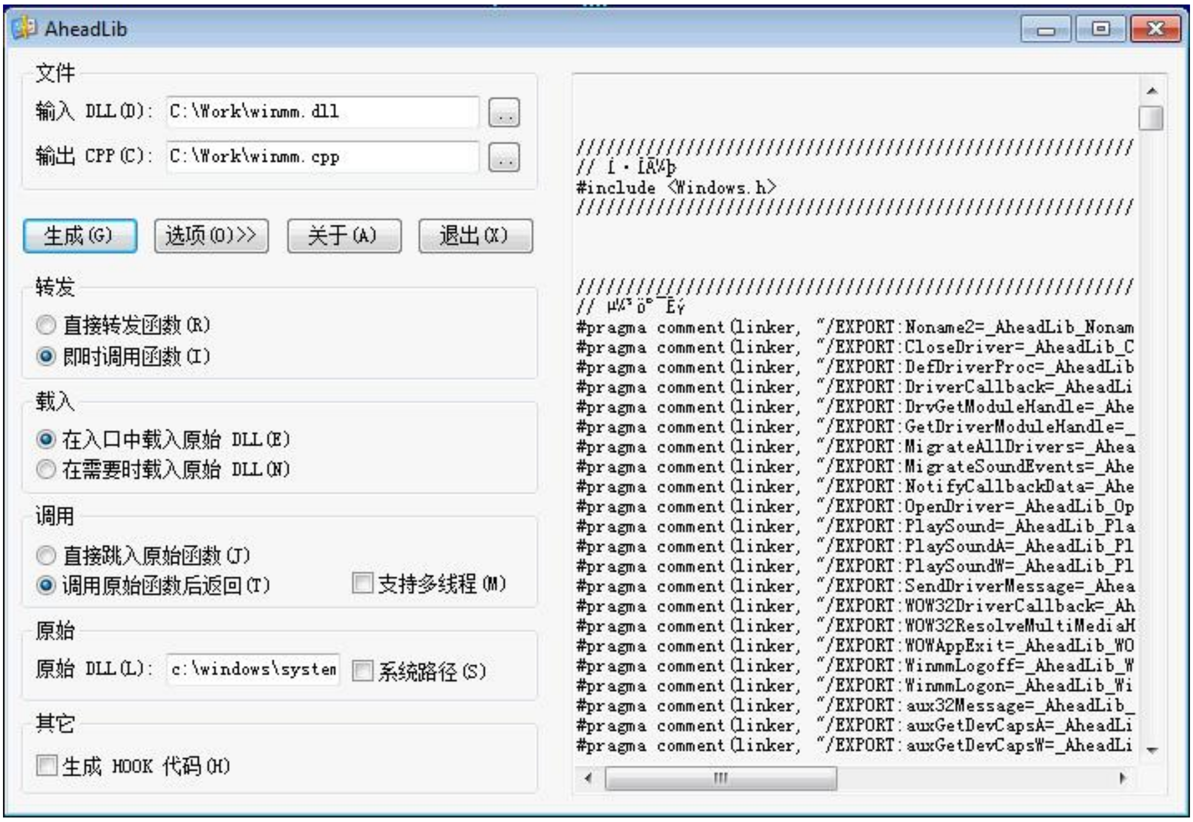
Legitimate winmm.dll is a Windows system library that provides multimedia functions. It is located in the %WINDIR%\System32 folder. The attackers counted on this being a library providing basic system functions and hence the probability of it being loaded by some program is very high (this is also valid for apphelp.dll). For example, winmm.dll is loaded by explorer.exe, which is launched during operating system startup and is essential for Windows user interface.

The mechanism to start the malware is simple: if some benign application depends on Windows winmm.dll (located in %WINDIR%\System32\winmm.dll) but the evil twin library with the same name (winmm.dll) is located in the folder of benign application, the malicious library will be loaded instead of the system one.

Taking advantage of their control of an infected computer, the attackers place a malicious library in the %WINDIR% folder. The same folder also contains explorer.exe. This enables the attackers to ensure that the malicious DLL is loaded at system startup: explorer.exe loads the malicious winmm.dll from the %WINDIR% folder as soon as it launches during system startup.

But how can a program which depends on the original library work correctly if a malicious winmm.dll is loaded instead of the original library? Very easy: the malicious library is designed to work as a proxy for the original winmm.dll from the %WINDIR%\System32 folder.

The cyber-criminals did not reinvent the wheel to make sure that everything works properly. They relied on a tool known as AheadLib, which was developed by security researchers to analyze malware.



文件

输入 DLL(D): C:\Work\winmm.dll

输出 CPP(C): C:\Work\winmm.cpp

转发

○ 直接转发函数(R)

● 即时调用函数(I)

载入

● 在入口中载入原始 DLL(E)

○ 在需要时载入原始 DLL(N)

调用

○ 直接跳入原始函数(J)

● 调用原始函数后返回(T)

□ 支持多线程(M)

原始

原始 DLL(L): c:\windows\systen

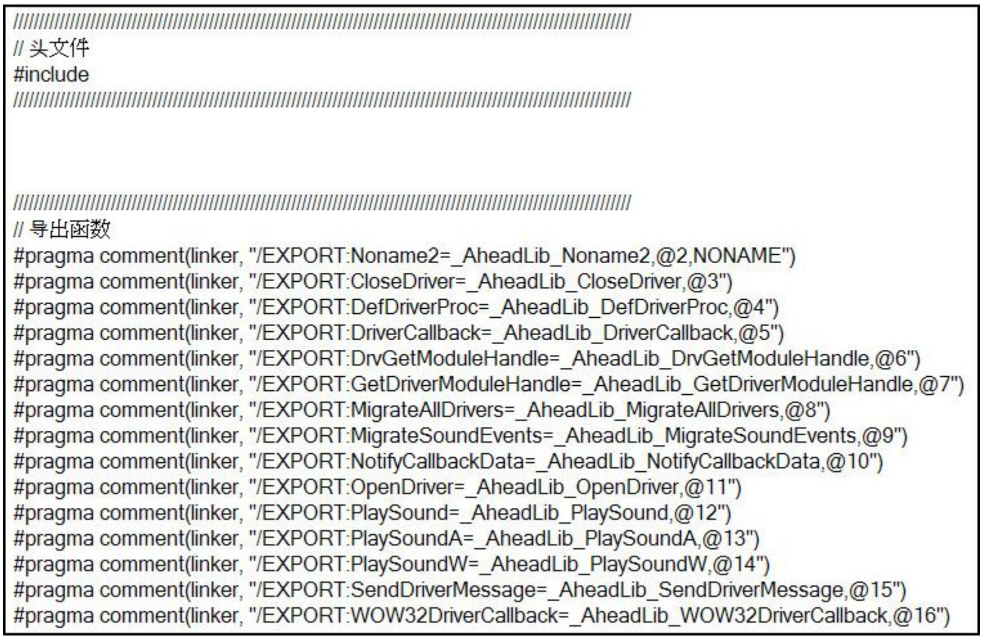
□ 系统路径(S)

其它

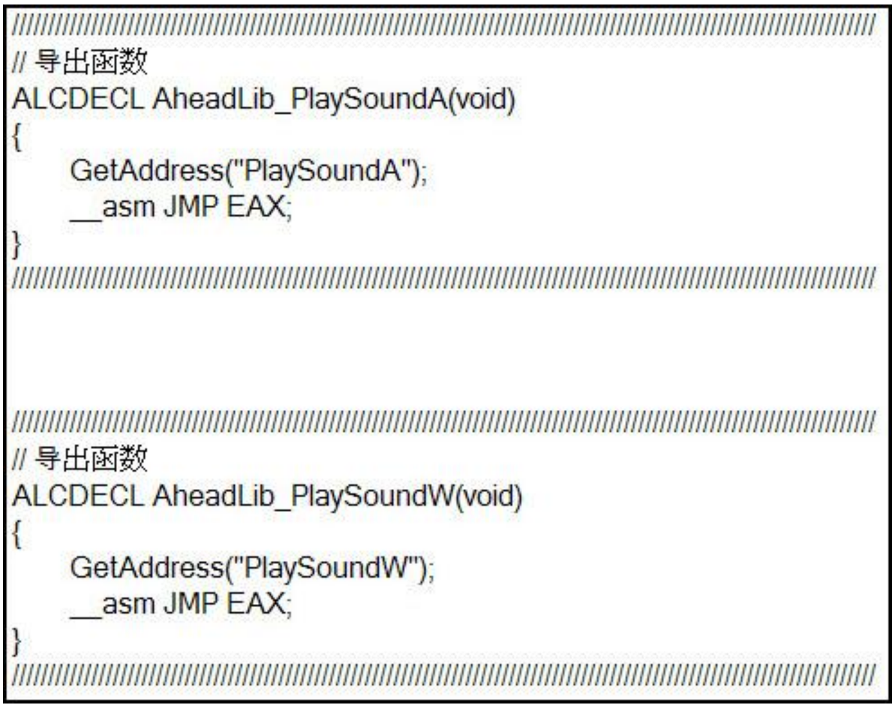
□ 生成 HOOK 代码(H)

```
/////////////////////////////////////////////////////////
// Í·ÎÄ¼þ
#include <Windows.h>
/////////////////////////////////////////////////////////




/////////////////////////////////////////////////////////
// µ¼³ö°¯Êý
#pragma comment(linker, "/EXPORT:Noname2=_AheadLib_Nonam
#pragma comment(linker, "/EXPORT:CloseDriver=_AheadLib_C
#pragma comment(linker, "/EXPORT:DefDriverProc=_AheadLib
#pragma comment(linker, "/EXPORT:DriverCallback=_AheadLi
#pragma comment(linker, "/EXPORT:DrvGetModuleHandle=_Ahe
#pragma comment(linker, "/EXPORT:GetDriverModuleHandle=_
#pragma comment(linker, "/EXPORT:MigrateAllDrivers=_Ahea
#pragma comment(linker, "/EXPORT:MigrateSoundEvents=_Ahe
#pragma comment(linker, "/EXPORT:NotifyCallbackData=_Ahe
#pragma comment(linker, "/EXPORT:OpenDriver=_AheadLib_Op
#pragma comment(linker, "/EXPORT:PlaySound=_AheadLib_Pla
#pragma comment(linker, "/EXPORT:PlaySoundA=_AheadLib_Pl
#pragma comment(linker, "/EXPORT:PlaySoundW=_AheadLib_Pl
#pragma comment(linker, "/EXPORT:SendDriverMessage=_Ahea
#pragma comment(linker, "/EXPORT:WOW32DriverCallback=_Ah
#pragma comment(linker, "/EXPORT:WOW32ResolveMultiMediaH
#pragma comment(linker, "/EXPORT:WOWAppExit=_AheadLib_WO
#pragma comment(linker, "/EXPORT:WinmmLogoff=_AheadLib_W
#pragma comment(linker, "/EXPORT:WinmmLogon=_AheadLib_Wi
#pragma comment(linker, "/EXPORT:aux32Message=_AheadLib_
#pragma comment(linker, "/EXPORT:auxGetDevCapsA=_AheadLi
#pragma comment(linker, "/EXPORT:auxGetDevCapsW=_AheadLi
```

This program, which is designed to facilitate the analysis of malicious libraries, was created by a Chinese developer employed by an Asian anti-virus vendor. The program accepts a DLL on input and produces a C code which hooks the functions included in the library. The C code is compiled back into a DLL, which can then be used as a proxy and provide flexible way to analyze behavior of malicious file.



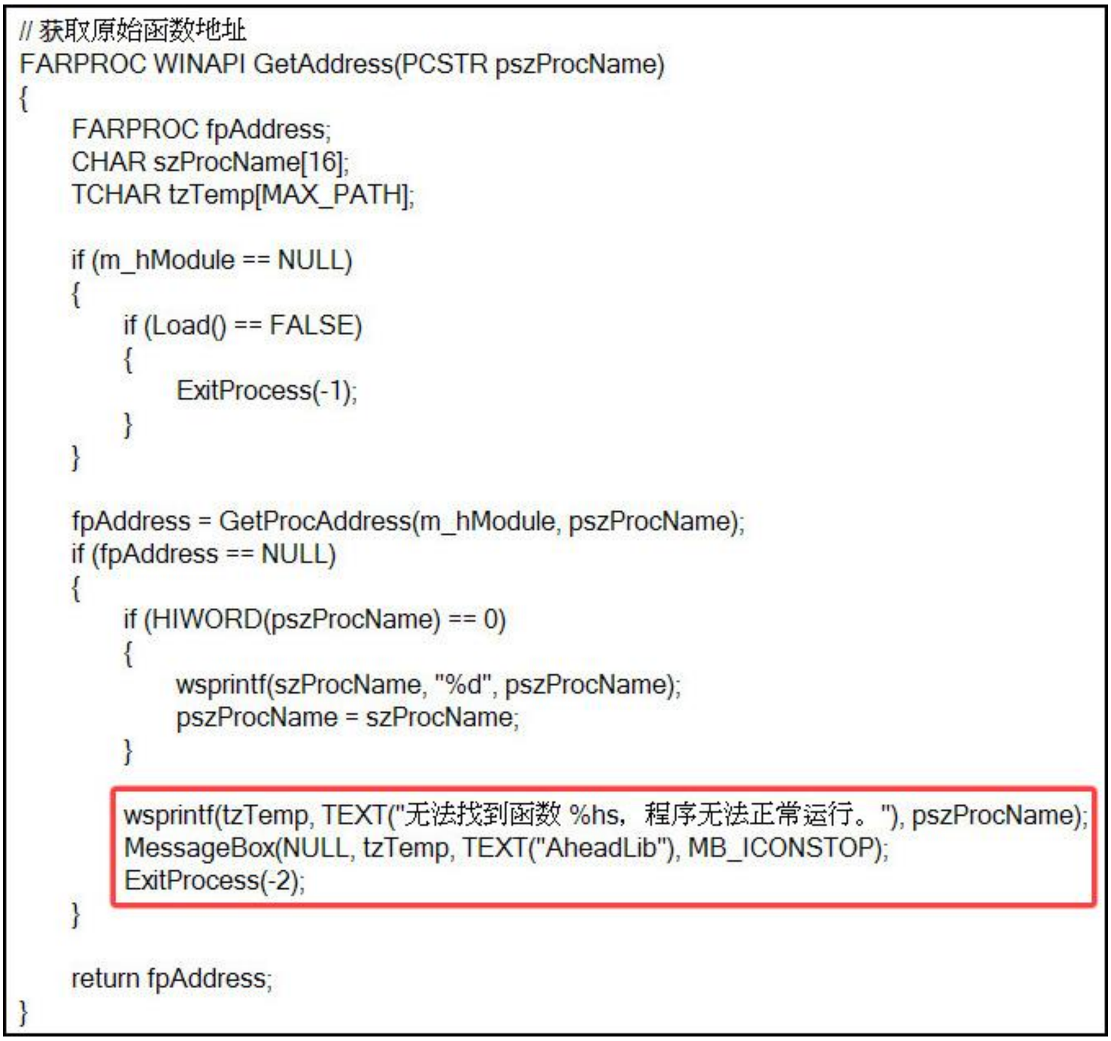
```
////////////////////////////////////////////////////////////////////////////////////////////////////////////
// 头文件
#include
////////////////////////////////////////////////////////////////////////////////////////////////////////////



////////////////////////////////////////////////////////////////////////////////////////////////////////////
// 导出函数
#pragma comment(linker, "/EXPORT:Noname2=_AheadLib_Noname2,@2,NONAME")
#pragma comment(linker, "/EXPORT:CloseDriver=_AheadLib_CloseDriver,@3")
#pragma comment(linker, "/EXPORT:DefDriverProc=_AheadLib_DefDriverProc,@4")
#pragma comment(linker, "/EXPORT:DriverCallback=_AheadLib_DriverCallback,@5")
#pragma comment(linker, "/EXPORT:DrvGetModuleHandle=_AheadLib_DrvGetModuleHandle,@6")
#pragma comment(linker, "/EXPORT:GetDriverModuleHandle=_AheadLib_GetDriverModuleHandle,@7")
#pragma comment(linker, "/EXPORT:MigrateAllDrivers=_AheadLib_MigrateAllDrivers,@8")
#pragma comment(linker, "/EXPORT:MigrateSoundEvents=_AheadLib_MigrateSoundEvents,@9")
#pragma comment(linker, "/EXPORT:NotifyCallbackData=_AheadLib_NotifyCallbackData,@10")
#pragma comment(linker, "/EXPORT:OpenDriver=_AheadLib_OpenDriver,@11")
#pragma comment(linker, "/EXPORT:PlaySound=_AheadLib_PlaySound,@12")
#pragma comment(linker, "/EXPORT:PlaySoundA=_AheadLib_PlaySoundA,@13")
#pragma comment(linker, "/EXPORT:PlaySoundW=_AheadLib_PlaySoundW,@14")
#pragma comment(linker, "/EXPORT:SendDriverMessage=_AheadLib_SendDriverMessage,@15")
#pragma comment(linker, "/EXPORT:WOW32DriverCallback=_AheadLib_WOW32DriverCallback,@16")
```



```
////////////////////////////////////////////////////////////////////////////////////////////////////////
// 导出函数
ALCDECL AheadLib_PlaySoundA(void)
{
    GetAddress("PlaySoundA");
    __asm JMP EAX;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////



////////////////////////////////////////////////////////////////////////////////////////////////////////
// 导出函数
ALCDECL AheadLib_PlaySoundW(void)
{
    GetAddress("PlaySoundW");
    __asm JMP EAX;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////
```

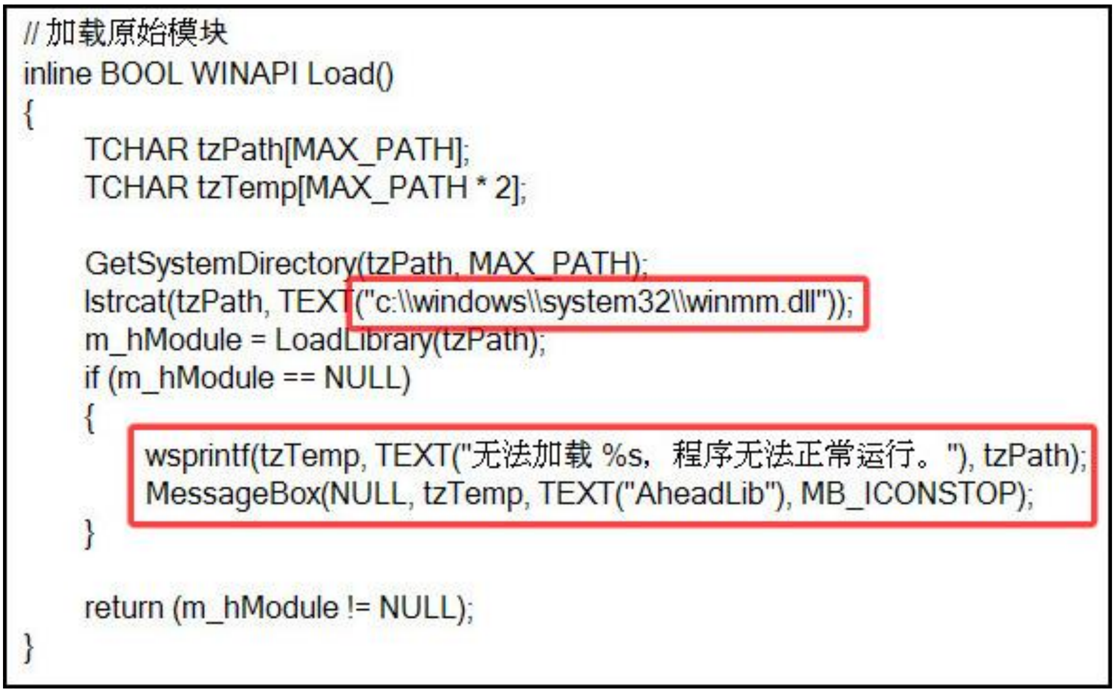
Hook functions (code generated by the legitimate program AheadLib)

The flexibility of this tool allows to customize the logics of malicious application during analysis and overload functions code to provide some debugging output. Some code can be added to display parameters of the hooked functions in order to find out which values are passed to the original functions when they are called. This method is used in so called dynamic analysis of malicious applications.



```
// 获取原始函数地址
FARPROC WINAPI GetAddress(PCSTR pszProcName)
{
    FARPROC fpAddress;
    CHAR szProcName[16];
    TCHAR tzTemp[MAX_PATH];

    if (m_hModule == NULL)
    {
        if (Load() == FALSE)
        {
            ExitProcess(-1);
        }
    }

    fpAddress = GetProcAddress(m_hModule, pszProcName);
    if (fpAddress == NULL)
    {
        if (HIWORD(pszProcName) == 0)
        {
            wsprintf(szProcName, "%d", pszProcName);
            pszProcName = szProcName;
        }

        wsprintf(tzTemp, TEXT("无法找到函数 %hs，程序无法正常运行。"), pszProcName);
        MessageBox(NULL, tzTemp, TEXT("AheadLib"), MB_ICONSTOP);
        ExitProcess(-2);
    }

    return fpAddress;
}
```

Determining the addresses of the real functions
(error message in the frame: “Function %hs cannot be found,
the program will not operate correctly”)

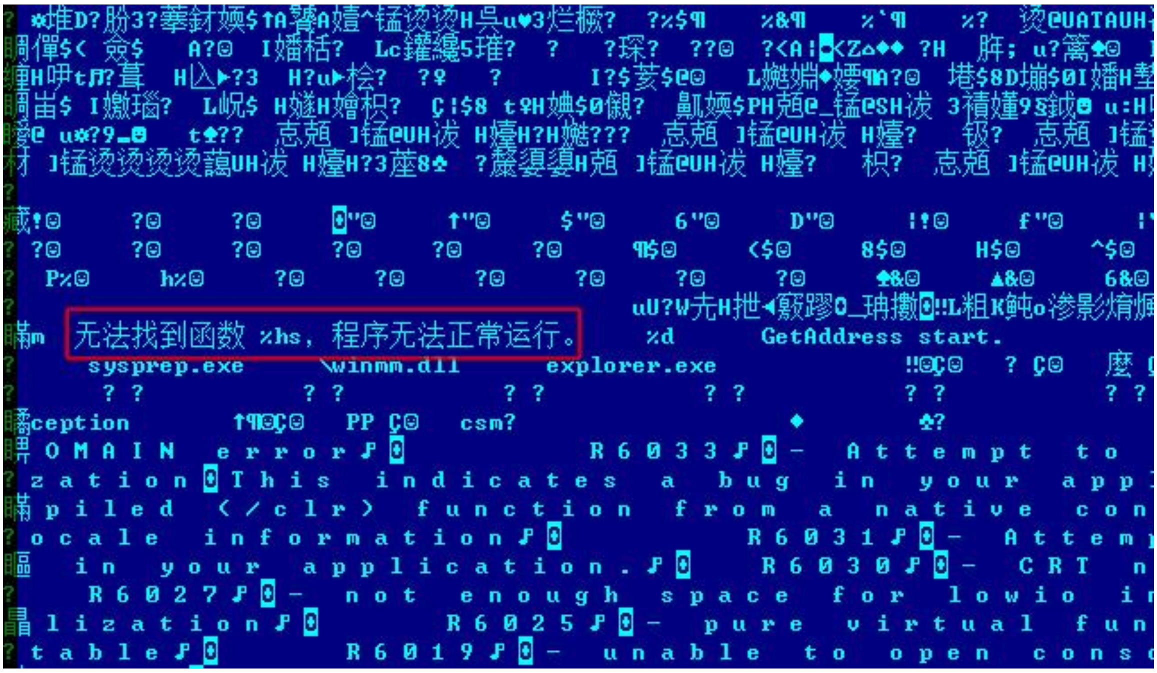


```
// 加载原始模块
inline BOOL WINAPI Load()
{
    TCHAR tzPath[MAX_PATH];
    TCHAR tzTemp[MAX_PATH * 2];

    GetSystemDirectory(tzPath, MAX_PATH);
    lstrcat(tzPath, TEXT("c:\\windows\\system32\\winmm.dll"));
    m_hModule = LoadLibrary(tzPath);
    if (m_hModule == NULL)
    {
        wsprintf(tzTemp, TEXT("无法加载 %s，程序无法正常运行。"), tzPath);
        MessageBox(NULL, tzTemp, TEXT("AheadLib"), MB_ICONSTOP);
    }

    return (m_hModule != NULL);
}
```

Modified module loading the original DLL
(error message in the frame: "%s cannot be loaded, the program will not operate correctly")

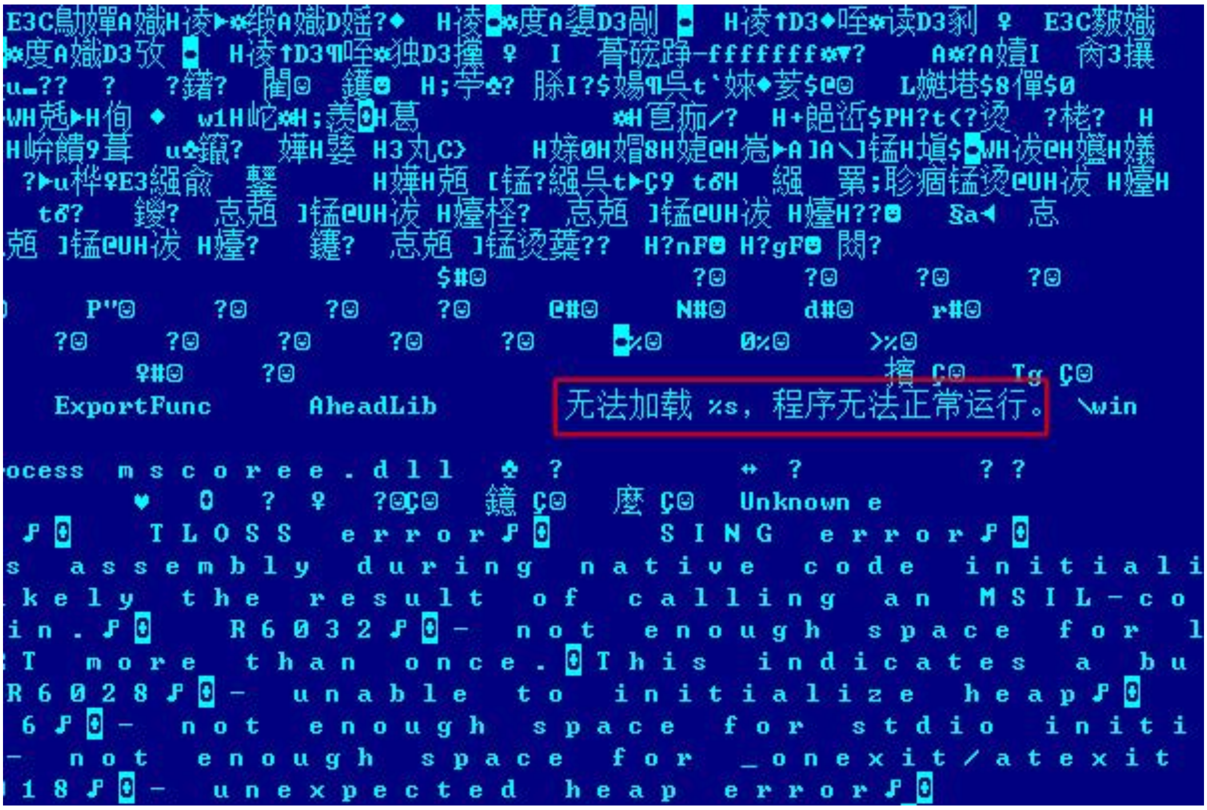
Ironically, the malware authors have found this to be a convenient application for creating malicious proxy-libraries. They specified a system library (winmm.dll) as a parameter for AheadsLib tool and produced a source code template to create a proxy DLL – in the form of C file. By overloading some functions with the malicious payload, the attackers created a complete piece of malware that included all the features of the system DLL.

Strangely, the attackers kept the code for AheadLib debug messages in the early versions of their malware (marked with red in the screenshots above). These strings can also be found in compiled malicious binaries:



```
无法找到函数 %hs，程序无法正常运行。   %d      GetAddress start.
  sysprep.exe      \winmm.dll      explorer.exe
ception        PP        csm?
OMAIN error            R6033 - Attempt to
zation  This indicates a bug in your app
piled (/clr) function from a native con
ocale information          R6031 - Attem
in your application.        R6030 - CRT n
R6027 - not enough space for lowio i
lization          R6025 - pure virtual fun
table          R6019 - unable to open cons
```

The function %hs cannot be found, the program will not operate correctly



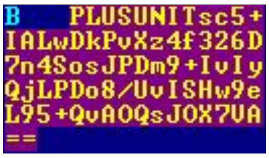
```
E3C勴蠈A孅H渗▶☼缎A孅D媱?◆  H渗◘☼度A嫢D3剮 ◘  H渗↑D3◆哐☼读D3刹 ♀  E3C颏孅
☼度A孅D3弢 ◘  H渗↑D3¶哐☼独D3擸 ♀  I  胔碂踭-fffffff☼▼?    A☼?A嬉I  肏3攘
u▁??  ?  ?鍺?  閵☺  鑳☻  H;苧♠? 脎I?$婸¶呉t`妹◆荽$@☺   L媿堪$8僤$0
WH兘▶H伺 ◆  w1H屹☼H;羡◘H葛           ☼H亘痂/?  H+鹧近$PH?t<?烫  ?栳?  H
H峅饋9葺  u♠鑹?  媶H嬖 H3丸C>    H婇0H媢8H婑@H嶌▶A]A\]锰H媜$◘WH诐@H嬗H嬟
 ?▶u桦♀E3繦兪  鑿      H媶H苑 [锰?繦吳t▶Ç9 tδH  繦  罽;肜痼锰烫@UH诐 H嬗H
  tδ?   鏤?  忠苑 ]锰@UH诐 H嬗柽?  忠苑 ]锰@UH诐 H嬗H??☻   §a◄  忠
苑 ]锰@UH诐 H嬗?    鑳?  忠苑 ]锰烫蕤??  H?nF☻ H?gF☻ 閟?
                                          $#☺         ?☺        ?☺        ?☺        ?☺
     P"☺      ?☺      ?☺      ?☺      @#☺       N#☺       d#☺       r#☺
   ?☺      ?☺      ?☺      ?☺      ?☺      ◘%☺       0%☺       >%☺
        ♀#☺      ?☺                                              擿 Ç☺    Tg Ç☺
   ExportFunc      AheadLib       无法加载 %s, 程序无法正常运行。  \win

ocess  m s c o r e e . d l l   ♠  ?          ↔  ?           ? ?
           ♥   ☺   ?   ♀   ?☺Ç☺   鐿 Ç☺    麼 Ç☺   Unknown e
 ♪ ◘      T L O S S   e r r o r ♪ ◘        S I N G   e r r o r ♪ ◘
s   a s s e m b l y   d u r i n g   n a t i v e   c o d e   i n i t i a l i
 k e l y   t h e   r e s u l t   o f   c a l l i n g   a n   M S I L - c o
i n . ♪ ◘    R 6 0 3 2 ♪ ◘ -   n o t   e n o u g h   s p a c e   f o r   l
 T   m o r e   t h a n   o n c e . ◘ T h i s   i n d i c a t e s   a   b u
R 6 0 2 8 ♪ ◘ -   u n a b l e   t o   i n i t i a l i z e   h e a p ♪ ◘
 6 ♪ ◘ -   n o t   e n o u g h   s p a c e   f o r   s t d i o   i n i t i
-   n o t   e n o u g h   s p a c e   f o r   _ o n e x i t / a t e x i t
 1 8 ♪ ◘ -   u n e x p e c t e d   h e a p   e r r o r ♪ ◘
```

%s cannot be loaded, the program will not operate correctly

Later, these fragments were removed from the C file generated by AheadLib.

Control DLL

The winmm.dll malicious library maintains another library in its body, which is decrypted and loaded into the process memory without creating any files on local disk. According to file version info the original name of this library is "PlusDLL.dll". This is the platform’s main control component. When the additional DLL has been properly allocated in the memory, winmm.dll passes control to it with a parameter – a string which contains bot settings. The settings string, in encrypted form, is also located in the winmm.dll body – after the magic word "PLUSUNIT".



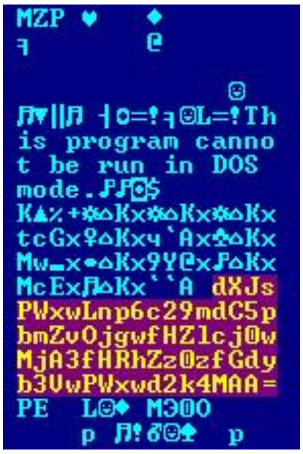
```
B    PLUSUNITsc5+
IALwDkPvXz4f326D
7n4SosJPDm9+IvIy
QjLPDo8/UvISHw9e
L95+QvAOQsJOX7VA
==
```

Encrypted bot settings

After decryption, the string contains the following:

url=lp.gasoft.us:80|ver=1018|tag=33|group=lp80wi

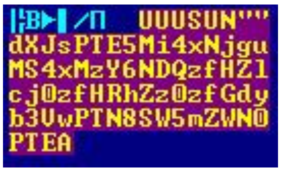
Apparently, when the Winnti malware managed to get into focus of security researchers: the authors made modifications of the methods used to store these initial settings. In some samples, the settings were hidden even in the executable file’s header:



```
MZP ♥    ♦
╕        @

               ☺
╓▼||╓ ┤○=!┐☺L=!Th
is program canno
t be run in DOS
mode.♪♫◙$
K▲%+☼⌂Kx☼⌂Kx☼⌂Kx
tcGx♀⌂Kxч`Ax♠⌂Kx
Mw▬x•⌂Kx9Y@x♪⌂Kx
McEx╓⌂Kx``A dXJs
PWxwLnp6c29mdC5p
bmZvOjgwfHZ1cj0w
MjA3fHRhZz0zfGdy
b3VwPWxwd2k4MAA=
PE   L☺♦ M3OO
     p ╓!δ☺♠   p
```

Encrypted settings in the header of malicious executable

In other variants, the ‘PLUSUNIT’ magic string was modified:



```
|╡B►▌⁄Π   UUUSUN""
dXJsPTE5Mi4xNjgu
MS4xMzY6NDQzfHZ1
cj0zfHRhZz0zfGdy
b3VwPTN8SW5mZWN0
PTEA
```

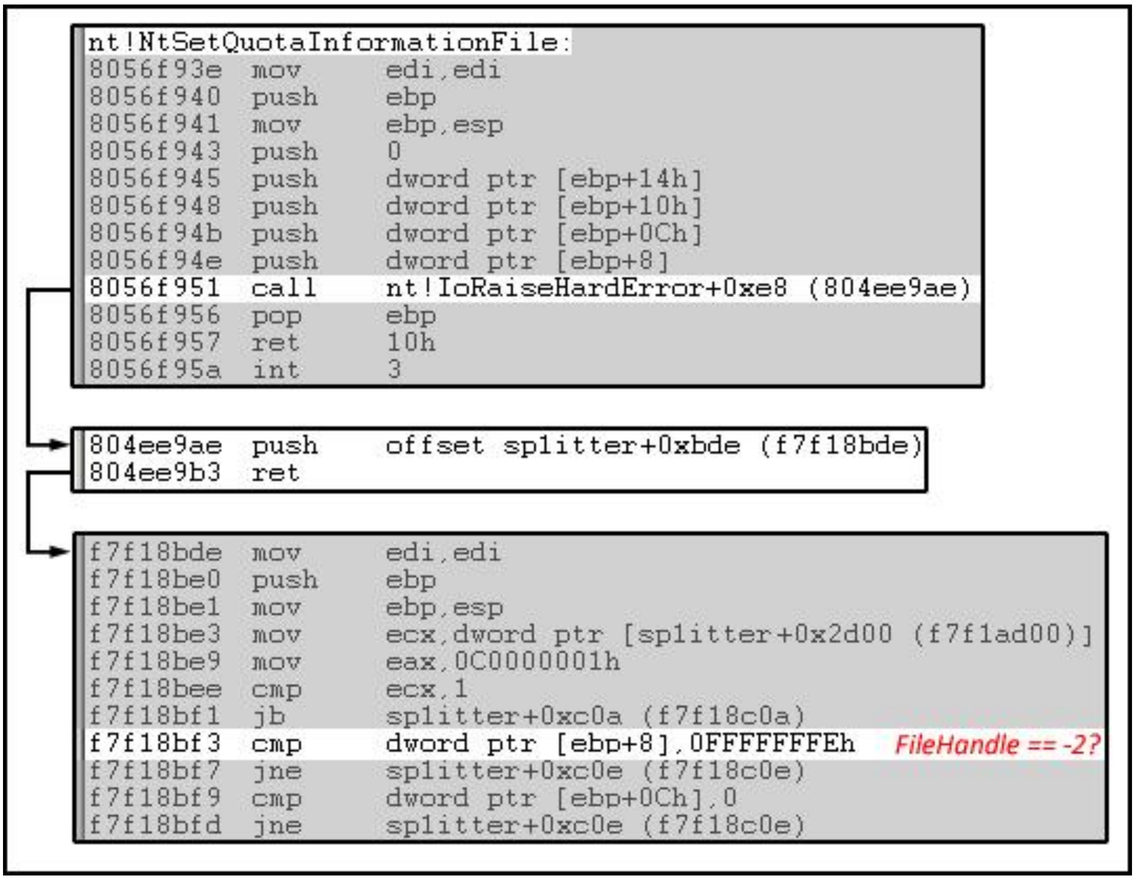
UUUSUN”” instead of PLUSUNIT

The PlusDLL library has an embedded driver. The driver is stored in %WINDIR%\System32\<drivername.sys> file, registered as a service and started by NtLoadDriver system API function. Immediately after that, the driver’s file is removed, as well as all the registry entries created during service registration. The executable preserved the original driver names which are "PortLess" and "PointFilter"; however, the driver files used during infection are saved as "sp1itter.sys" and "acplec.sys".

The purpose of the driver is to hide network connections established by the malware. For example, if the user decides to check a list of established connections (e.g., using the ‘netstat –a’ command or the tcpview program) while the bot is communicating to the control center, the driver will protect and hide the malware connections. This approach is used by many rootkits on the Windows platform.

The driver uses an interesting method to get the list of addresses to protect connections with. This information is available in the PlusDLL control library, which normally operates in the context of the explorer.exe process when the infection is active on the computer. The address information is sent from the user space (from PlusDLL) to the kernel space, where the driver works, via call to NtSetQuotaInformationFile API function.

During initialization, the driver hooks the NtSetQuotaInformationFile function:



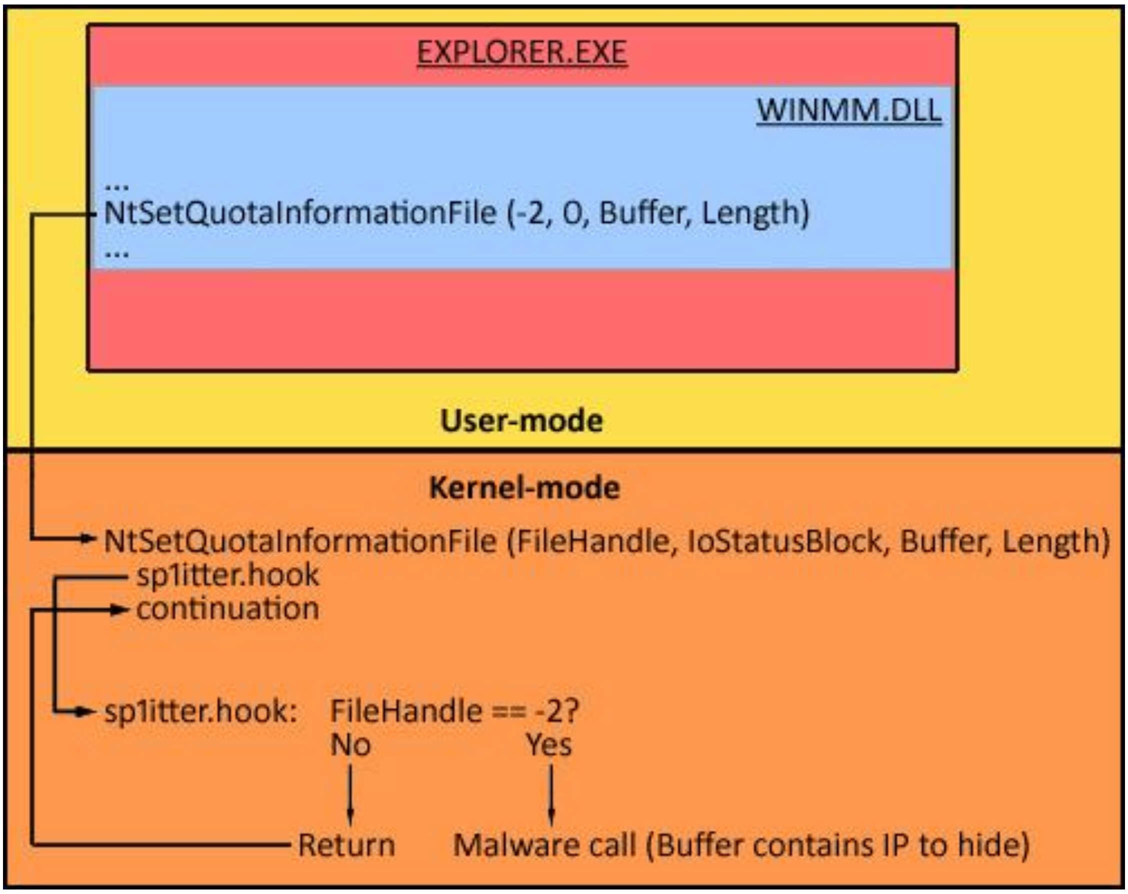
```
nt!NtSetQuotaInformationFile:
8056f93e mov     edi,edi
8056f940 push    ebp
8056f941 mov     ebp,esp
8056f943 push    0
8056f945 push    dword ptr [ebp+14h]
8056f948 push    dword ptr [ebp+10h]
8056f94b push    dword ptr [ebp+0Ch]
8056f94e push    dword ptr [ebp+8]
8056f951 call    nt!IoRaiseHardError+0xe8 (804ee9ae)
8056f956 pop     ebp
8056f957 ret     10h
8056f95a int     3

804ee9ae push    offset splitter+0xbde (f7f18bde)
804ee9b3 ret

f7f18bde mov     edi,edi
f7f18be0 push    ebp
f7f18be1 mov     ebp,esp
f7f18be3 mov     ecx,dword ptr [splitter+0x2d00 (f7f1ad00)]
f7f18be9 mov     eax,0C0000001h
f7f18bee cmp     ecx,1
f7f18bf1 jb      splitter+0xc0a (f7f18c0a)
f7f18bf3 cmp     dword ptr [ebp+8],0FFFFFFFEh   FileHandle == -2?
f7f18bf7 jne     splitter+0xc0e (f7f18c0e)
f7f18bf9 cmp     dword ptr [ebp+0Ch],0
f7f18bfd jne     splitter+0xc0e (f7f18c0e)
```

Hook on NtSetQuotaInformationFile function

Every time the function is called, the driver checks its parameters: to be precise it is HANDLE FileHandle and PVOID Buffer parameters**.**

The FileHandle parameter holds a descriptor of the partition on the hard drive where the function is expected to set disk quotas.

The Buffer parameter is a memory buffer with information of new quotas to be set. The driver checks whether the value of the FileHandle parameter is equal to minus two. When the system calls the NtSetQuotaInformationFile function to actually change the quotas, the descriptor must be associated with one of the disks. Normally such descriptors in the Windows system are positive integers which obviously means that it cannot be equal to minus two. The negative value is set by the PlusDLL library in order to make the driver detect that the NtSetQuotaInformationFile function was called by that library. When calling NtSetQuotaInformationFile, PlusDLL sends information about the network addresses to be protected by the driver via the Buffer parameter. If FileHandle is not equal to minus two, the hook function in the driver passes control to system's original code of NtSetQuoataInformationFile API function and everyhin works as it should be on an uninfected system.

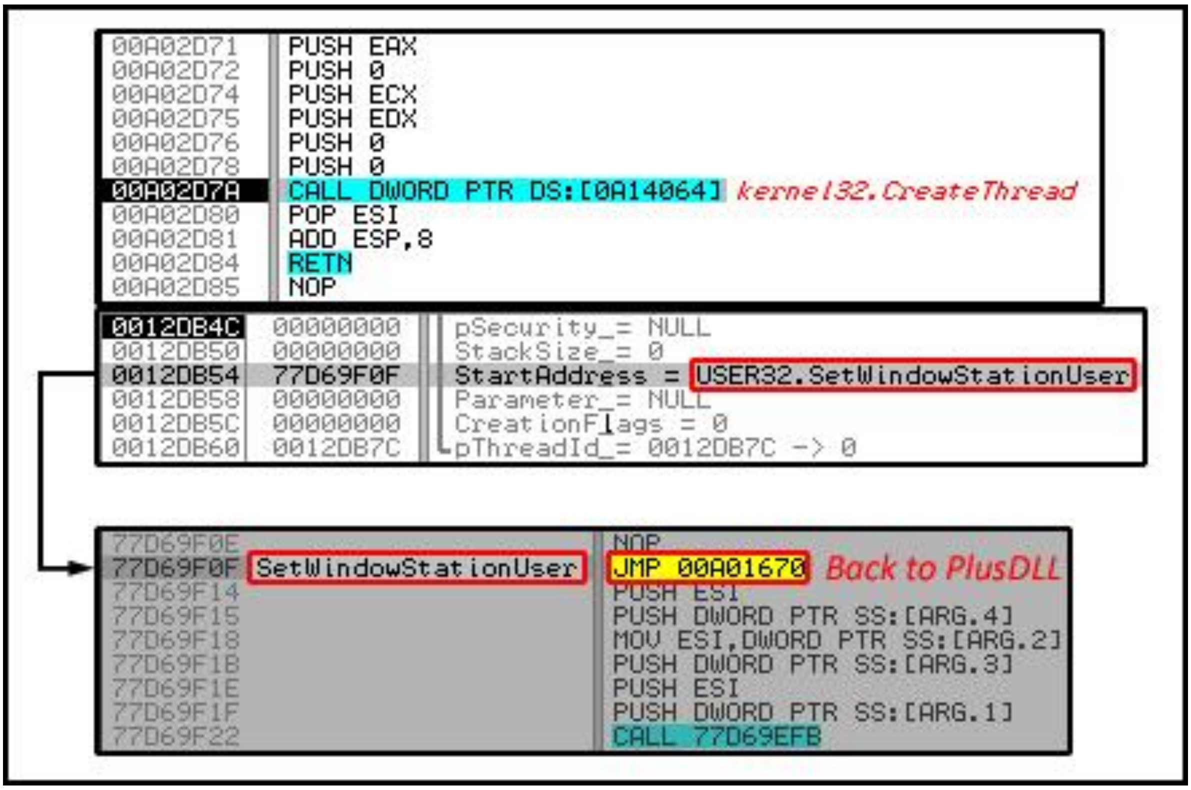

EXPLORER.EXE
WINMM.DLL
...
NtSetQuotaInformationFile (-2, 0, Buffer, Length)
...
User-mode
Kernel-mode
NtSetQuotaInformationFile (FileHandle, IoStatusBlock, Buffer, Length)
sp1itter.hook
continuation
sp1itter.hook: FileHandle == -2?
No
Yes
Return
Malware call (Buffer contains IP to hide)

Sending data from the PlusDLL.dll library to the sp1itter.sys driver

Note that 64-bit versions of Windows do not allow unsigned drivers to run. The malicious driver's 64-bit versions were signed using stolen certificates. During the time that we have been tracking the Winnti group, we found 11 certificates that were used to sign the malware used by the group (not necessarily drivers only). Ten of them belong to various companies in the gaming industry.

Launching the main function

As mentioned above, the PlusDLL library is a control module. Let's look at how the cybercriminals implemented the transition to perform the malicious DLL's main tasks. They could have simply called an appropriate function directly or created a separate thread in which to execute it, but for some reason they resorted to a trick: the code of the SetWindowStationUser function in the user32.dll library was modified. After modification, the function's first command became jmp <addr>, where <addr> is the address of the function in the PlusDLL library which implements the malicious library's main features.



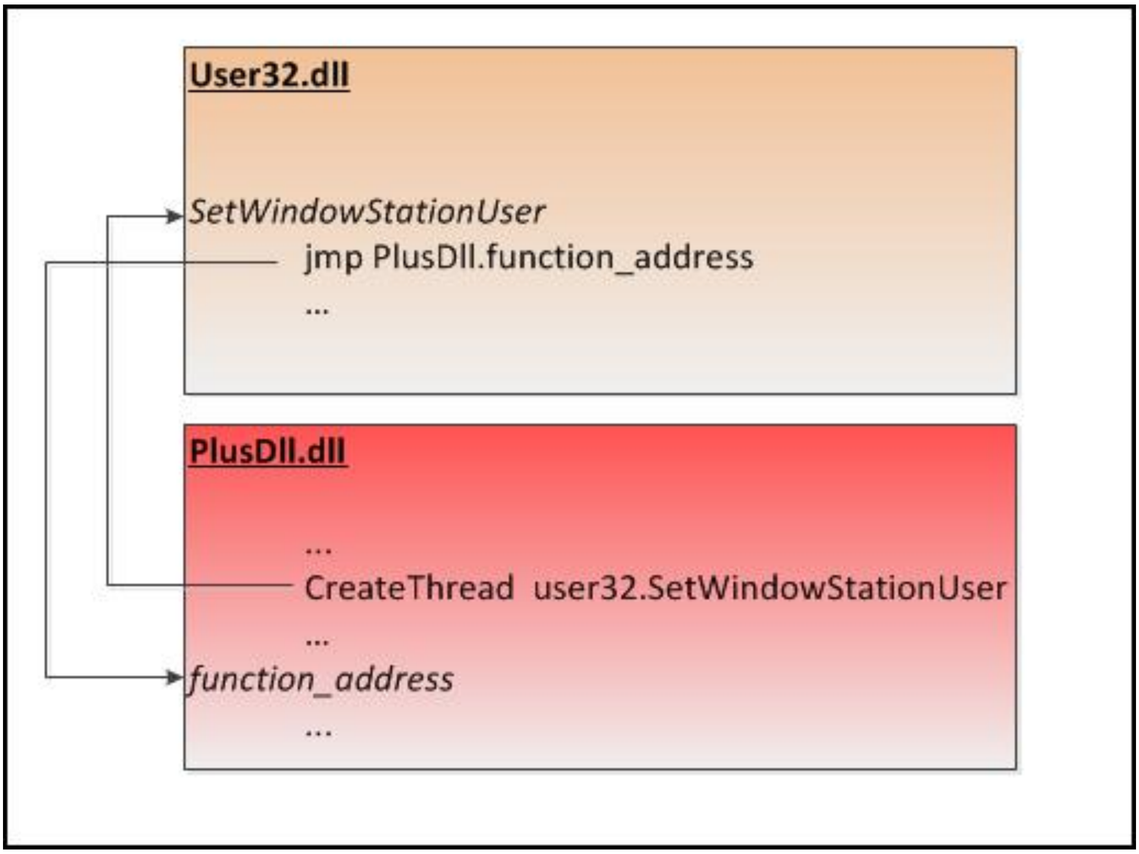
```
00A02D71    PUSH EAX
00A02D72    PUSH 0
00A02D74    PUSH ECX
00A02D75    PUSH EDX
00A02D76    PUSH 0
00A02D78    PUSH 0
00A02D7A    CALL DWORD PTR DS:[0A14064]  kernel32.CreateThread
00A02D80    POP ESI
00A02D81    ADD ESP,8
00A02D84    RETN
00A02D85    NOP
```

```
0012DB4C  00000000   pSecurity_= NULL
0012DB50  00000000   StackSize_= 0
0012DB54  77D69F0F   StartAddress = USER32.SetWindowStationUser
0012DB58  00000000   Parameter_= NULL
0012DB5C  00000000   CreationFlags = 0
0012DB60  0012DB7C   pThreadId_= 0012DB7C -> 0
```

```
77D69F0E                          NOP
77D69F0F  SetWindowStationUser    JMP 00A01670   Back to PlusDLL
77D69F14                          PUSH ESI
77D69F15                          PUSH DWORD PTR SS:[ARG.4]
77D69F18                          MOV ESI,DWORD PTR SS:[ARG.2]
77D69F1B                          PUSH DWORD PTR SS:[ARG.3]
77D69F1E                          PUSH ESI
77D69F1F                          PUSH DWORD PTR SS:[ARG.1]
77D69F22                          CALL 77D69EFB
```

Hook on SetWindowStationUser

Immediately after this modification, a thread is created (CreateThread) executing code starting from the SetWindowStationUser function address. As a result, when control is eventually passed to this function, the inserted command jmp <addr> returns control back to the PlusDLL code.



Malicious DLL launching its own code by creating a thread that supposedly calls SetWindowStationUser

The same method is used to execute two more functions in the PlusDLL library. One of them is used to initialize network routines; the other executes procedures terminating the malicious program at the very end. The only difference is that instead of SetWindowStationUser, the code of two other functions from user32.dll is modified – EndTask and WinHelpW, respectively.

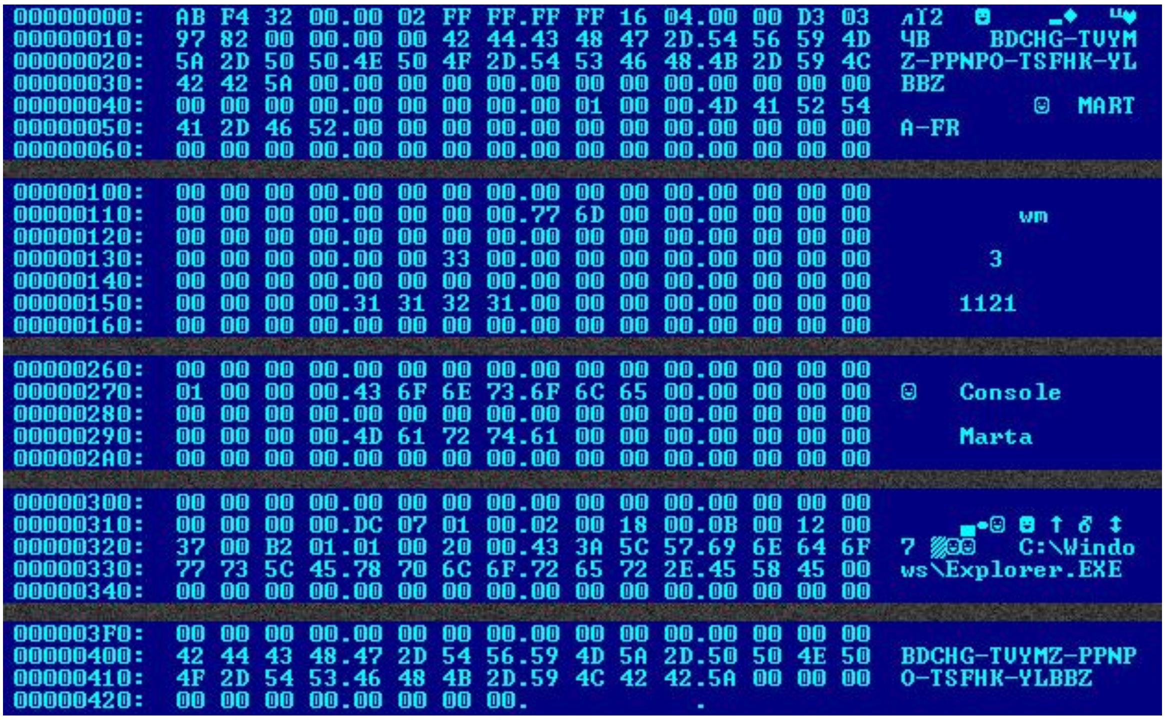
It is likely that this was done in order to hide the real addresses of functions in PlusDLL in case its code was analyzed based on its execution logs using an automatic system (sandbox) that looks at all function calls. If this trick is used, an execution log would only show threads launched from the addresses of the functions SetWindowStationUser, EndTask and WinHelpW, which could potentially confuse researchers.

Another possibility is that this is an anti-emulation feature. Perhaps the emulators built into some anti-virus products are unable to cope with these 'leaps' – in this case, emulation will not result in the execution of malicious functions, which also suits the cybercriminals' purposes.

Target Functionality

So what does PlusDLL control? It turns out that the target functionality is implemented in different files. Each file provides a specific remote control feature and is downloaded from the attackers' server every time the system starts up. These files are not saved on disk or in the registry but are loaded directly into the memory.

At the very start of the operation, after launching the driver, PlusDLL collects information about the infected system. A unique identifier for the infected computer is generated based on information about the hard drive and the network adapter's MAC address, e.g., TKVFP-XZTTL-KXFWH-RBJLF-FXWJR. The attackers are interested primarily in the computer's name, the program which loaded the malicious library, as well as information about remote desktop sessions (session name, client name, user name and session time). All of this data is collected in a buffer, which is then compressed and sent to the attackers' control center. The buffer may look like this:



```
00000000:  AB F4 32 00.00 02 FF FF.FF FF 16 04.00 00 D3 03  лЇ2  ☻      ▬♦   ╚♥
00000010:  97 82 00 00.00 00 42 44.43 48 47 2D.54 56 59 4D  ЧВ     BDCHG-TVYM
00000020:  5A 2D 50 50.4E 50 4F 2D.54 53 46 48.4B 2D 59 4C  Z-PPNPO-TSFHK-YL
00000030:  42 42 5A 00.00 00 00 00.00 00 00 00.00 00 00 00  BBZ
00000040:  00 00 00 00.00 00 00 00.00 01 00 00.4D 41 52 54          ☺  MART
00000050:  41 2D 46 52.00 00 00 00.00 00 00 00.00 00 00 00  A-FR
00000060:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00

00000100:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000110:  00 00 00 00.00 00 00 00.77 6D 00 00.00 00 00 00         wm
00000120:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000130:  00 00 00 00.00 00 33 00.00 00 00 00.00 00 00 00       3
00000140:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000150:  00 00 00 00.31 31 32 31.00 00 00 00.00 00 00 00      1121
00000160:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00

00000260:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000270:  01 00 00 00.43 6F 6E 73.6F 6C 65 00.00 00 00 00  ☺   Console
00000280:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000290:  00 00 00 00.4D 61 72 74.61 00 00 00.00 00 00 00      Marta
000002A0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00

00000300:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000310:  00 00 00 00.DC 07 01 00.02 00 18 00.0B 00 12 00      ▄•☺ ☻ ↑ ♂ ↕
00000320:  37 00 B2 01.01 00 20 00.43 3A 5C 57.69 6E 64 6F  7 ▓☺☺    C:\Windo
00000330:  77 73 5C 45.78 70 6C 6F.72 65 72 2E.45 58 45 00  ws\Explorer.EXE
00000340:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00

000003F0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000400:  42 44 43 48.47 2D 54 56.59 4D 5A 2D.50 50 4E 50  BDCHG-TVYMZ-PPNP
00000410:  4F 2D 54 53.46 48 4B 2D.59 4C 42 42.5A 00 00 00  O-TSFHK-YLBBZ
00000420:  00 00 00 00.00 00 00 00.
```

The bot sends information about an infected system to the control center

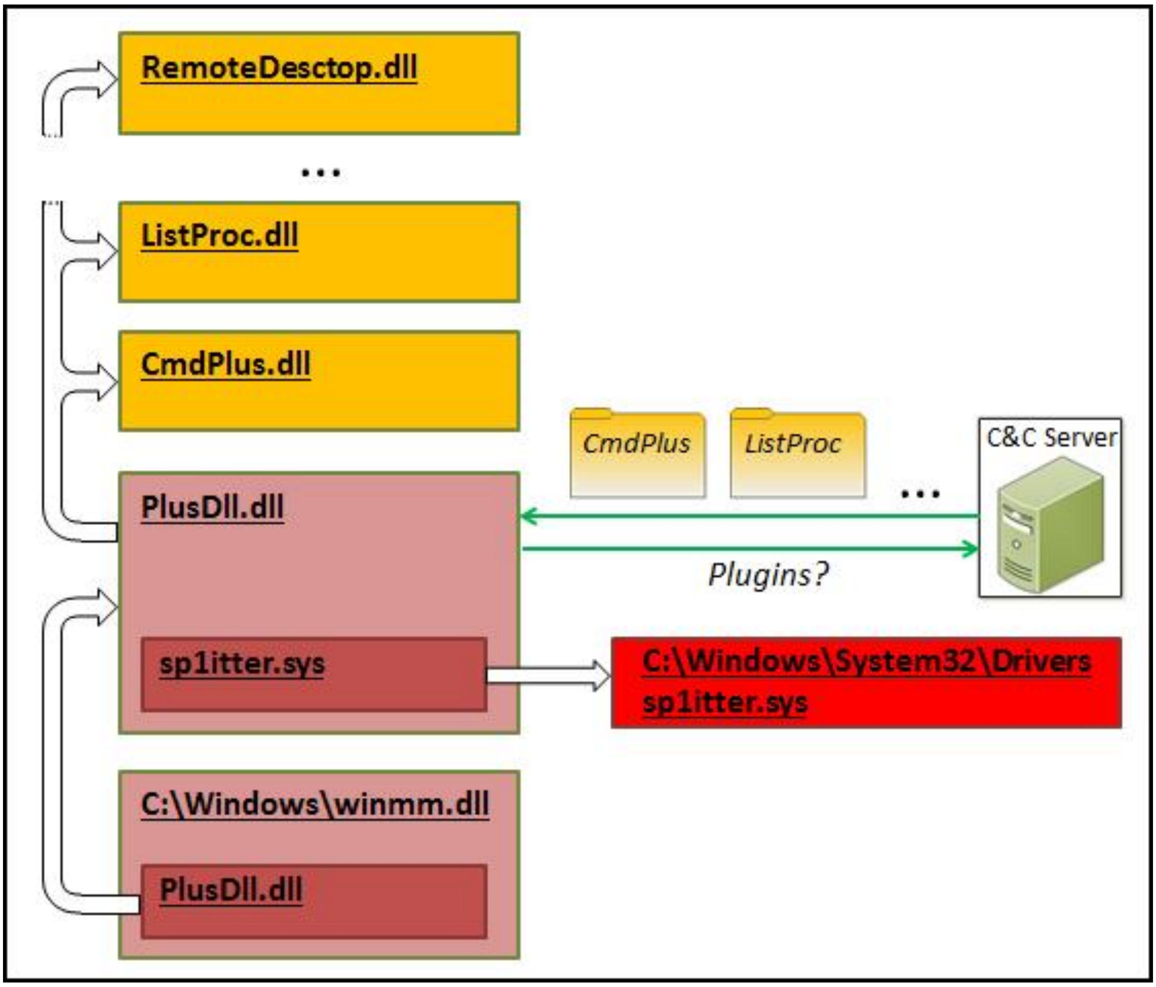
In reply to this initial message from the bot, the control center sends the list of available plugins. Plugins are DLL libraries that provide specific remote control functions. Upon receiving the list of plugins, the bot downloads them, allocates them in the memory and passes control to these libraries.

Different C2 servers could push different plugins. In total we have discovered eight functional libraries:

Plugin Name	Plugin Purpose
CmdPlus	Provide access to the system command line.
ListFileManager	Provide access to the file system: list directory contents, manipulate files.
ListProc	List or kill running processes.
ListService	List system services.
PortMap	Redirect traffic using port forwarding.
RemoteDesktop	Enable Remote Desktop service on the infected machine.
Socks5Client	Library for transferring data over the network using a SOCKS5 proxy server.
TransPlus	Enables the attacker to transfer files: receive files from the infected machine, download/create/save files, as well as execute programs on the infected computer.

These plugins form the core toolkit which is used by the perpetrators during attack.

Operation of the malicious platform



Operation flowchart at the initial stage

As you can see, the cybercriminals use an entire inventory of malicious tools to effectively control the remote computer. Moreover, they have taken measures to conceal their activities: the plugins do not explicitly appear anywhere except in the computer's memory; they do not get saved to the hard drive; the driver is deleted immediately after launch; all traces in the registry that could indicate this launch get deleted. Only the initial DLL remains on the disk that kick starts the entire process and contains an encrypted version of PlusDLL which is the control DLL.

One of the weak points in this architecture is that the driver does get saved to the hard drive before it launches, so anti-virus products can detect the emergence of this file. The situation is further exacerbated by the fact that the malicious drivers may be signed (although not all drivers in the Winnti samples that we detected were in fact signed). An unsigned driver in itself does not have the means to counter antivirus products and its code can be easily recognized as malicious, whereas signed drivers stand a better chance of remaining undetected by antivirus products: certain anti-virus products consider properly signed programs legitimate by default, so as to minimize the chances of false positive responses.

Kaspersky Lab’s products detect the malicious programs described above under the following verdicts:

The initial DLLs winmm.dll and apphelp.dll, the PlusDll.dll control DLLs, and functional loadable modules (CmdPlus.dll etc.) are detected as Backdoor.Win32.Winnti or Backdoor.Win64.Winnti.

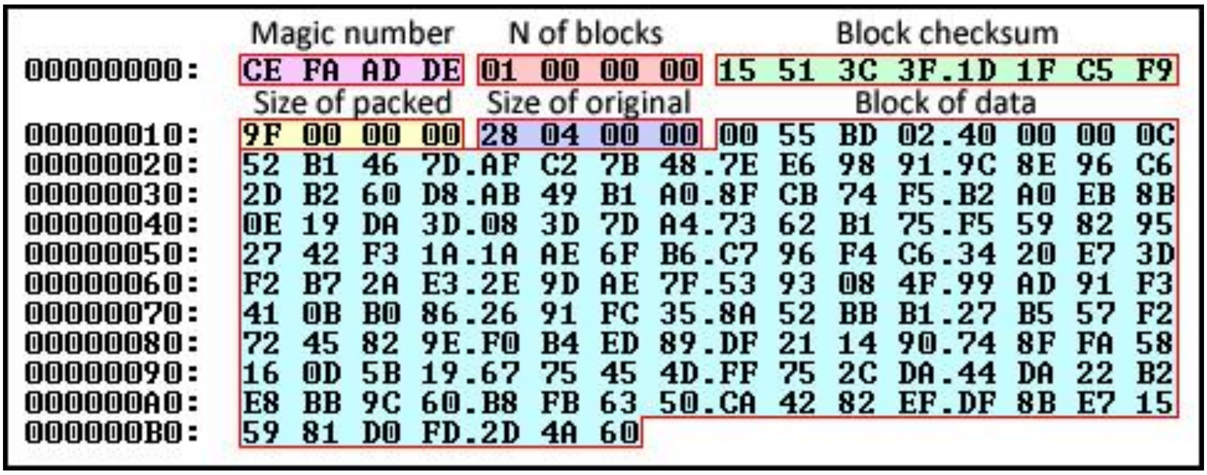
The drivers sp1itter.sys and acplec.sys are detected as Rootkit.Win32.Winnti or Rootkit.Win64.Winnti.

Communication with the C&C Server

The data transmitted during the communication between the bot and the C&C server, naturally, do not manifest themselves in any explicit form in online data traffic. Since an active remote control practice can generate substantial traffic, cybercriminals compress communication data with the algorithm LZMA, though they do not include the appropriate header inherent to this algorithm.

The data is transmitted over the TCP protocol. The samples that we analyzed established connections between C&C servers and ports 53, 80 and 443. This port selection is not surprising: they are associated with the protocols DNS, HTTP and HTTPS respectively. All three are routinely used in everyday operations, so they are enabled under most firewall policies. Besides, large amounts of data typically pass through these ports (with the possible exception of port 53), which makes it easier for the malicious traffic to remain inconspicuous.

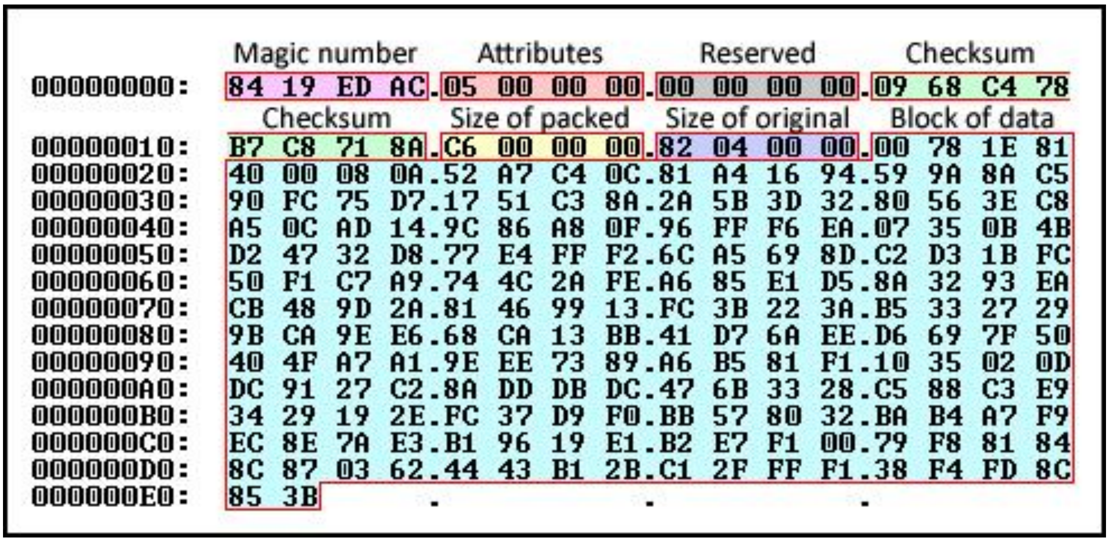
Although the ports are associated with certain protocols, the actual content of the traffic generated by the malicious program does not correspond to them. Early versions of the Winnti platform exhibited the following traffic structure when communicating with C&C: each block of transmitted data started with the magic number 0xdeadface, followed by the number of blocks (in a DWORD), then the hash of the transmitted block (8 bytes), the size of compressed data (DWORD), the size of source data (DWORD) and, finally, the actual compressed data.



```
00000000: CE FA AD DE 01 00 00 00 15 51 3C 3F.1D 1F C5 F9
00000010: 9F 00 00 00 28 04 00 00 00 55 BD 02.40 00 00 0C
00000020: 52 B1 46 7D.AF C2 7B 48.7E E6 98 91.9C 8E 96 C6
00000030: 2D B2 60 D8.AB 49 B1 A0.8F CB 74 F5.B2 A0 EB 8B
00000040: 0E 19 DA 3D.08 3D 7D A4.73 62 B1 75.F5 59 82 95
00000050: 27 42 F3 1A.1A AE 6F B6.C7 96 F4 C6.34 20 E7 3D
00000060: F2 B7 2A E3.2E 9D AE 7F.53 93 08 4F.99 AD 91 F3
00000070: 41 0B B0 86.26 91 FC 35.8A 52 BB B1.27 B5 57 F2
00000080: 72 45 82 9E.F0 B4 ED 89.DF 21 14 90.74 8F FA 58
00000090: 16 0D 5B 19.67 75 45 4D.FF 75 2C DA.44 DA 22 B2
000000A0: E8 BB 9C 60.B8 FB 63 50.CA 42 82 EF.DF 8B E7 15
000000B0: 59 81 D0 FD.2D 4A 60
```

The unit structure of a data block transmitted online in early versions of Winnti

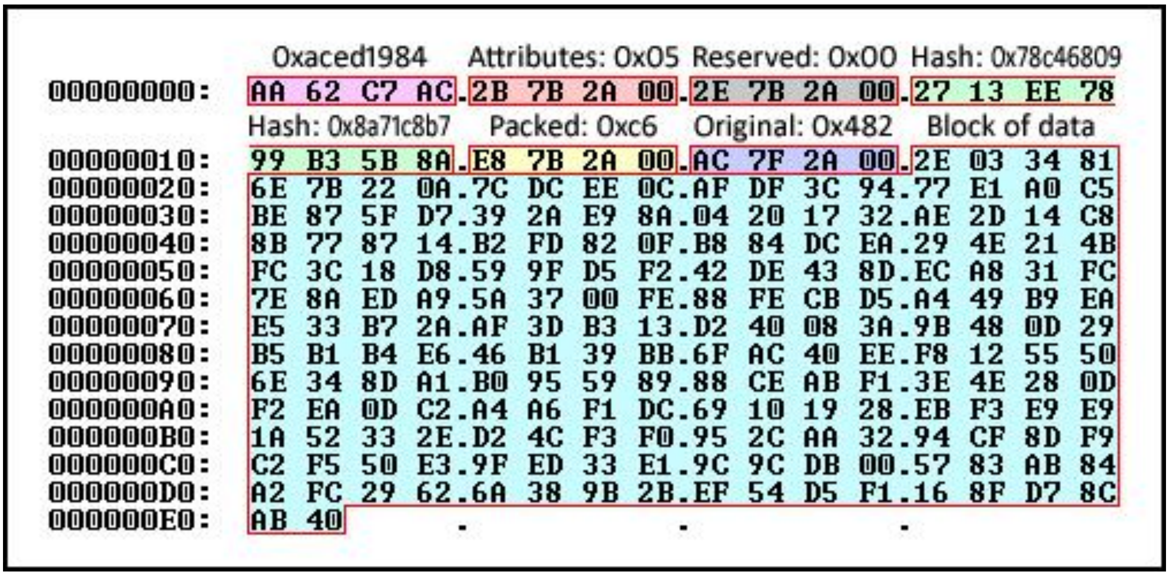
This is where another weak point of the Winnti family of backdoors becomes apparent. With this data structure, malicious network traffic could easily be spotted by, for example, the magic number 0xdeadface. The cybercriminals probably lost control over victim computers fairly frequently as corporate system administrators identified the intrusion by the unique headers in data packets with the help of IDS/IPS systems, and cleaned their networks. In 2011, new versions of Winnti backdoors appeared that, while still based on the same platform, started to use an updated protocol which included extra encryption to communicate with C&C, so the transmitted data no longer had static marks in them. Prior to encryption, the data has the following structure (very similar to the earlier format): the first 4 bytes are taken by the magic number 0xaced1984, then a DWORD of data packet description, the next DWORD carries a zero value, 8 bytes of the hash of the transmitted block, then a DWORD with the size of the compressed data, a DWORD with the size of the source data and then the actual compressed data:



```
00000000: 84 19 ED AC.05 00 00 00.00 00 00 00.09 68 C4 78
00000010: B7 C8 71 8A.C6 00 00 00.82 04 00 00.00 78 1E 81
00000020: 40 00 08 0A.52 A7 C4 0C.81 A4 16 94.59 9A 8A C5
00000030: 90 FC 75 D7.17 51 C3 8A.2A 5B 3D 32.80 56 3E C8
00000040: A5 0C AD 14.9C 86 A8 0F.96 FF F6 EA.07 35 0B 4B
00000050: D2 47 32 D8.77 E4 FF F2.6C A5 69 8D.C2 D3 1B FC
00000060: 50 F1 C7 A9.74 4C 2A FE.A6 85 E1 D5.8A 32 93 EA
00000070: CB 48 9D 2A.81 46 99 13.FC 3B 22 3A.B5 33 27 29
00000080: 9B CA 9E E6.68 CA 13 BB.41 D7 6A EE.D6 69 7F 50
00000090: 40 4F A7 A1.9E EE 73 89.A6 B5 81 F1.10 35 02 0D
000000A0: DC 91 27 C2.8A DD DB DC.47 6B 33 28.C5 88 C3 E9
000000B0: 34 29 19 2E.FC 37 D9 F0.BB 57 80 32.BA B4 A7 F9
000000C0: EC 8E 7A E3.B1 96 19 E1.B2 E7 F1 00.79 F8 81 84
000000D0: 8C 87 03 62.44 43 B1 2B.C1 2F FF F1.38 F4 FD 8C
000000E0: 85 3B       .           .           .
```

The unit structure of a data block transmitted online in newer versions of Winnti

Then the data is encrypted with regular XOR with a random DWORD size value, and in this form transmitted to the C&C. Knowing that the first four bytes in the source data must represent the value 0xaced1984, it is easy to restore the key for the XOR operation when the data were encrypted. This is how the above data (the XOR value was 0x002a7b2e) looked when it was intercepted in network traffic:



```
00000000: AA 62 C7 AC.2B 7B 2A 00.2E 7B 2A 00.27 13 EE 78
00000010: 99 B3 5B 8A.E8 7B 2A 00.AC 7F 2A 00.2E 03 34 81
00000020: 6E 7B 22 0A.7C DC EE 0C.AF DF 3C 94.77 E1 A0 C5
00000030: BE 87 5F D7.39 2A E9 8A.04 20 17 32.AE 2D 14 C8
00000040: 8B 77 87 14.B2 FD 82 0F.B8 84 DC EA.29 4E 21 4B
00000050: FC 3C 18 D8.59 9F D5 F2.42 DE 43 8D.EC A8 31 FC
00000060: 7E 8A ED A9.5A 37 00 FE.88 FE CB D5.A4 49 B9 EA
00000070: E5 33 B7 2A.AF 3D B3 13.D2 40 08 3A.9B 48 0D 29
00000080: B5 B1 B4 E6.46 B1 39 BB.6F AC 40 EE.F8 12 55 50
00000090: 6E 34 8D A1.B0 95 59 89.88 CE AB F1.3E 4E 28 0D
000000A0: F2 EA 0D C2.A4 A6 F1 DC.69 10 19 28.EB F3 E9 E9
000000B0: 1A 52 33 2E.D2 4C F3 F0.95 2C AA 32.94 CF 8D F9
000000C0: C2 F5 50 E3.9F ED 33 E1.9C 9C DB 00.57 83 AB 84
000000D0: A2 FC 29 62.6A 38 9B 2B.EF 54 D5 F1.16 8F D7 8C
000000E0: AB 40       .           .           .
```

Encrypted data block transmitted online, in the newer versions of Winnti

Since the encryption key (the value with which the source data are encrypted with the XOR operation) is different each time a fragment of data is transmitted, no more static unique labels can be found in the network traffic which would quickly identify the transmitted data as belonging to the Winnti backdoor. Employing this fast, basic method, the cybercriminals have made it much harder to expose their programs' traffic.

Whichever protocol is used (with or without extra encryption), the workflow of communication between the bot and the C&C stays the same at the initial stage of operation:

- The bot sends the first data block, thus signaling itself;
- In response, the C&C sends back the list of available plugins
- The bot starts to download plugins, sending one request at a time to download each plugin
- The C&C sends the requested plugin
- The bot sends a message that the plugin has arrived.

We should note here that, to expedite data downloading, the creators of this platform have quite skillfully implemented asynchronous data transmission in their protocol. For instance, the message that the bot has received the first plugin may only arrive at the C&C when nearly all the plugins have been already sent to the bot.

Having downloaded the malicious payload, the bot deploys the plugins in the memory and initializes them. Now it’s all set for complete remote control over the victim computer, and the bot switches to standby mode, waiting for the operator to connect and maintaining communication with the C&C by sending “empty” messages every 15 seconds or so.

Apart from supplying the plugins, no more automatic actions are performed by the C&C: all of the work to examine the infected computers is done manually by the attackers.

Real Case Investigation (Winnti 2.0)

Please note, that the following is published with approval from one of the attacked companies which preferred to remain anonymous. The real company name was replaced with "CompanyXYZ" or simply "XYZ".

On 21st September 2012, a Security Officer of CompanyXYZ contacted Kaspersky Lab and reported a cyber-attack incident. Anomalous activity was spotted at one of the corporate servers. One of the employees noticed a suspicious directory on the server which was created under his account. The folder had a large archived file with information that was regarded as company's intellectual property.

The anomalies were also confirmed in the network traffic by monitoring software. Several suspicious network connections were established from several computer systems, including network domain controllers, to IP addresses which were not associated with any corporate resources or any other known trusted networks.

The suspicious connections were established on ports 443 and 53. Below is the list of reported IP addresses:

211.60.126.164 (Seoul, South Korea)
113.196.70.169 (Taipei Taiwan)

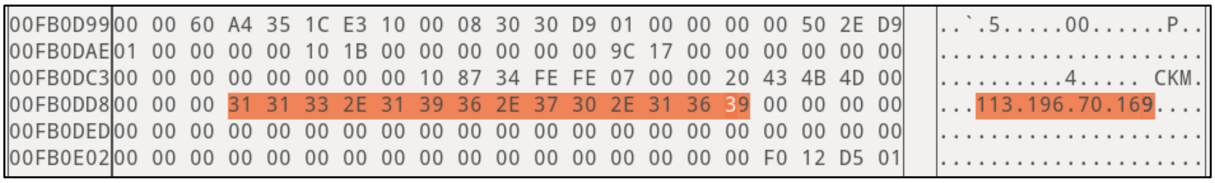
The security officer at CompanyXYZ did an on-site analysis and managed to locate the process which initiated the suspicious connections using SysInternals Process Explorer tool. The connections were initiated by a system process (svchost.exe). A full process dump using Process Explorer was made and shared with Kaspersky Lab. Our team immediately started searching for malware in the provided process dump.

A next day, one more dump of svchost.exe from another presumably infected machine was provided.

We also received an IP address and port that was spotted in the suspicious connections coming from infected machines: 188.120.246.88:80 (Russia).

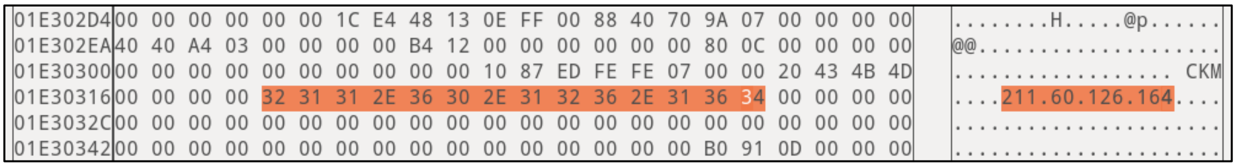
First Step Analysis

Quick search through the dumped processes revealed IP addresses mentioned by the company's security officers.



```
00FB0D99 00 00 60 A4 35 1C E3 10 00 08 30 30 D9 01 00 00 00 00 50 2E D9   ..`.5.....00......P..
00FB0DAE 01 00 00 00 00 10 1B 00 00 00 00 00 00 9C 17 00 00 00 00 00 00   .....................
00FB0DC3 00 00 00 00 00 00 00 00 10 87 34 FE FE 07 00 00 20 43 4B 4D 00   ..........4..... CKM.
00FB0DD8 00 00 00 31 31 33 2E 31 39 36 2E 37 30 2E 31 36 39 00 00 00 00   ...113.196.70.169....
00FB0DED 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   .....................
00FB0E02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F0 12 D5 01   .....................
```

Suspected malicious IP address in svchost.exe memory of Machine #1.



```
01E302D4 00 00 00 00 00 00 1C E4 48 13 0E FF 00 88 40 70 9A 07 00 00 00 00   ........H.....@p......
01E302EA 40 40 A4 03 00 00 00 00 B4 12 00 00 00 00 00 00 80 0C 00 00 00 00   @@....................
01E30300 00 00 00 00 00 00 00 00 00 00 10 87 ED FE FE 07 00 00 20 43 4B 4D   .................. CKM
01E30316 00 00 00 00 32 31 31 2E 36 30 2E 31 32 36 2E 31 36 34 00 00 00 00   ....211.60.126.164....
01E3032C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ......................
01E30342 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 91 0D 00 00 00   ......................
```

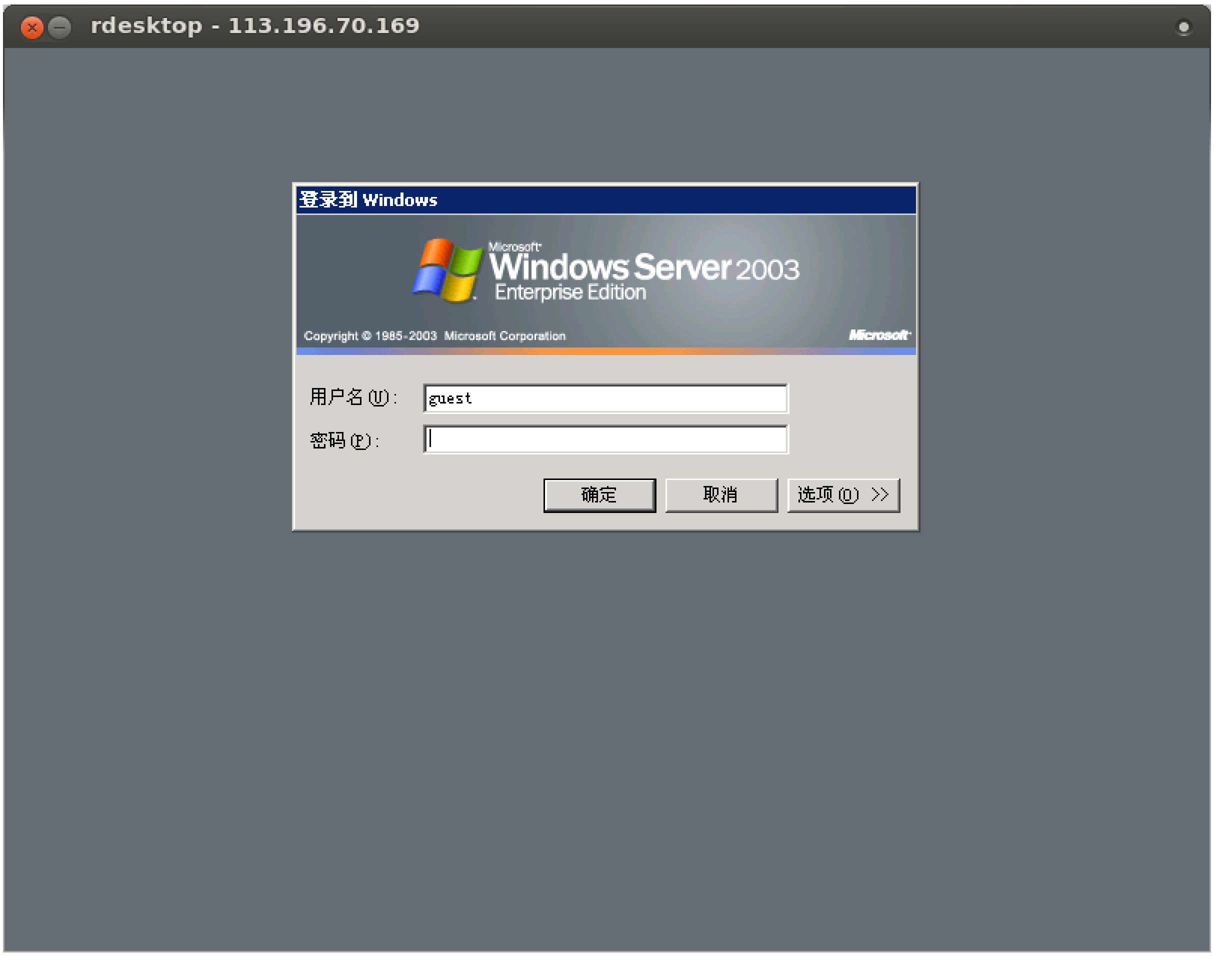
Suspected malicious IP address in svchost.exe memory of Machine #2.

We checked memory around location of the IP address and found no signs of executable code. The memory was most likely dynamically allocated on process heap and used as a temporary storage of resolved domain name. That is why we had to find another indicator of malicious module related to those IP addresses.

We initiated a port scan of the suspected hosts in parallel to memory analysis. Below is the result on the time of scanning:

```
Nmap scan report for 113.196.70.169
Host is up (0.29s latency).
Not shown: 997 filtered ports
PORT     STATE  SERVICE       VERSION
21/tcp   open   ftp           Xlight ftpd 2.0
80/tcp   closed http
3389/tcp open   ms-wbt-server Microsoft Terminal Service
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

The server was running Windows Terminal Service or was used as a proxy linked to some Terminal Server. Establishing connection via RDP client usually reveals default system locale which is used on welcome screen. Below is what we found upon connection:



rdesktop - 113.196.70.169

登录到 Windows

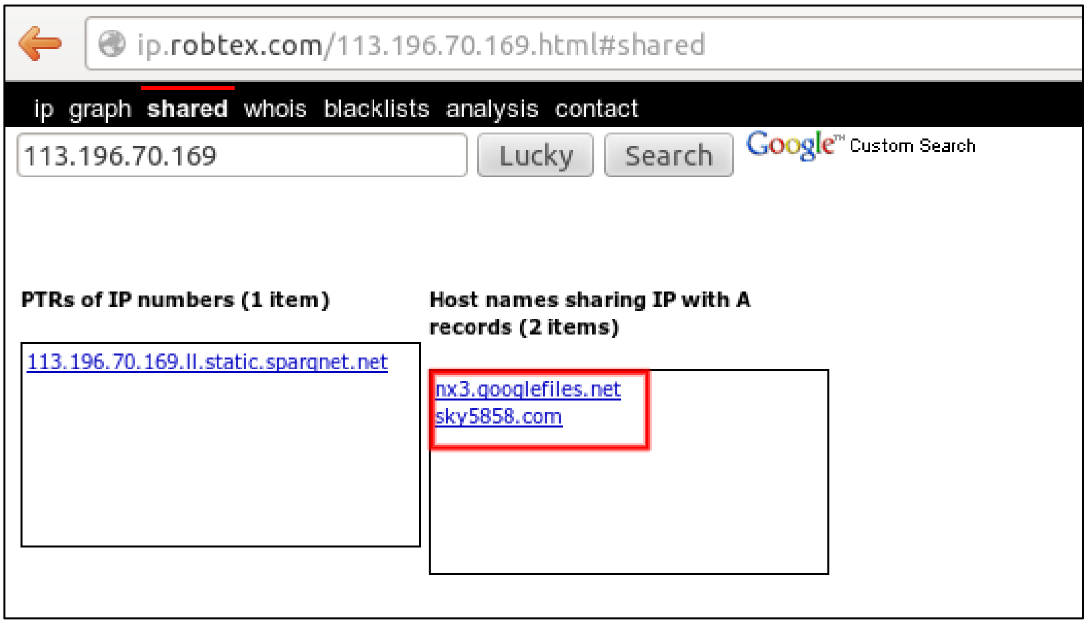
用户名(U): guest

密码(P):

Microsoft Windows Server 2003
Enterprise Edition
Copyright © 1985-2003 Microsoft Corporation
Microsoft
确定
取消
选项(O) >>

Chinese locale on terminal server welcome screen at 113.196.70.169

Checking one of IP addresses on robtex.com brought two possible domain names:


ip.robtex.com/113.196.70.169.html#shared
ip graph shared whois blacklists analysis contact
113.196.70.169
Lucky
Search
Google™ Custom Search

PTRs of IP numbers (1 item)

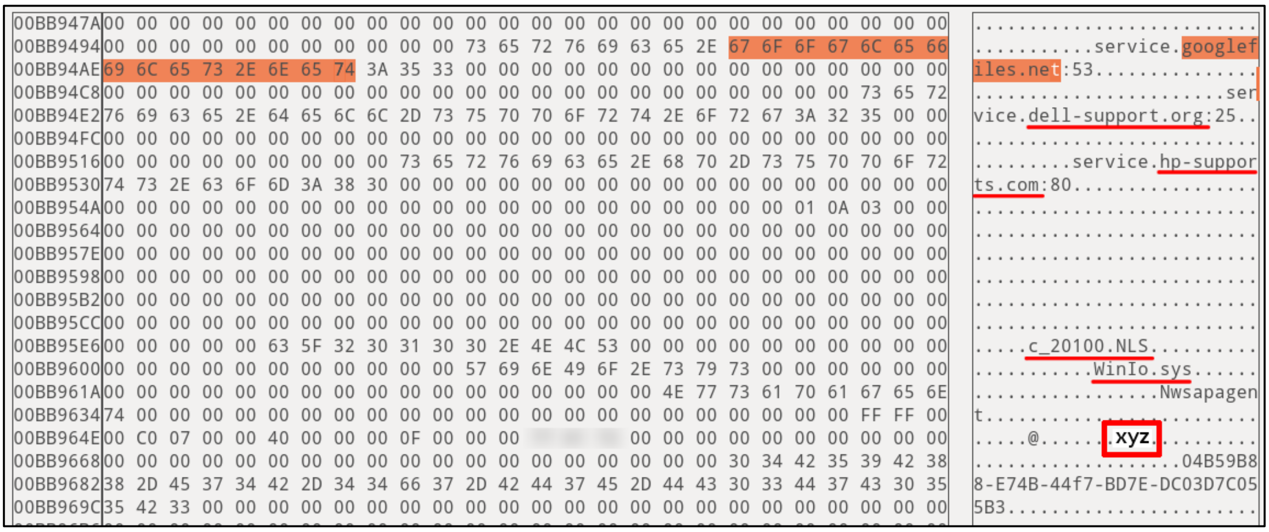
113.196.70.169.ll.static.sparqnet.net

Host names sharing IP with A records (2 items)

nx3.googlefiles.net
sky5858.com

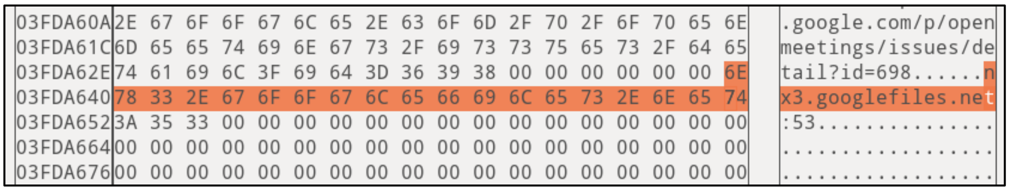
Robtex shared host names for IP 113.196.70.169.

One of these domains was found in the memory of dumped svchost process.



```
00BB947A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB9494 00 00 00 00 00 00 00 00 00 00 00 73 65 72 76 69 63 65 2E 67 6F 6F 67 6C 65 66  ...........service.googlef
00BB94AE 69 6C 65 73 2E 6E 65 74 3A 35 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  iles.net:53...............
00BB94C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 73 65 72  .......................ser
00BB94E2 76 69 63 65 2E 64 65 6C 6C 2D 73 75 70 70 6F 72 74 2E 6F 72 67 3A 32 35 00 00  vice.dell-support.org:25..
00BB94FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB9516 00 00 00 00 00 00 00 00 00 73 65 72 76 69 63 65 2E 68 70 2D 73 75 70 70 6F 72  .........service.hp-suppor
00BB9530 74 73 2E 63 6F 6D 3A 38 30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ts.com:80.................
00BB954A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 0A 03 00 00  ..........................
00BB9564 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB957E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB9598 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB95B2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB95CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..........................
00BB95E6 00 00 00 00 00 63 5F 32 30 31 30 30 2E 4E 4C 53 00 00 00 00 00 00 00 00 00 00  .....c_20100.NLS..........
00BB9600 00 00 00 00 00 00 00 00 00 00 00 57 69 6E 49 6F 2E 73 79 73 00 00 00 00 00 00  ...........WinIo.sys......
00BB961A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4E 77 73 61 70 61 67 65 6E  .................Nwsapagen
00BB9634 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF 00  t.........................
00BB964E 00 C0 07 00 00 40 00 00 00 0F 00 00 00 [redacted] 00 00 00 00 00 00 00 00 00 00  .....@.......xyz..........
00BB9668 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 30 34 42 35 39 42 38  ...................04B59B8
00BB9682 38 2D 45 37 34 42 2D 34 34 66 37 2D 42 44 37 45 2D 44 43 30 33 44 37 43 30 35  8-E74B-44f7-BD7E-DC03D7C05
00BB969C 35 42 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  5B3.......................
```

Domain name related to the suspected IP address on Machine #2.



```
03FDA60A 2E 67 6F 6F 67 6C 65 2E 63 6F 6D 2F 70 2F 6F 70 65 6E  .google.com/p/open
03FDA61C 6D 65 65 74 69 6E 67 73 2F 69 73 73 75 65 73 2F 64 65  meetings/issues/de
03FDA62E 74 61 69 6C 3F 69 64 3D 36 39 38 00 00 00 00 00 00 6E  tail?id=698......n
03FDA640 78 33 2E 67 6F 6F 67 6C 65 66 69 6C 65 73 2E 6E 65 74  x3.googlefiles.net
03FDA652 3A 35 33 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  :53...............
03FDA664 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..................
03FDA676 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..................
```

Part of executable configuration seen in svchost memory dump of Machine #1.

Googlefiles.net domain was also found in svchost dump of the Machine #1. Besides that, several other domain names were discovered in the same memory block:

service.interdriver.net
service.googlefiles.net
service.dell-support.org
service.hp-supports.com

Next step was to locate the nearest PE header in the memory of svchost and extract the executable module. After fixing alignment of the sections the file was ready for further static analysis.

Date and time from PE header showed that the executable was prepared about a year before current attack was revealed:

TimeDateStamp: "2011-10-13 07:21:50"

The executable was a 64-bit application which means that the attackers had already known that CompanyXYZ used 64-bit systems.

The IP address **188.120.246.88**, which was seen in suspicious connection was also checked. Connecting to the port 80 of that address with simple TCP client displayed an HTTP GET request:

```
GET /G-Content_XYZ.rar HTTP/1.1
Accept: */*
Cache-Control: no-cache
Connection: Keep-Alive
Host: 127.0.0.1:81
Pragma: no-cache
Range: bytes=23021988299-27335921161
Referer: http://127.0.0.1:81
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
```

Usually the request is sent by the connecting client, but here the chat between client and server is obviously inversed. That is probably done by the attackers' tunneling setup which established a TCP connection with some local web server within the company network and an external host that received the stolen data. According to the request, the attackers were downloading a file called G-Content_XYZ.rar, which seems to be an archive of over 25Gb long. The transfer process was instantly interrupted by Security Officers of the company.

Tactical Pattern Recognition

The embedded configuration shows some file names. C_20100.NLS was discovered later as the file hosting the same malicious code on the hard drive. WinIo.sys is a driver module on Microsoft Windows Server systems used to process networking requests.

Another interesting piece of data was in a short string "xyz", which probably refers to the attack campaign name and was defined by the attackers, who deliberately put that name to tag the malware. The word "xyz" most likely stands for the campaign name which comes from the attacked company's name "CompanyXYZ".

That was the first evidence that we were dealing with a well-prepared targeted attack against CompanyXYZ. From our previous experience, we have seen several targeted attacks against gaming companies and some of them were also tagged after the name of the companies. In all those attacks there was a recognizable pattern of the attackers: they always used third-level domain names for the command and control server of the malware while second-level domain name usually resolved to 127.0.0.1 or was a public DDNS domain. A quick check confirmed that this tactical pattern was present in this case as well. Since then, we believed that it is the same attackers we already knew about. This group of attackers was internally labeled “Winnti” by one of our researchers, who named it after one of the very first discovered executable malicious modules.

Active Attack Countermeasures

As soon as we discovered additional configuration, secondary domain names and IP addresses that could be used to control the infected hosts, we instantly reported it to the CompanyXYZ’s Security Officer, who instantly adjusted network firewall rules to block all connections to the attackers’ hosts.

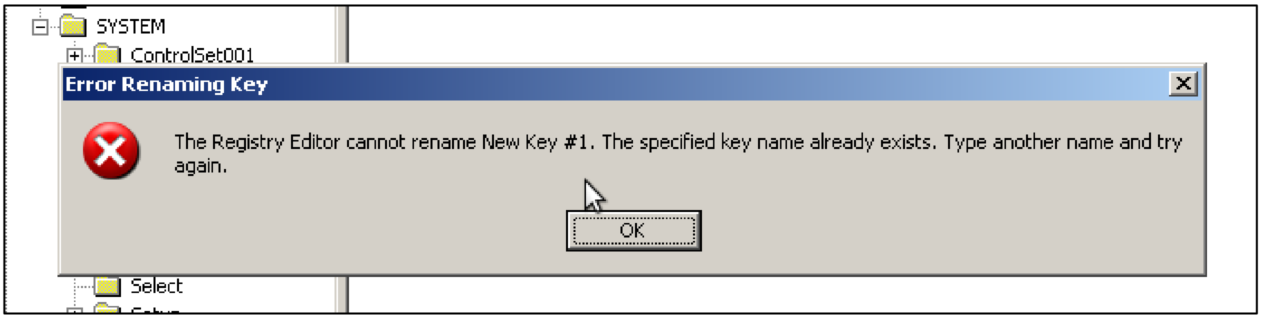
Assisted remote system analysis of another infected machine resulted in discovery of C_20100.NLS file in the Windows system directory and a reference in the system registry to start malicious module as system service:

```
HKLM\System\CurrentControlSet\services\Nwsapagent\Parameters\

 ServiceDll = C:\Windows\system32\c_20100.NLS

 ServiceMain = StartMain

 ServiceDllUnloadOnStop = 1
```

Date of registry key creation was the first discovered time of the attack (however, we found an earlier date later):

```
Thu Sep  6 04:26:19 2012
```

Malicious service registry settings were hidden by a rootkit module, however it helped to identify an infection as the registry key name was the same on all the affected computers. Simple creation of a key named HKLM\System\CurrentControlSet\services\Nwsapagent could fail if the system was infected.



Error Renaming Key

The Registry Editor cannot rename New Key #1. The specified key name already exists. Type another name and try again.

Rootkit detection method - registry key renaming fails if the key already exists.

The rootkit module protected the registry key, but it didn't protect the executable module stored on the hard drive. It was possible to rename c_20100.NLS file, reboot the machine and clean the registry.

Alternative and even more reliable method was to reboot into Windows Safe Mode, clean the registry key and delete the c_20100.NLS file. This method was used by company's System Administrators to find other modules that were not in c_20100.NLS.

The Infection Vector

Since the infection was located and cleaned, the next step was to locate the breach used by the attackers to penetrate the network. Security Officers of the company suggested to start checking from a distinct host they have

suspected. The host (lets call it Machine #3) belong to an employee without network administrators rights. It was known that it had connected to the attackers' IPs like the server systems.
The affected company’s security officers obtained a copy of the hard drive of the suspected machine and provided a remote access to the disk image. Browsing through the directory structure based on the suspected and adjacent dates of infection (01-06 September 2012) revealed a couple of suspicious files that could have been related to the attack:

C:\RECYCLER\en.exe
Type: PE file
Created: **2012-09-06 04:08:53 UTC**
Size: 405504
MD5: cf119a66d4c3e2355c1ec4ac316a7130

C:\WINDOWS\system32\drivers\tcprelay.sys
Type: PE file (native)
Created: **2012-09-05 17:27:04 UTC**
Size: 99912
MD5: 0b105cd6ecdfe5724c7db52135aa47ef

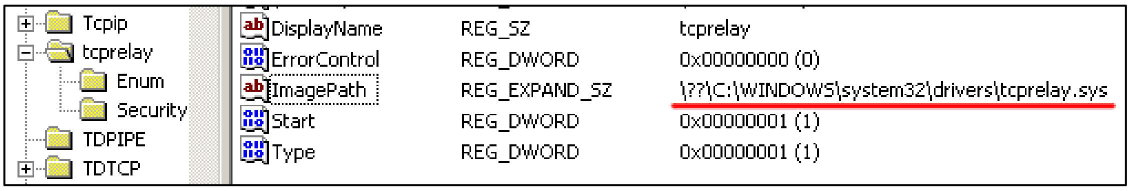
Preliminary analysis of tcprelay.sys proved that it was a malicious file which had another encrypted executable file embedded in it. This gave an even earlier suspected timestamp of infection:

2012-09-05 17:27:04 UTC or 2012-09-05 20:27:04 (local system timezone, UTC+3)

At the time of check there was no reference in the registry that was linked to tcprelay.sys, perhaps due the fact that system administrators had already cleaned the registry. This was confirmed by a file in local Administrator's Desktop folder:

C:\Documents and Settings\Administrator\Desktop\1.reg (created on 2012-09-24 12:44:07 UTC)

The file had an exported registry data, which had been removed from the registry during system cleanup on 24th September 2012. Here is the original contents of the registry key (**HKLM\SYSTEM\CurrentControlSet\Services\tcprelay**) before it was removed:

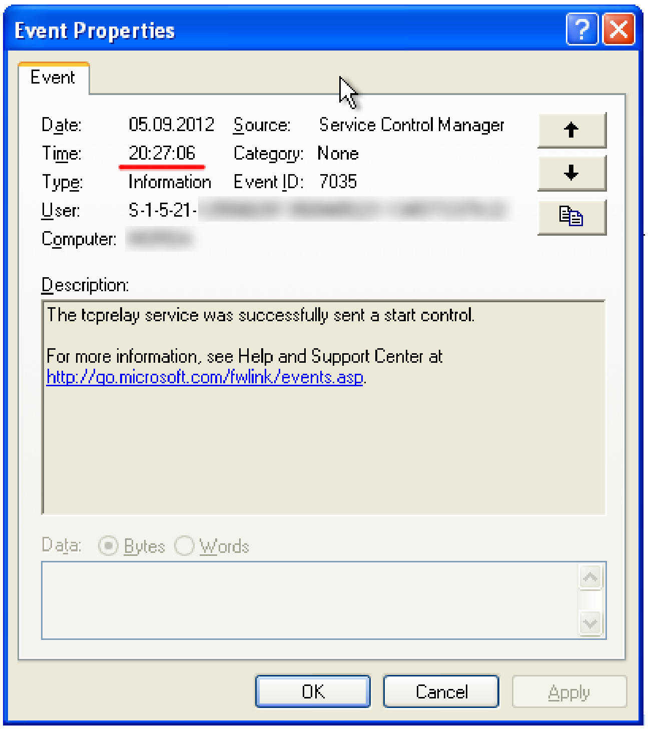


DisplayName	REG_SZ	tcprelay
ErrorControl	REG_DWORD	0x00000000 (0)
ImagePath	REG_EXPAND_SZ	\??\C:\WINDOWS\system32\drivers\tcprelay.sys
Start	REG_DWORD	0x00000001 (1)
Type	REG_DWORD	0x00000001 (1)

Tcprelay.sys registry settings with original file path.

Once the infection on the machine was confirmed we started looking for the origins of the malicious files. From our previous experience of Winnti gang tactics, we knew that they are keen on sending targeted emails with attached executables. Security Officers helped us check all the emails stored in local Outlook database file on suspected dates of infection, however that didn't reveal anything suspicious.

We have also found system event log files which were copied and analyzed. Event logs had records of tcprelay service start timestamps which confirmed the discovered date of infection. User SID corresponded to the local user account according to the registry.



Date: 05.09.2012 Source: Service Control Manager
Time: 20:27:06 Category: None
Type: Information Event ID: 7035
User: S-1-5-21-[redacted]
Computer: [redacted]

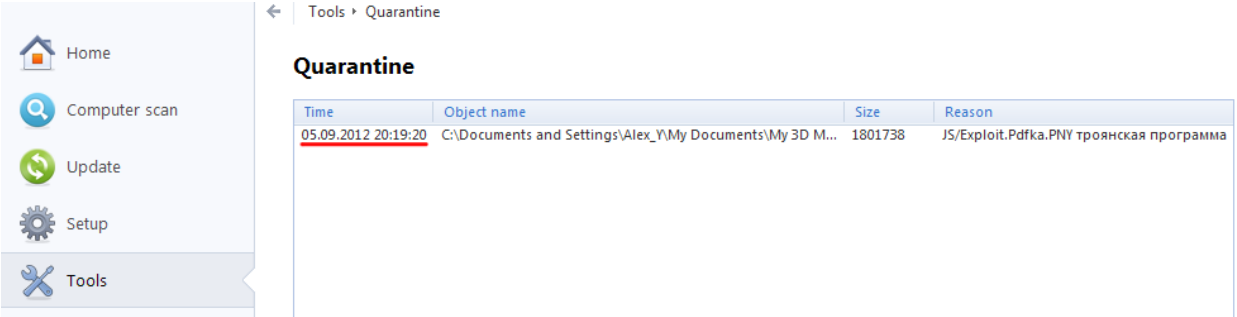
Description:
The tcprelay service was successfully sent a start control.

For more information, see Help and Support Center at http://go.microsoft.com/fwlink/events.asp.

Data: ● Bytes ○ Words

Tcprelay service first start time from the Event Log

The Machine #3 had an anti-virus program installed. Checking detection logs of the anti-virus on the suspected date of infection (05.09.2012) showed that there was a single detection right before tcprelay service first start.

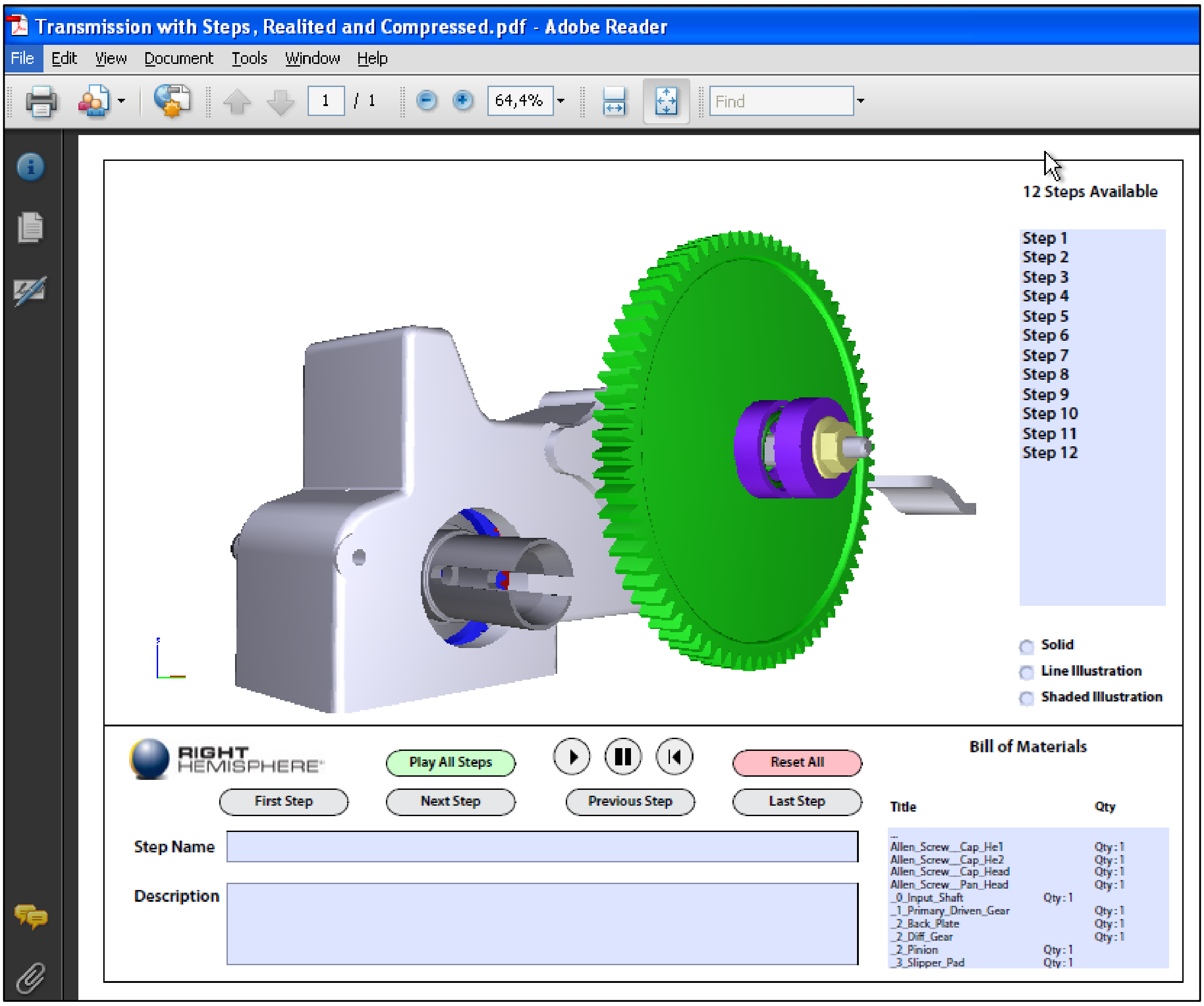


Quarantine

Time	Object name	Size	Reason
05.09.2012 20:19:20	C:\Documents and Settings\Alex_Y\My Documents\My 3D M...	1801738	JS/Exploit.Pdfka.PNY троянская программа

Part of the antivirus quarantine log.

We recovered the PDF document called “*Transmission with Steps, Realited and Compressed.pdf*” from the anti-virus quarantine and prepared to find an exploit inside. The PDF had a lot of obfuscated JavaScript code inside, however we believe that it was not related to the original infection of the system. It was clean and the anti-virus detected it by mistake, probably because of some suspicious obfuscated JavaScript code.



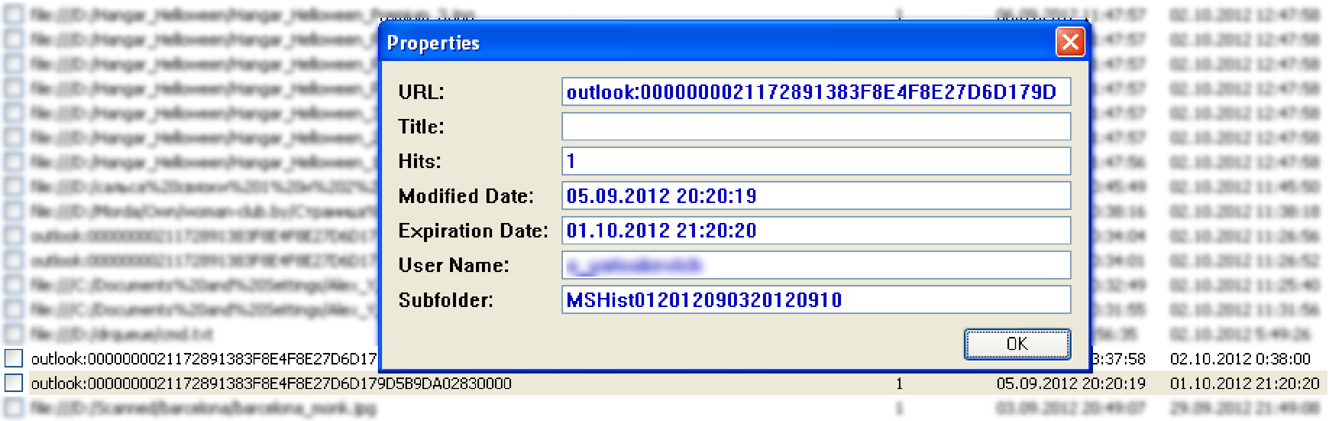
Bill of Materials

Title	Qty
..	
Allen_Screw__Cap_He1	Qty : 1
Allen_Screw__Cap_He2	Qty : 1
Allen_Screw__Cap_Head	Qty : 1
Allen_Screw__Pan_Head	Qty : 1
_0_Input_Shaft	Qty : 1
_1_Primary_Driven_Gear	Qty : 1
_2_Back_Plate	Qty : 1
_2_Diff_Gear	Qty : 1
_2_Pinion	Qty : 1
_3_Slipper_Pad	Qty : 1

PDF document detected by the antivirus as malicious.

The JavaScript code inside the PDF was used to process an interactive form inside the PDF and support dynamic interactive 3D model embedded in the document using Adobe 3D technology.

After that, we checked the infected machine’s browser history. The Internet Explorer history log files showed that the user was reading email right before the infection of his machine.


[redacted]

Properties

URL: outlook:00000000211728913 83F8E4F8E27D6D179D

Title:

Hits: 1

Modified Date: 05.09.2012 20:20:19

Expiration Date: 01.10.2012 21:20:20

User Name: [redacted]

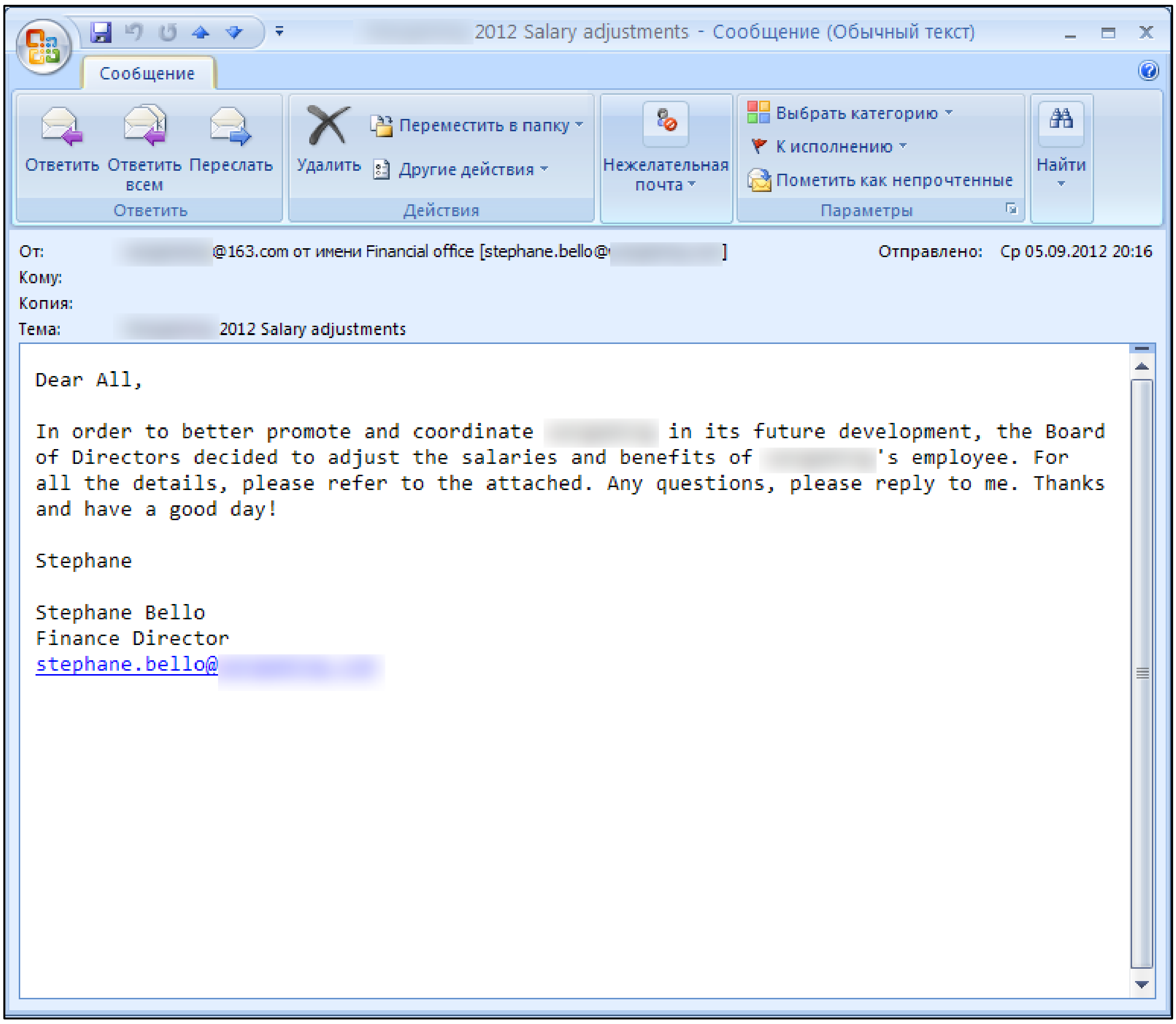
Subfolder: MSHist012012090320120910

```
□ outlook:0000000021172891383F8E4F8E27D6D17                    3:37:58               02.10.2012 0:38:00
□ outlook:0000000021172891383F8E4F8E27D6D179D5B9DA02830000   1   05.09.2012 20:20:19   01.10.2012 21:20:20
```

[redacted]

Internet Explorer log history record: html file from Outlook.

With that in mind, we analyzed the Outlook local database again. This time we used several techniques to recover emails that were deleted from the Trash folder. This helped to partly recover a message which arrived on the day of infection.

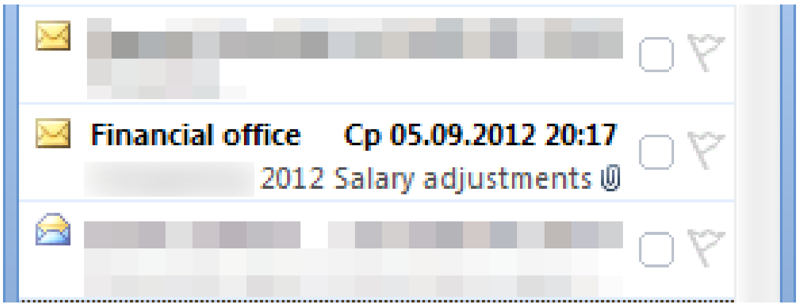


От: [redacted]@163.com от имени Financial office [stephane.bello@[redacted]]
Отправлено: Ср 05.09.2012 20:16
Кому:
Копия:
Тема: [redacted] 2012 Salary adjustments

```
Dear All,

In order to better promote and coordinate [redacted] in its future development, the Board
of Directors decided to adjust the salaries and benefits of [redacted]'s employee. For
all the details, please refer to the attached. Any questions, please reply to me. Thanks
and have a good day!

Stephane

Stephane Bello
Finance Director
stephane.bello@[redacted]
```

Recovered targeted attack email on Machine#3.

The text of the message supposed to contain an attachment, however the attachment and MIME headers of the email were completely lost and couldn't be recovered. However, it was clear that the email was a targeted attack against the employee of the company. It was sent from companyxxyz@163.com and replaced "From" field in the email body which made it look like a legitimate email in the list of messages in Outlook.



[redacted]

Financial office Ср 05.09.2012 20:17
[redacted] 2012 Salary adjustments

[redacted]

Targetted attack email in the list of Outlook messages.

We discovered a Windows prefetch file in the system directory, that was created when the malicious attachment was opened. The timestamp correlates with the time of infection.

C:\WINDOWS\Prefetch\CompanyXYZ EMPLOYEE SALARY ADJ-1AF9D56A.pf

Time of creation: 2012-09-05 19:52:00 (local timezone, UTC+03)

Unfortunately, the prefetch file format is proprietary and there is nothing interesting in those files, except the original executable file name. Full path of the malicious executable that infected the first computer in the company was:

C:\Documents and Settings\<Username>\LocalSettings\Temp\RAR$EX00.156\CompanyXYZ EMPLOYEE SALARY ADJUSTMENTS EBOOK.EXE

According to the file path, this executable was a part of an archive, which was opened with WinRAR installed on the system.

Upon discovery, we requested the Security Officers to provide us with full MIME   as well as to check who else may have received the same message. The check discovered series of emails sent to several publicly known email addresses. In all cases the text message was the same as shown above, however sent from different mailboxes. The Return-Path MIME filed seemed to have the original email addresses of the attackers:

companyxxyz@163.com
company.xyz@gmx.com

The attackers used the same IP to send out emails: 118.142.11.114

```
inetnum:  118.140.0.0 - 118.143.255.255
netname:HGC
descr:              Hutchison Global Communications
country:  HK
person:   ITMM HGC
nic-hdl:  IH17-AP
e-mail:   hgcnetwork@hgc.com.hk
address:  9/F Low Block ,
address:  Hutchison Telecom Tower,
address:  99 Cheung Fai Rd, Tsing Yi,
address:  HONG KONG
phone:    +852-21229555
fax-no:   +852-21239523
```

The emails we checked had the same attachment of 96782 bytes named “*Salary adjustments.zip*”. There was only one file inside ZIP archive, called “*CompanyXYZ Employee Salary Adjustments Ebook.exe*”. Full details about this application are provided further down in current report.

To summarize, the targeted attack started from an email sent at 05.09.2012 19:12 (UTC+03).
It resulted in system infection at 05.09.2012 19:52 (UTC+03).

Full File Analysis

Salary adjustments.zip File

Size: 96782
MD5: 1b56416fefa2d2c863f3b46dfb6dc353
Location: targeted attack email message attachment
Creation time (author's timezone): 2012-09-05 14:29:10

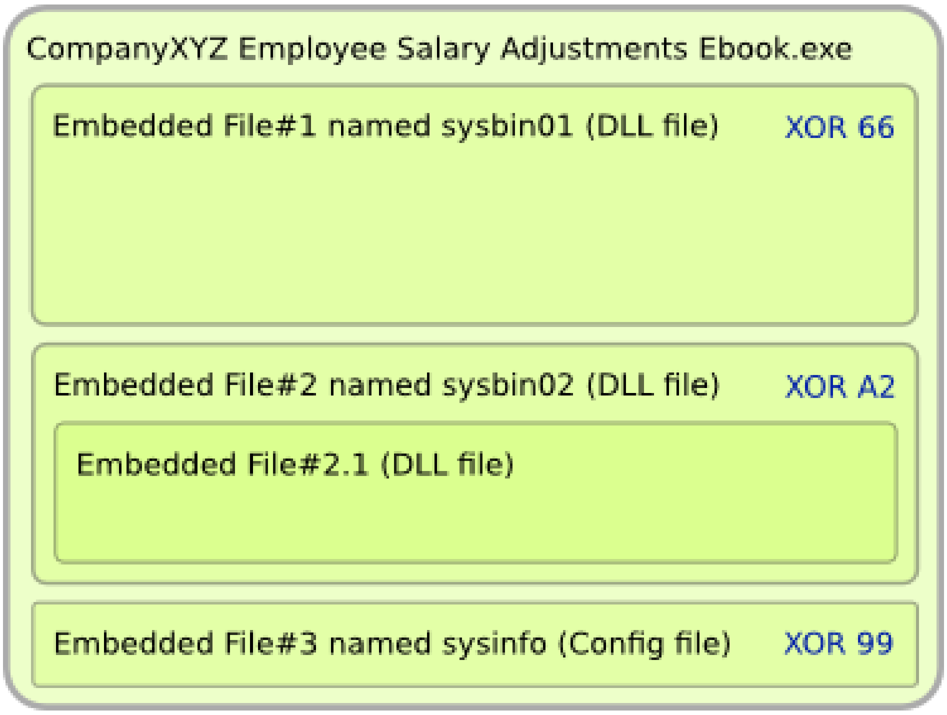
This file is just a container for "*CompanyXYZ Employee Salary Adjustments Ebook.exe*".

CompanyXYZ Employee Salary Adjustments Ebook.exe File

Size: 122880
MD5: 6ef66c2336b2b5aaa697c2d0ab2b66e2
Location: "Salary adjustments.zip"
Creation time: unavailable
Link time (UTC): 2012-07-21 18:50:18

Internal name: **FlashUpdate.EXE**

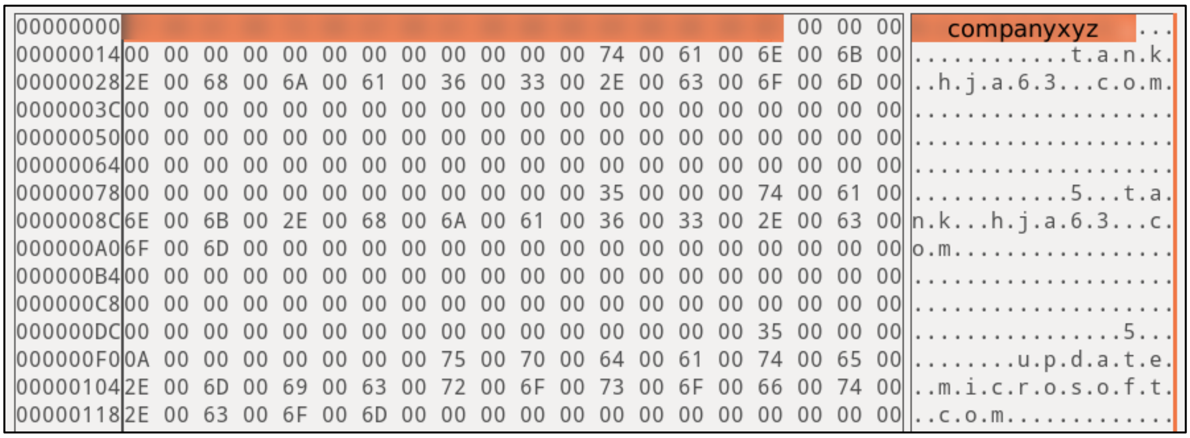
This application is a wrapper for another embedded executable modules. It serves as a dropper of malware.


CompanyXYZ Employee Salary Adjustments Ebook.exe
Embedded File#1 named sysbin01 (DLL file)
XOR 66
Embedded File#2 named sysbin02 (DLL file)
XOR A2
Embedded File#2.1 (DLL file)
Embedded File#3 named sysinfo (Config file)
XOR 99

Malware dropper file structure

Notable fact: this application has a resource section inside and the default locale is set to Chinese Simplified.

The file creates three long binary data registry keys, two of which are encrypted executable modules and one encrypted config from the body of the original dropper. These values are encrypted with simple 1-byte XOR.



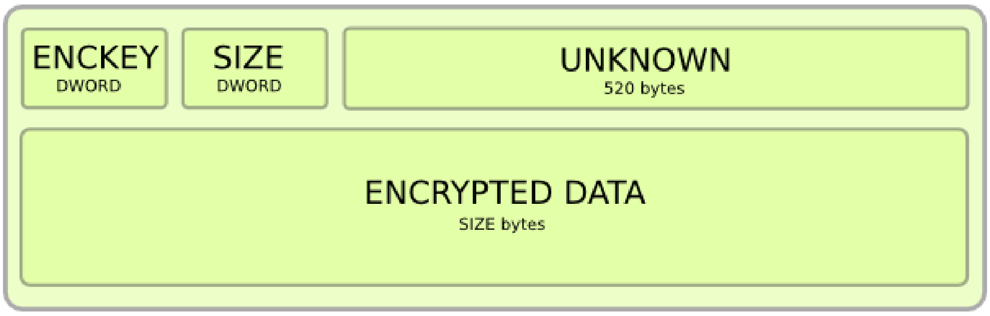
```
00000000 [redacted]                                            00 00 00  companyxyz...
00000014 00 00 00 00 00 00 00 00 00 00 00 00 74 00 61 00 6E 00 6B 00  ............t.a.n.k.
00000028 2E 00 68 00 6A 00 61 00 36 00 33 00 2E 00 63 00 6F 00 6D 00  ..h.j.a.6.3...c.o.m.
0000003C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
00000064 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
00000078 00 00 00 00 00 00 00 00 00 00 00 00 35 00 00 00 74 00 61 00  ............5...t.a.
0000008C 6E 00 6B 00 2E 00 68 00 6A 00 61 00 36 00 33 00 2E 00 63 00  n.k...h.j.a.6.3...c.
000000A0 6F 00 6D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  o.m.................
000000B4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
000000C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....................
000000DC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 35 00 00 00  ................5...
000000F0 0A 00 00 00 00 00 00 00 75 00 70 00 64 00 61 00 74 00 65 00  ........u.p.d.a.t.e.
00000104 2E 00 6D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00 74 00  ..m.i.c.r.o.s.o.f.t.
00000118 2E 00 63 00 6F 00 6D 00 00 00 00 00 00 00 00 00 00 00 00 00  ..c.o.m.............
```

Decrypted sysinfo config contents

Sysinfo config module is used by sysbin01. Apparently it starts with the company name and has three domain names, one of which is most likely used to check Internet connectivity (update.microsoft.com).

Sysbin01 module is a loader component. It creates several threads running various jobs.

Sysbin01.thread#1 attempts to load %TEMP%\<ComputerName>.ax file and decrypts it.



<ComputerName>.ax file structure

We checked the system but couldn't find <ComputerName>.ax file in the Temp folder of the user, however we found other .ax-files that seemed to be related because of the date of file creation.

File name: C:\Documents and Settings\%User%\Local Settings\Temp\%ComputerName%_p.ax
File size: 2660
Creation time (UTC): 2012-09-06 06:22:42
MD5: unavailable (the system went offline before we discovered the filepath).

File name: C:\Documents and Settings\%User%\Local Settings\Temp\uid.ax
File size: 16
Creation time (UTC): 2012-09-06 05:03:06
MD5: unavailable (the system went offline before we discovered the filepath).

According to the code that loads <ComputerName>.ax it is an encrypted executable file, which is decrypted and loaded to memory by own loader routine in the sysbin01 module.

Sysbin01.thread#2 spawns a new instance of **Sysbin01.thread#3** every 10 seconds during, that is done 3 times.

Sysbin01.thread#3

This thread is the most important. It reads the configuration from the registry and connects to the C&C servers specified in the config via direct tcp connection or via proxy that is fetched from the the settings of locally logged in user profile. The config had the following C&C: **tank.hja63.com**. It sends a "POST /<HEXNUMBER>" request with User-Agent "lynx", the data after HTTP header is just "AA", expected answer is also "AA".

This thread also creates %TEMP%\uid.ax and stores current system unique ID, which is generated by CoCreateGuid system API (16 bytes). It is able to receive and save data from the C&C server to a file. It also monitors windows of explorer.exe and copies textual data from password fields if the user types in, stolen data is saved to a file first.

After all threads are launched, the main thread waits for termination of **Sysbin01.thread#3**, which is created first and then exits.

sysbin02 module behaviors depends on currently running processes. There is an embedded DLL file according to Figure 15 in sysbin02.

If the system has a running process named "360tray.exe", then the embedded file is stored in %SYSTEM%\MFC42LOC.DLL, then copies the source executable (FlashUpdate.exe) to %TEMP%\Flash.tmp and runs a new process from that location via WMI Win32_Process.Create method.

If the system has a running process named "bdagent.exe", then it copies the source executable (FlashUpdate.exe) to %TEMP%\Flash.tmp, decodes an embedded Base64 string and executes. The string has the following text after decoding:

reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v FlashUpdate /t REG_EXPAND_SZ /d """"%APPDATA%\FlashUpdate.exe""" -update activex" /f

The module also saves current module file path to the registry in the following key location: HKCU\Software\Classes\path

Next it patches the tmp file with two dword "AAAA" values which looks like corruption of embedded encrypted sysbin modules inside. The meaning of this action is currently unclear.

Then it moves Flash.tmp file to FlashUpdate.exe by and starts a new process from new location.

Finally, if there is not "qqpctray.exe" process running, and this seemed to be the case for the analyzed system, it copies the source executable (FlashUpdate.exe) to %TEMP%\Flash.tmp, patches the new file and increases its size by adding system explorer.exe file contents to the resource section "RC Data" 20 times. The purpose of this is to make the new executable look like the real update service of Adobe Flash, it simply stuffs the file with executable code of another application. Then it moves the file to new location *%APPDATA%\FlashUpdate.exe*, saves new module file path to the registry in the following key location: HKCU\Software\Classes\path and starts a new process from there.

c_20100.NLS (aka SrvCore.dll) File

Size: 15847156
MD5: 5778178a1b259c3127b678a49cd23e53
Location: C:\WINDOWS\system32\c_20100.NLS
Creation time (UTC): unavailable
Link time (UTC): 2011-09-16 13:23:34

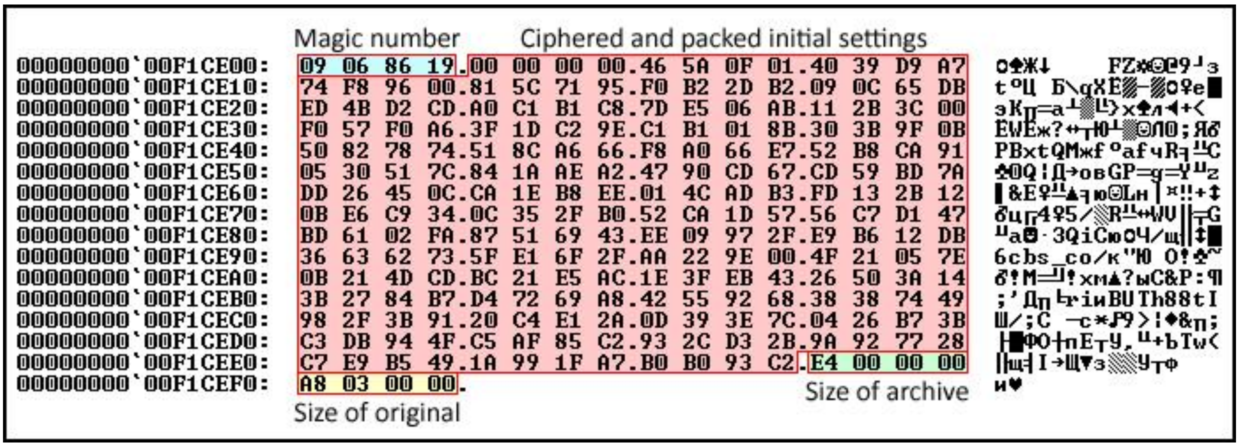
Summary

c_20100.NLS works in two modes. The first mode is a load as a dynamic library and the second is a launch as a service. Both branches have the same core functionality.

This module is a universal executable code loader with no embedded payload. Its main purpose is to connect to the C&C server, download and store the encrypted payload in the system registry. It is also responsible for loading, decrypting and running the payload module from the registry after system restart.

Details

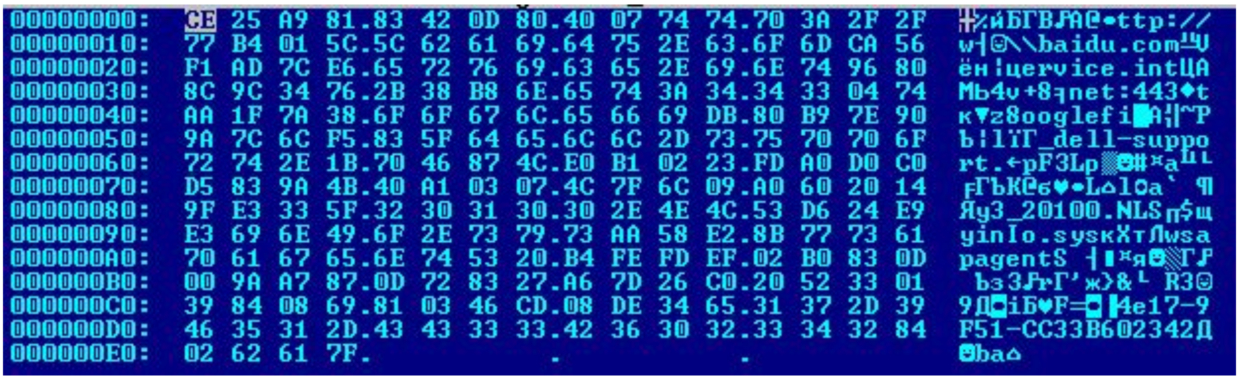
c_20100.NLS contains a ciphered block with initial settings. This ciphered block resides at the very end of the file of this malicious program and is decrypted in the beginning of execution. Structure of block:



```
00000000`00F1CE00:  09 06 86 19.00 00 00 00.46 5A 0F 01.40 39 D9 A7  o♠Ж↓    FZ☼☺@9┘з
00000000`00F1CE10:  74 F8 96 00.81 5C 71 95.F0 B2 2D B2.09 0C 65 DB  t°Ц Б\qХЁ▓-▓o♀e█
00000000`00F1CE20:  ED 4B D2 CD.A0 C1 B1 C8.7D E5 06 AB.11 2B 3C 00  эKπ═а┴▒╚}х♠л◄+<
00000000`00F1CE30:  F0 57 F0 A6.3F 1D C2 9E.C1 B1 01 8B.30 3B 9F 0B  ЁWЁж?↔┬Ю┴▒☺Л0;Яб
00000000`00F1CE40:  50 82 78 74.51 8C A6 66.F8 A0 66 E7.52 B8 CA 91  PВxtQМжf°afчR╕╩С
00000000`00F1CE50:  05 30 51 7C.84 1A AE A2.47 90 CD 67.CD 59 BD 7A  ♣0Q|Д→овGP═g═Y╜z
00000000`00F1CE60:  DD 26 45 0C.CA 1E B8 EE.01 4C AD B3.FD 13 2B 12  ▌&E♀╩▲╕ю☺Lн│¤‼+↕
00000000`00F1CE70:  0B E6 C9 34.0C 35 2F B0.52 CA 1D 57.56 C7 D1 47  ♂ц╔4♀5/░R╩↔WV╟╤G
00000000`00F1CE80:  BD 61 02 FA.87 51 69 43.EE 09 97 2F.E9 B6 12 DB  ╜a☻·ЗQiCю○Ч/щ╢↕█
00000000`00F1CE90:  36 63 62 73.5F E1 6F 2F.AA 22 9E 00.4F 21 05 7E  6cbs_сo/к"Ю O!♣~
00000000`00F1CEA0:  0B 21 4D CD.BC 21 E5 AC.1E 3F EB 43.26 50 3A 14  ♂!M═╝!хм▲?ыC&P:¶
00000000`00F1CEB0:  3B 27 84 B7.D4 72 69 A8.42 55 92 68.38 38 74 49  ;'Дπ╘riиBUТh88tI
00000000`00F1CEC0:  98 2F 3B 91.20 C4 E1 2A.0D 39 3E 7C.04 26 B7 3B  Ш/;С ─с*♪9>|♦&╖;
00000000`00F1CED0:  C3 DB 94 4F.C5 AF 85 C2.93 2C D3 2B.9A 92 77 28  ├█ФO┼пЕ┬У,╙+ЪТw(
00000000`00F1CEE0:  C7 E9 B5 49.1A 99 1F A7.B0 B0 93 C2.E4 00 00 00  ╟щ╡I→Щ▼з░░У┬ф
00000000`00F1CEF0:  A8 03 00 00.                                      и♥
```

Initial settings in the end of file

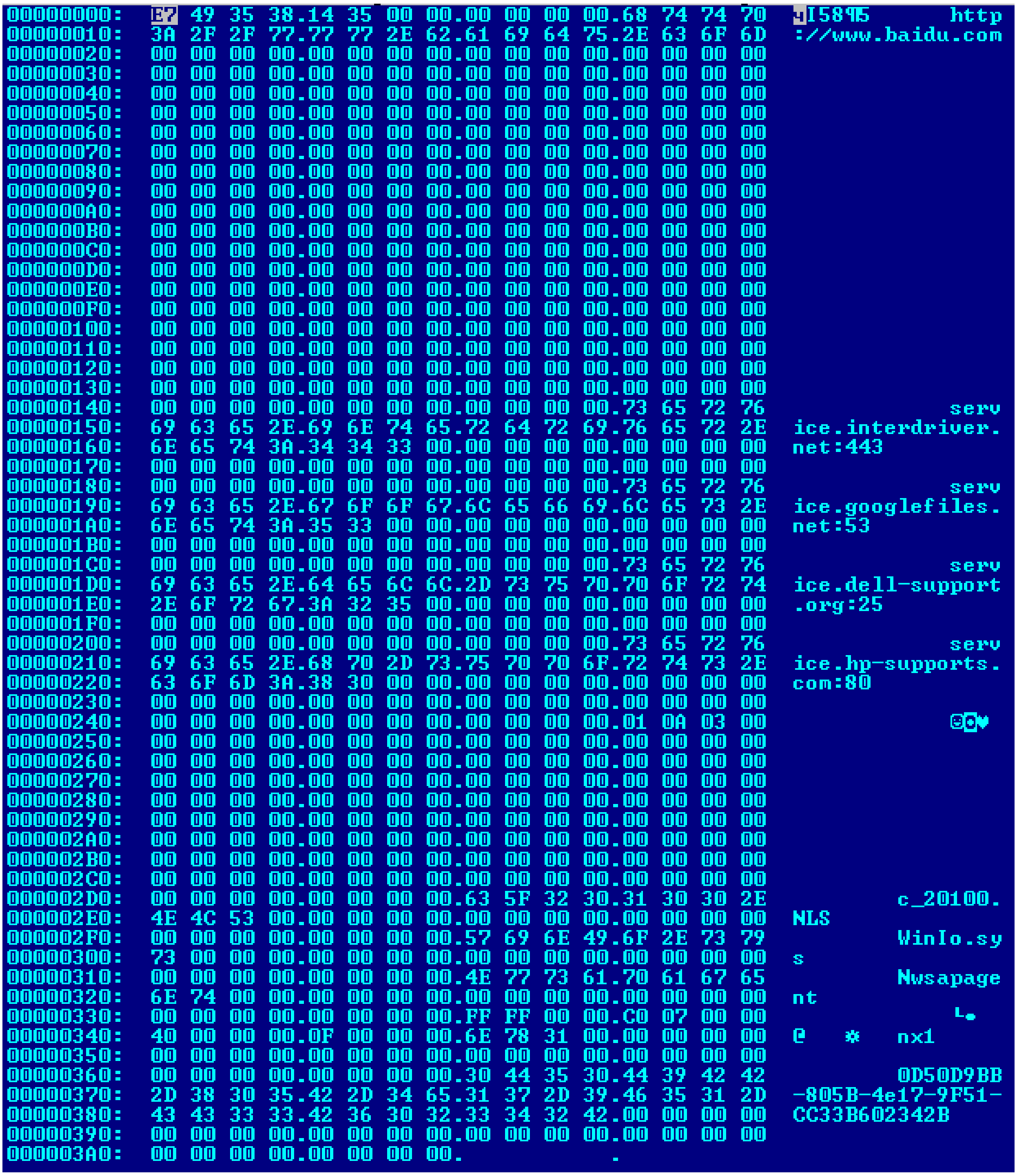
The malicious program XORs the magic number with a hardcoded value *0x19860609*, converts a resulted value into a HEX-string and uses that string as a key for RC4 cipher algorithm. In this case string-key represents "*00000000*" because of the magic number is equal to the hardcoded XORing value. With that key malicious program decrypts (RC4) ciphered archive. The archive has the following data:



```
00000000:  CE 25 A9 81.83 42 0D 80.40 07 74 74.70 3A 2F 2F  ╬%йБГB♪А@•ttp://
00000010:  77 B4 01 5C.5C 62 61 69.64 75 2E 63.6F 6D CA 56  w┤☺\\baidu.com╩V
00000020:  F1 AD 7C E6.65 72 76 69.63 65 2E 69.6E 74 96 80  ён|цervice.intЦА
00000030:  8C 9C 34 76.2B 38 B8 6E.65 74 3A 34.34 33 04 74  МЬ4v+8╕net:443♦t
00000040:  AA 1F 7A 38.6F 6F 67 6C.65 66 69 DB.80 B9 7E 90  к▼z8ooglefi█А╣~Р
00000050:  9A 7C 6C F5.83 5F 64 65.6C 6C 2D 73.75 70 70 6F  Ъ|lїГ_dell-suppo
00000060:  72 74 2E 1B.70 46 87 4C.E0 B1 02 23.FD A0 D0 C0  rt.←pFЗLр▒☻#¤а╨└
00000070:  D5 83 9A 4B.40 A1 03 07.4C 7F 6C 09.A0 60 20 14  ╒ГЪK@б♥•L⌂l○а` ¶
00000080:  9F E3 33 5F.32 30 31 30.30 2E 4E 4C.53 D6 24 E9  Яу3_20100.NLS╓$щ
00000090:  E3 69 6E 49.6F 2E 73 79.73 AA 58 E2.8B 77 73 61  уinIo.sysкXтЛwsa
000000A0:  70 61 67 65.6E 74 53 20.B4 FE FD EF.02 B0 83 0D  pagentS ┤■¤я☻░Г♪
000000B0:  00 9A A7 87.0D 72 83 27.A6 7D 26 C0.20 52 33 01   ЪзЗ♪rГ'ж}&└ R3☺
000000C0:  39 84 08 69.81 03 46 CD.08 DE 34 65.31 37 2D 39  9Д◘iБ♥F═◘▐4e17-9
000000D0:  46 35 31 2D.43 43 33 33.42 36 30 32.33 34 32 84  F51-CC33B602342Д
000000E0:  02 62 61 7F.                                      ☻ba⌂
```

Archive of initial settings

Custom LZ-like compression algorithm resembling was used to pack initial settings. After unpacking the following data appears:



```
00000000:  E7 49 35 38.14 35 00 00.00 00 00 00.68 74 74 70  чI58¶5      http
00000010:  3A 2F 2F 77.77 77 2E 62.61 69 64 75.2E 63 6F 6D  ://www.baidu.com
00000020:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000030:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000040:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000050:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000060:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000070:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000080:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000090:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000000A0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000000B0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000000C0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000000D0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000000E0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000000F0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000100:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000110:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000120:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000130:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000140:  00 00 00 00.00 00 00 00.00 00 00 00.73 65 72 76              serv
00000150:  69 63 65 2E.69 6E 74 65.72 64 72 69.76 65 72 2E  ice.interdriver.
00000160:  6E 65 74 3A.34 34 33 00.00 00 00 00.00 00 00 00  net:443
00000170:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000180:  00 00 00 00.00 00 00 00.00 00 00 00.73 65 72 76              serv
00000190:  69 63 65 2E.67 6F 6F 67.6C 65 66 69.6C 65 73 2E  ice.googlefiles.
000001A0:  6E 65 74 3A.35 33 00 00.00 00 00 00.00 00 00 00  net:53
000001B0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000001C0:  00 00 00 00.00 00 00 00.00 00 00 00.73 65 72 76              serv
000001D0:  69 63 65 2E.64 65 6C 6C.2D 73 75 70.70 6F 72 74  ice.dell-support
000001E0:  2E 6F 72 67.3A 32 35 00.00 00 00 00.00 00 00 00  .org:25
000001F0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000200:  00 00 00 00.00 00 00 00.00 00 00 00.73 65 72 76              serv
00000210:  69 63 65 2E.68 70 2D 73.75 70 70 6F.72 74 73 2E  ice.hp-supports.
00000220:  63 6F 6D 3A.38 30 00 00.00 00 00 00.00 00 00 00  com:80
00000230:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000240:  00 00 00 00.00 00 00 00.00 00 00 00.01 0A 03 00              ☺◙♥
00000250:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000260:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000270:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000280:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000290:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000002A0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000002B0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000002C0:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000002D0:  00 00 00 00.00 00 00 00.63 5F 32 30.31 30 30 2E          c_20100.
000002E0:  4E 4C 53 00.00 00 00 00.00 00 00 00.00 00 00 00  NLS
000002F0:  00 00 00 00.00 00 00 00.57 69 6E 49.6F 2E 73 79          WinIo.sy
00000300:  73 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00  s
00000310:  00 00 00 00.00 00 00 00.4E 77 73 61.70 61 67 65          Nwsapage
00000320:  6E 74 00 00.00 00 00 00.00 00 00 00.00 00 00 00  nt
00000330:  00 00 00 00.00 00 00 00.FF FF 00 00.C0 07 00 00              └•
00000340:  40 00 00 00.0F 00 00 00.6E 78 31 00.00 00 00 00  @   ☼   nx1
00000350:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
00000360:  00 00 00 00.00 00 00 00.30 44 35 30.44 39 42 42          0D50D9BB
00000370:  2D 38 30 35.42 2D 34 65.31 37 2D 39.46 35 31 2D  -805B-4e17-9F51-
00000380:  43 43 33 33.42 36 30 32.33 34 32 42.00 00 00 00  CC33B602342B
00000390:  00 00 00 00.00 00 00 00.00 00 00 00.00 00 00 00
000003A0:  00 00 00 00.00 00 00 00.          .
```

The Initial settings

The malicious program tries to read registry value "*SrvCode*" by registry path: *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion*. That value is expected to contain ciphered with RC4 data. To decrypt it program uses 2nd integer of initial settings (in this case 0x3514) XORed by hardcoded byte 0x12. Result is converted into a HEX-string and is used as RC4 key for further decryption (here it is "*00003506*"). That registry value appears if this malicious program had already worked on the system and received data from the C&C server in the past. Content of "*SrvCode*" poses a ciphered executable which should be loaded into the memory and run.

If "*SrvCode*" is not found malware makes attempts to connect to one of the specified C&C servers.

C&C Server Address Selection

Initial settings define the type of C&C format. Byte at offset 0x24C stores the C&C type value:

0x00: the malware uses 4 URL-based C&Cs placed at 0x4C, 0x8C, 0xCC and 0x10C offsets. By all appearances these are public resources (forums, blog platforms and so on) where the attackers leave messages with specially crafted content for a bot. If connection fails, the malware tries another approach.

0x01: the malware uses attackers' hardcoded servers and connects to host and port specified at offsets: 0x14C, 0x18C, 0x1CC and 0x20C. If connection fails the malware tries another approach.

If URL-based scheme is used then malware loads a web-page by specified in settings URL. The target text has to begin and end with special hardcoded delimiters: "B9273C17" – start, "B6A74634" – end. The malware reads contents of the webpages until it finds a proper page with delimiters. If found, the malware takes the text between delimiters and treats it as data of hex string, converts it to the binary data and decrypts resulted data using RC4 algorithm with hardcoded key "rtyr_45_trf". For example:

"B9273C17E67024277AE02E2A8A780B243C0BCA88FE85A1B6A7463",

The data between delimiters:

"E67024277AE02E2A8A780B243C0BCA88FE85A1",

It is converted into binary: *0xe6 0x70 0x24 … 0xa1* and this buffer is decrypted with RC4 key "*rtyr_45_trf*". Result is a host and port of C&C: "**nx2.intercpu.com:25**".

If the host-port schema is used then malware simply connects to the hardcoded C&C servers directly.

Communication with C&C Server

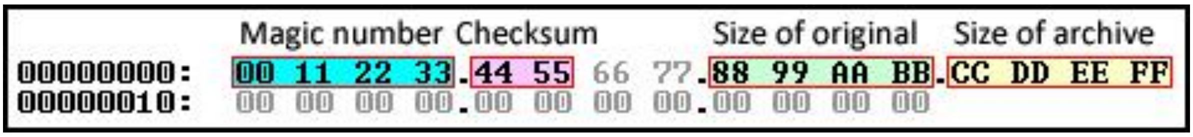
Once a working C&C server is found the malware sends specially crafted ciphered buffer to via TCP/IP. On request from a bot a C&C server replies with several blocks of data described below:

1st block

0xC bytes of header: *0x1000010, 0x1000010,* <reserved 4 bytes>.

2nd block

0x1C bytes (due to absence of real respond from the C&C I'm presenting an example buffer of this block containing bytes *0x00, 0x11, 0x22, … , 0xFF, 0x00, 0x0, 0x0, …, 0x00*):



```
00000000: 00 11 22 33.44 55 66 77.88 99 AA BB.CC DD EE FF
00000010: 00 00 00 00.00 00 00 00.00 00 00 00
```

2nd block of 0x1c bytes: example

First DWORD of this buffer (here, *0x33221100*) is a magic number which is XORed with the value *0x1986052*. Resulted lowest byte is used to XOR unpacked data.

Word at offset *0x4* (here, *0x5544*) poses a checksum of unpacked data which should correspond with actual received content.

DWORD starting at offset *0x8* (here, *0xBBAA9988*) represents a size of unpacked data.

Value at marked place at *0xC* offset (in example picture it is dword *0xFFEEDDCC*) represents a size of next block of data to be received. That data will pose an archive, hence this value represents a size of packed data.

3rd block

The 3rd block poses an archive of ciphered data. Being received, unpacked and decrypted, data is ciphered again with RC4 and stored into "*SrvCode*" value of registry by mentioned above registry path.

The eceived data is processed as an executable file to run. The malware places the executable in memory, prepares for running and makes call to the entry point of the new code. Then it waits when following event will be triggered:

Global\D5ACF9F6-C8B3-47d1-9768-57162E1F5FDB

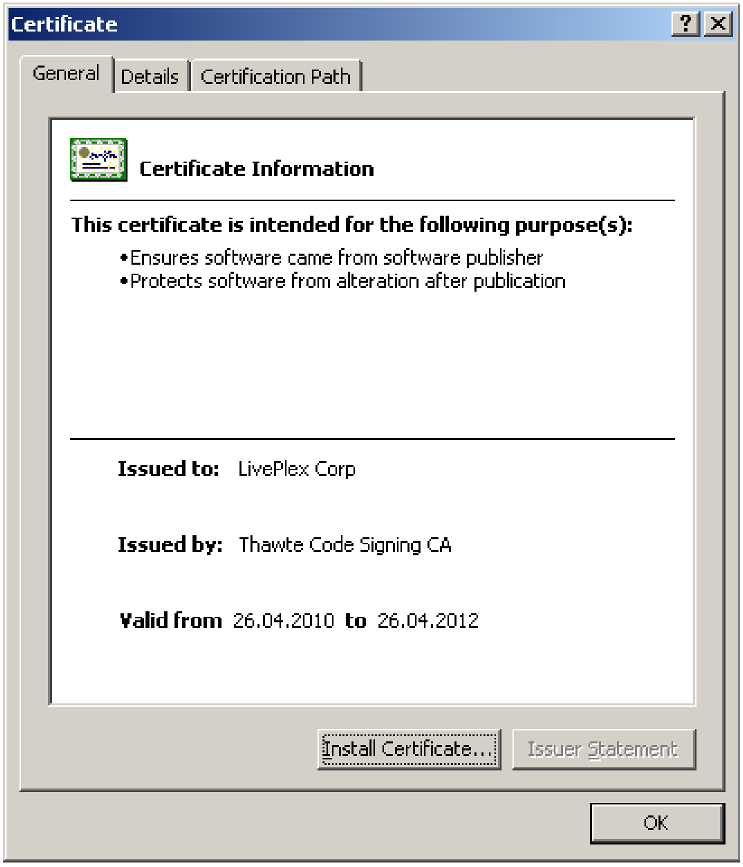
When triggered, the malware finishes execution. During the process of finishing it deletes registry value "*SrvCode*" along with values "*DrvCode*" and "*KeyCode*" from the same registry path although this malware was not creating them.

Tcprelay.sys File

Size: 99912
MD5: 0b105cd6ecdfe5724c7db52135aa47ef
Location: C:\WINDOWS\system32\drivers\tcprelay.sys
Creation time (UTC): 2012-09-05 17:27:04
Link time (UTC): 2011-12-21 13:55:03

This file is a Microsoft Windows native application, which is loaded as a driver and had a valid digital signature in 2012.

The certificate was issued by LivePlex Corp, which can be found online by searching for the company name. One of their webpages is here: http://www.linkedin.com/company/liveplex



Certificate Information

This certificate is intended for the following purpose(s):

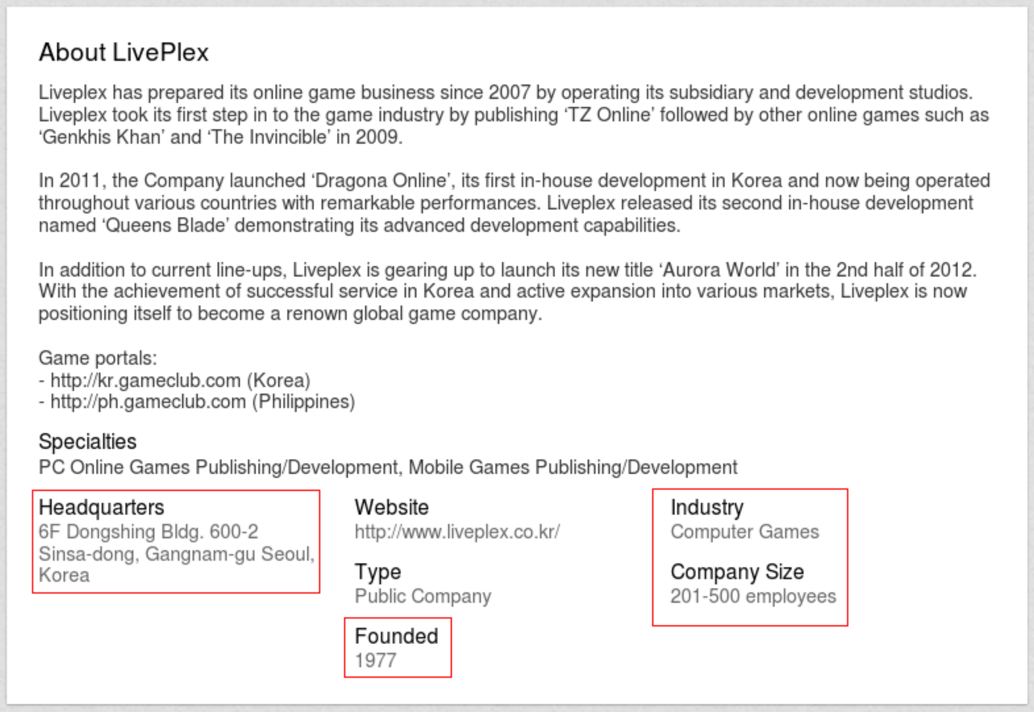
- Ensures software came from software publisher
- Protects software from alteration after publication

Issued to: LivePlex Corp

Issued by: Thawte Code Signing CA

Valid from 26.04.2010 to 26.04.2012

Digital certificate of Tcprelay.sys



About LivePlex

Liveplex has prepared its online game business since 2007 by operating its subsidiary and development studios. Liveplex took its first step in to the game industry by publishing 'TZ Online' followed by other online games such as 'Genkhis Khan' and 'The Invincible' in 2009.

In 2011, the Company launched 'Dragona Online', its first in-house development in Korea and now being operated throughout various countries with remarkable performances. Liveplex released its second in-house development named 'Queens Blade' demonstrating its advanced development capabilities.

In addition to current line-ups, Liveplex is gearing up to launch its new title 'Aurora World' in the 2nd half of 2012. With the achievement of successful service in Korea and active expansion into various markets, Liveplex is now positioning itself to become a renown global game company.

Game portals:
- http://kr.gameclub.com (Korea)
- http://ph.gameclub.com (Philippines)

Specialties

PC Online Games Publishing/Development, Mobile Games Publishing/Development

Headquarters
6F Dongshing Bldg. 600-2
Sinsa-dong, Gangnam-gu Seoul,
Korea

Website
http://www.liveplex.co.kr/

Type
Public Company

Founded
1977

Industry
Computer Games

Company Size
201-500 employees

LivePlex profile page on LinkedIn

When the driver is loaded it decrypts an embedded DLL file, which is immediately injected into the address space of services.exe process. Then the driver sets up some rootkit functionality to hide TCP connections by patching the system tcp/ip driver.

The injected DLL was called s.dll at the time of compilation and is yet another module for analysis.

S.dll File

Size: 77825
MD5: 1716889fcee461e7cde5128c14d206cb
Location: inside tcprelay.sys
Creation time (UTC): 2012-09-05 17:27:04
Link time (UTC): 2011-03-01 09:07:12

This opens system event named "401d-b49a-93cf7a18e5b3" and sets event to fired state if it exists. The code checks for proxy server configuration by impersonating a logged in user and fetching settings from the registry. It can work both with Socks and HTTP proxies. The module attempts to connect to the list of 8 domains, consisting of the following command and control servers (some of them are used more than once):

a1.googletrait.com
a1.nexongame.net
a1.reegame.net
mail.nexongame.net

It automatically looks for open C&C ports in the following order 53,443,8080,25,80,3690,1433,80.

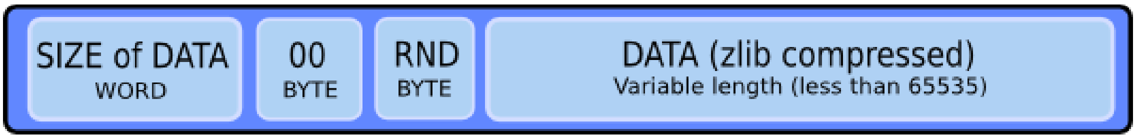
During connection over HTTP proxy it uses the following User-Agent string: "MyApp/0".

The application is linked with libmysql.dll and Zlib (v.1.2.3). Current Zlib version is 1.2.7 and was released on 2nd May 2012, while version 1.2.3 seems to be released in July 2005. Zlib version 1.2.4 was released on March 2010, so the original module was probably designed somewhere after July 2005 and before March 2010.

Then it collects system information, which includes the following:

Host name
OS Service Pack version
System default language ID and Code page
List of local drives with free space
Internal hardcoded identifier ("12-21")
Process commandline
Logged in user name
System directory path
Amount of free system memory
CPU name
Terminal services port number

The information is stored in a buffer that begins with hardcoded header magic number: 0xDF1F1ED3. The block is compressed using Zlib (v.1.2.3) compress2 method with compression level 8. The data is compressed later and prepended by a 4-bytes header as shown below.



Format of a message sent to C&C

After submitting system information the module expects 4 byte response code from the server after which it sends one 00 byte to complete the handshake procedure.

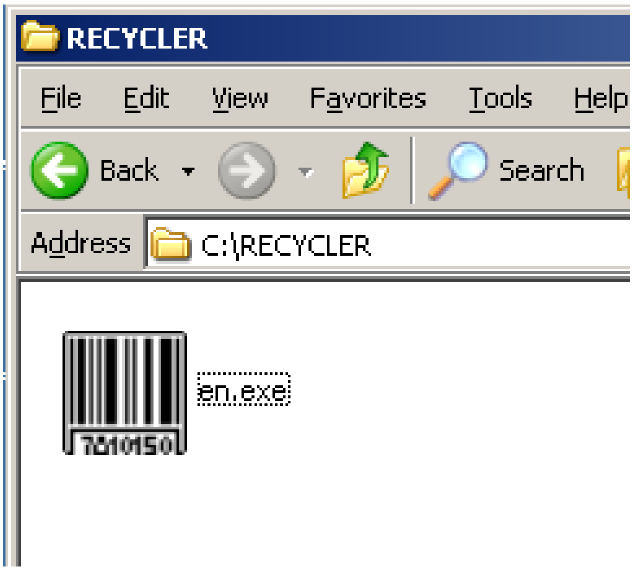
Then the module expects an interactive communication session with the remote operator. It provides capability to run various commands including (command names were defined during reverse engineering):

process_list
kill_process
dir_list
smbshare_list
smbshare_mount
dir_make
file_delete
file_move
file_upload
file_open
file_write
file_close
file_find
url_download_to_file
process_start
process_start_and_get_output
dll_load
dll_call_export
screen_getsnapshot
screen_set_cursor_position
screen_send_input
tcpproxy_open_connection
tcpproxy_close_connection
mysql_connect
mysql_fetch
mysql_disconnect
driver_tcpreplay_interact
tcpsession_close
quit

A command output is compressed using Zlib and sent to the server in asynchronous mode. To summarize, it is obvious that this executable module is a backdoor, capable of taking screenshots, stealing files, downloading new files from the Internet, starting and killing processes, including interactive Windows shell commands, file search and interaction with mysql database server.

En.exe File

Size: 405504
MD5: cf119a66d4c3e2355c1ec4ac316a7130
Location: C:\RECYCLER\en.exe
Creation time (UTC): 2012-09-06 04:08:53
Link time (UTC): 2009-11-17 16:02:04



An icon embedded in en.exe is a default application icon from MS Visual Studio

This application is a dropper, it fetches a resource called EXEFILE from current application and saves it into following paths:

<CURRENT DIR>\dllcache\sethc.exe

C:\WINDOWS\system32\sethc.exe

Then the module uses undocumented Windows API from SFC_OS.dll, a function called SfcFileException to update the system version of C:\WINDOWS\system32\sethc.exe.

The file C:\WINDOWS\system32\sethc.exe (SET High Contrast) is to enable the High Contrast accessibility feature in order to allow people with visual impairments to log in. SETHC is activated at logon screen with LeftAlt+LeftShift+PrintScreen key combination.

By replacing C:\Windows\SYSTEM32\SETHC.EXE with a custom application an attacker can run an arbitrary application with SYSTEM privileges running in zero session (in separate desktop space from normal applications).

After the new file replaced the system sethc.exe application, current module adjusts the privileges of sethc.exe to disable access to the file from any other application. This is achieved by calling external system tools cacls.

Replace access rights to the files, allow everyone full access:

```
cacls C:\WINDOWS\system32\sethc.exe /c /e /p everyone:f
```

```
cacls <CURRENT DIR>\dllcache\sethc.exe /c /e /p everyone:f
```

Revoke access to the file for everyone, leave only system readonly access:

cacls C:\WINDOWS\system32\sethc.exe /t /c /e /r everyone

cacls C:\WINDOWS\system32\sethc.exe /t /c /e /r administrators

cacls C:\WINDOWS\system32\sethc.exe /t /c /e /r users

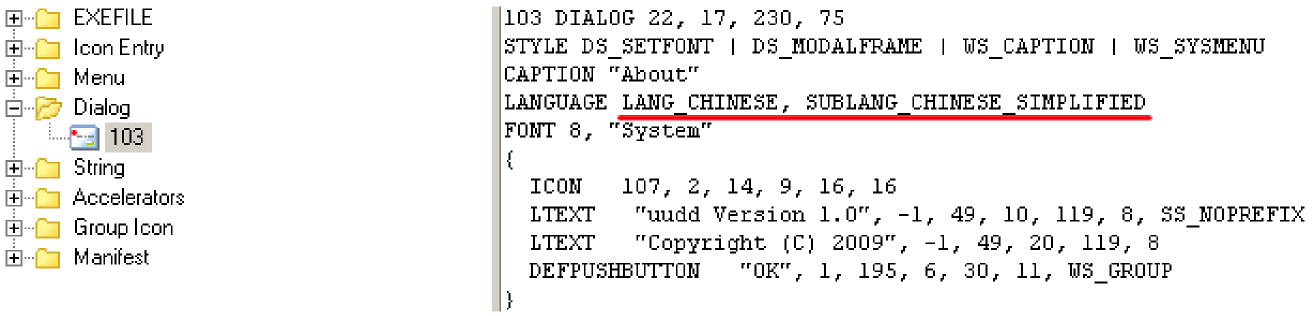
cacls C:\WINDOWS\system32\sethc.exe /t /c /e /r system

cacls C:\WINDOWS\system32\sethc.exe /t /c /e /r “Power Users”

cacls C:\WINDOWS\system32\sethc.exe /c /e /p system:r

The dropper also changes the file timestamp. It is set identical to C:\WINDOWS\system32\ntvdm.exe.

The dropper application has a resource section with Menu, Dialog templates and other information put by the MS Visual Studio Application Wizard. It includes default system locale from the developer's system, which is Chinese Simplified.



```
103 DIALOG 22, 17, 230, 75
STYLE DS_SETFONT | DS_MODALFRAME | WS_CAPTION | WS_SYSMENU
CAPTION "About"
LANGUAGE LANG_CHINESE, SUBLANG_CHINESE_SIMPLIFIED
FONT 8, "System"
{
  ICON   107, 2, 14, 9, 16, 16
  LTEXT   "uudd Version 1.0", -1, 49, 10, 119, 8, SS_NOPREFIX
  LTEXT   "Copyright (C) 2009", -1, 49, 20, 119, 8
  DEFPUSHBUTTON   "OK", 1, 195, 6, 30, 11, WS_GROUP
}
```

Chinese locale in resource section of En.exe

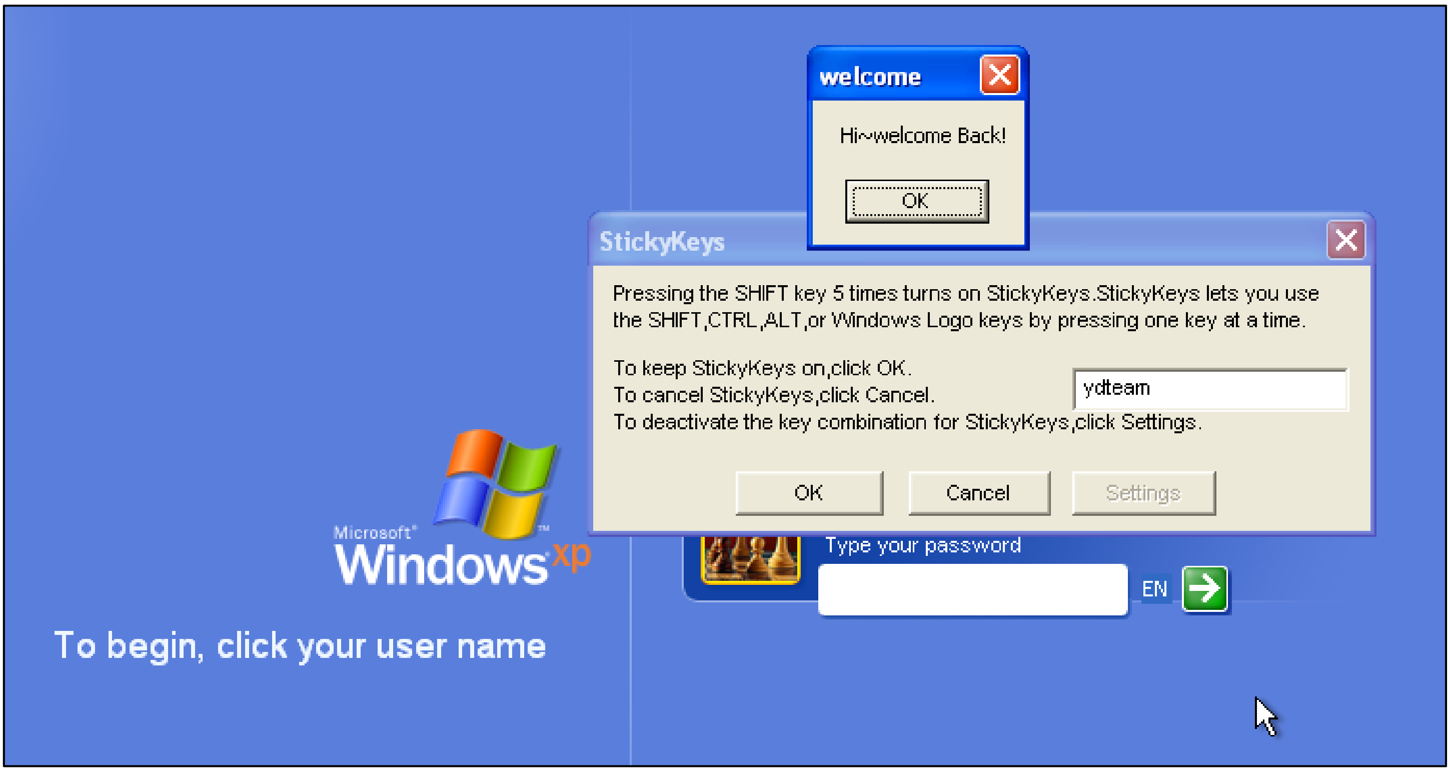
The dropped application (from resource EXEFILE) is described below as sethc.exe.

Sethc.exe File

Size: 20480
MD5: 3ba06424e8244f17a8d269c4d40c39c9
Location: resource section of En.exe
Link time (UTC): 2009-05-16 07:09:35

This small file has very basic functionality. It is written using MS Visual C++ with MFC and is used to render a simple dialog window. Like En.exe it has resource section, describing the dialog window and default locale is set to Chinese Simplified.

Once it replaced local system sethc.exe tool it can be invoked when the desktop is locked with LeftCtrl+LeftShift+PrintScr key combination. This brings a dialog Window similar to system StickyKeys application. However, if you press Ctrl+Alt+F you will immediately see a hidden input box. If you enter “ydteam” in the input box and press Ctrl+Alt+K, the application will welcome you with a message box and will execute a TaskManager.



welcome

Hi~welcome Back!

StickyKeys

Pressing the SHIFT key 5 times turns on StickyKeys.StickyKeys lets you use the SHIFT,CTRL,ALT,or Windows Logo keys by pressing one key at a time.

To keep StickyKeys on,click OK.
To cancel StickyKeys,click Cancel.
To deactivate the key combination for StickyKeys,click Settings.

ydteam

Fake SetHighContrast application in action

As far as sethc.exe is executed with privileges of local system, the task manager also inherits these privileges and is capable of killing any other process as well as starting any other application with system rights. Apparently, this is a backdoor to the system. An attacker can run cmd.exe, add local users with administrative privileges and log in. We checked if the tool was publicly shared on the Internet, but couldn't find a page distributing it freely. That is why we assume that it is developed and used privately.

Full list of C&Cs

Below is full list of all collected domains and IP-addresses of C&C servers have they been mentioned in initial settings of *c_20100.nls* or hidden in text messages at public places in Internet:

C&Cs from public resources:

27.115.103.198:8885
27.115.103.195:8885
114.222.36.32:10000
27.115.103.195:23456
27.115.103.195:10000
nx2.joymax.in:80
nx3.joymax.in:80

nx2.intercpu.com:25 (174.36.138.30)
nx3.intercpu.com:25 (174.36.138.30)
nx3.interdriver.net:53 (119.240.212.110)
stan227.guicp.net:8008

Hardcoded C&C from the malware:

service.interdriver.net:443 (98.126.218.64, 199.188.106.231)
service.googlefiles.net:53 (98.126.218.64, 199.188.106.231)
service.dell-support.org:25
service.hp-supports.com:80
tank.hja63.com
a1.googletrait.com
a1.nexongame.net
a1.reegame.net
mail.nexongame.net

Interestingly, there is an overlap of C&Cs from public resources and hardcoded domains:

nx3.interdriver.net:53 <===> service.interdriver.net:443

*The nx3.interdriver.net was published by **awertasegfae@yahoo.com** and was discovered at*

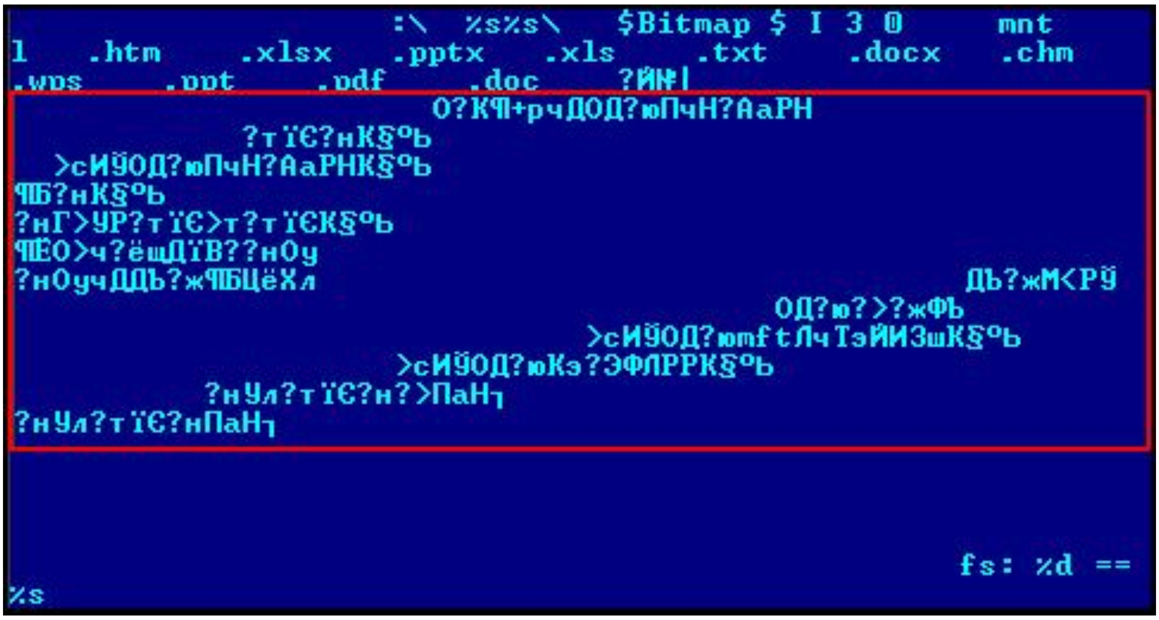
http://awertasegfae.blogspot.ru/2011/10/first-test.html. This means that at least the individual who owns awertasegfae@yahoo.com for sure belongs to the same gang who attacked CompanyXYZ.

Source of Attacks

So, who is behind Winnti? While analyzing the malicious files that we detected during our investigations we found some details which may cast some light on the source of the attacks.

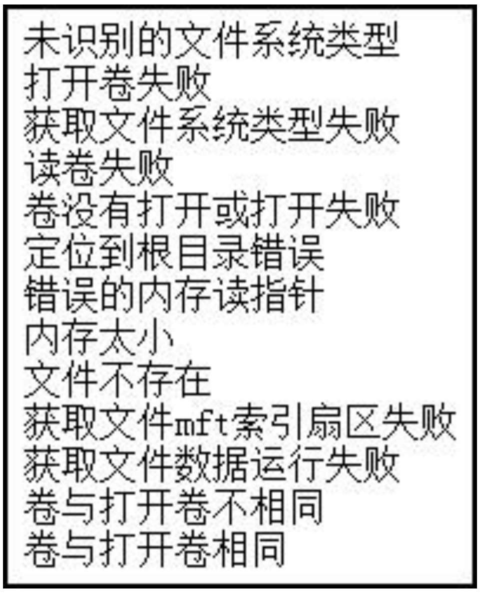
As part of our investigation, we monitored exactly what the cybercriminals did on an infected PC. In particular, they downloaded an auxiliary program ff._exe to the Config.Msi folder on the infected machine. This code searches for HTML, MS Excel, MS Word, Adobe, PowerPoint and MS Works documents and text files (.txt) on the hard drive.

Debugging lines were found in ff._exe_ that possibly point to the nationality of the cybercriminals. They were not immediately noticeable because they looked like this in the editor:



```
                          :\   %s%s\    $Bitmap $ I 3 0      mnt
l    .htm      .xlsx     .pptx     .xls      .txt      .docx     .chm
.wps      .ppt      .pdf      .doc      ?ЙN¦
                                        О?К¶+рчДОД?юПчН?АаРН
                  ?тЇЄ?нК§ºЬ
   >сИЎОД?юПчН?АаРНК§ºЬ
¶Б?нК§ºЬ
?нГ>ЧР?тЇЄ>т?тЇЄК§ºЬ
¶ЁО>ч?ёщДЇВ??нОу
?нОучДДЬ?ж¶БЦёХл                                            ДЬ?жМ<РЎ
                                                        ОД?ю?>?жФЬ
                                           >сИЎОД?юmftЛчТэЙИЗшК§ºЬ
                            >сИЎОД?юКэ?ЭФЛРРК§ºЬ
             ?нЧл?тЇЄ?н?>ПаН┐
?нЧл?тЇЄ?нПаН┐



                                                                  fs: %d ==
%s
```

However, during a detailed analysis it emerged that the text is in Chinese Simplified GBK coding. This is what these lines look in Chinese:



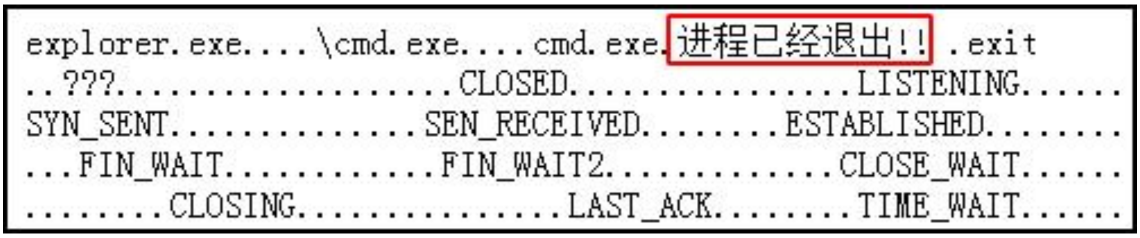
未识别的文件系统类型
打开卷失败
获取文件系统类型失败
读卷失败
卷没有打开或打开失败
定位到根目录错误
错误的内存读指针
内存太小
文件不存在
获取文件mft索引扇区失败
获取文件数据运行失败
卷与打开卷不相同
卷与打开卷相同

Below is a machine translation of this text into English:

Not identify the type of file system
Below is a translation of the text by interpreter
Open the volume failed
Failed to get the file system type
Failed to read volume
Volumes do not open or open failed
Navigate to the root directory of the error
Error memory read pointer
Memory is too small
File does not exist
Failed to get the file mft index sector
Access to file data fail
Volume and open volumes are not the same
The same volume and open volume

In addition, cybercriminals used the AheadLib program to create malicious libraries (for details, see the second part of the article). This is a program with a Chinese interface.

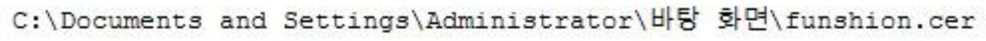
Chinese text was also found in one of the components of the malicious program CmdPlus.dll plug-in:



```
explorer.exe....\cmd.exe....cmd.exe.进程已经退出!! .exit
..???..................CLOSED...............LISTENING......
SYN_SENT..............SEN_RECEIVED........ESTABLISHED........
...FIN_WAIT...........FIN_WAIT2.............CLOSE_WAIT......
........CLOSING...............LAST_ACK........TIME_WAIT......
```

Translation: The process is complete!!

It would appear that the attackers can at least speak Chinese. However, not everything is so clear cut: because the file transfer plug-in has not been implemented entirely safely, a command which includes the attackers' local path (where the file comes from and where it is saved to) arrives during the process of downloading/uploading files on the infected system. While monitoring the cybercriminals' activity on the infected machine, we noticed they uploaded the certificate they found in the infected system, and the network traffic reflected the local path indicating the place where they saved the file on their computer:



```
C:\Documents and Settings\Administrator\바탕 화면\funshion.cer
```

These characters appear to be Korean, meaning "desktop". This means the attackers were working on a Korean Windows operating system. Therefore, we can presume that the attack is not exclusively the work of Chinese-speaking cybercriminals.

The Search for Attackers (XYZ incident)

Locating the attacker is one of the most non-trivial parts of the research. The attackers normally do not leave any traces in the malware that can be directly bound to their real identities. That is why we have to use all available bits of information that seems to find other unique related content on the Internet or any other available data sources. One of the important stages is to extract unique identifiers/nicknames/tags that can be discovered on the Internet and after that find individuals who are related to creation or distribution of this content.

YDTeam Hacking Group

The string “ydteam” looked non-random and we decided to check it on the Internet. It turned out that YDteam is a hackers group name and has a lot of references on Chinese segment of the Internet:

http://zhikou.yo2.cn/ - probably a team member web blog

http://www.exploit-db.com/exploits/11053/ - PoC exploit for Chinese media player by the team member called “t-bag”

Another team member called “b4che10r” according to

http://zzsky.5d6d.net/archiver/tid-127.html

http://hi.baidu.com/0x255/item/22cbbfe97ca9963c87d9de41

http://www.indetectables.net/viewtopic.php?f=87&t=22185&view=print

b4che10r's personal blog: http://blog.taskkill.net/

Another team member called “Shalyse” according to

http://forum.cnsec.org/thread-50222-1-1.html

Another team member called “killer” according to

http://zzsky.5d6d.net/archiver/tid-127.html

There was a website ydteam.cn that seems to be related to the activity of the group. According to the domaintools.com database, it was registered on 2009-10-06 15:12 and put on hold around 2010-10-08. The original WHOIS information from domaintools.com:

```
Domain Name: ydteam.cn
ROID: 20091006s10001s23027085-cn
Domain Status: ok
Registrant Organization: 魏楠
Registrant Name: 魏楠
Administrative Email: wn6805@126.com
Sponsoring Registrar: 北京新网数码信息技术有限公司
Name Server:ns.xinnetdns.com
Name Server:ns.xinnet.cn
```

Registration Date: 2009-10-06 15:12
Expiration Date: 2010-10-06 15:12

Registrant name 魏楠 (Wei Nan) seems to be represented in the mailbox wn6805@126.com, which could mean the owner of the website used real identity. The domain was most likely registered by the team leader.

The email itself was used on several other websites. For example

http://tieba.baidu.com/f?ct=335544320&lm=0&rn=30&tn=baiduPostBrowser&sc=0&z=633089789&pn=0&word=%BC%AF%C4%FE%D2%BB%D6%D0

The webpage above has a post offering to “help with cheap shopping online”. That is most likely related to a fraudulent activity of the email owner (stolen Internet-banking credentials or credit card information). The same page reveals a QQ id of that individual and a username:

QQ:	97676416
Username:	大头禹

Another page http://www.gtvod.com/gtvod/jsp/public/personal/index.jsp?id=20100127213936126005 shows information about the user registered with name “wn3118” and the same email:

E-mail:	wn6805@126.com
Date of Birth:	1992-12-21
Marital Status:	Unmarried

Another page http://tieba.baidu.com/p/652667782 has a message from profile “灬low-key, wn” (which links to wn6805@126.com). Profile information reveals gender of the individual:
http://www.baidu.com/p/%E7%81%AC%E4%BD%8E%E8%B0%83%E4%B8%B6wn/detail

Gender: Male

There are few essays in Chinese probably written by the individual owning wn6805@126.com while studying at Junior High School:

(posted on 2008-09-24): http://www.zww.cn/zuowen/html/25/258151.htm

(posted on 2008-10-05) http://www.zww.cn/zuowen/html/25/263081.htm

(posted on 2009-04-08): http://www.zww.cn/zuowen/html/51/350029.htm

A page from zww.cn also shows some details about the author:
http://www.zww.cn/zw/myzw.asp?u=%CA%A7%C8%A5%B0%AE

Birthday:	1992-12-21 (confirms previous finding)
QQ:	251985076
Joined:	2008-09-16 22:35:00
Last login:	2009-06-09 10:37:00

Searching for the QQ id 251985076 brings to http://blog.sina.com.cn/dahuadl that has
User mobile number: 13847416805

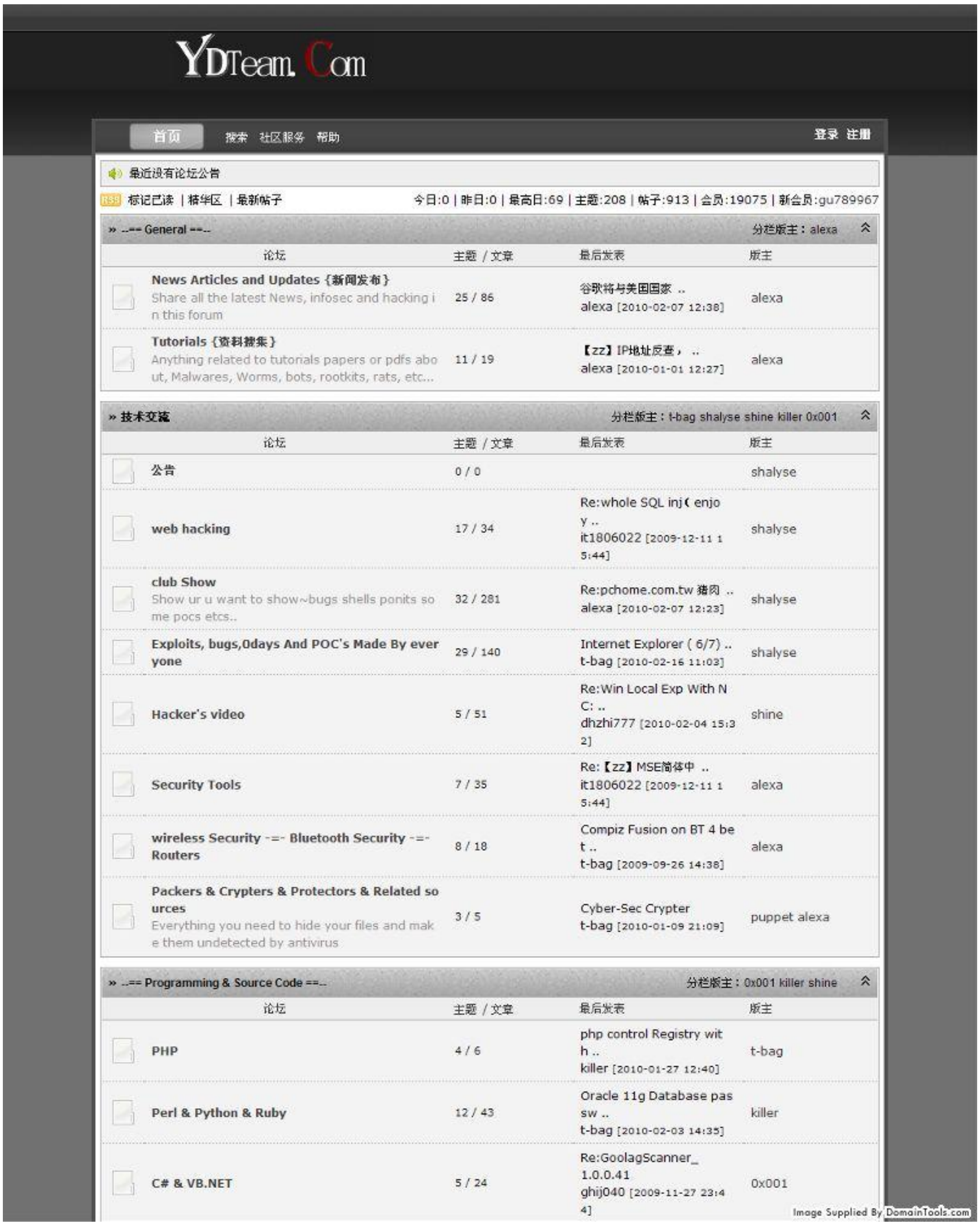
The hackers team also seemed to own ydteam.com for some time according to reference at http://zzsky.5d6d.net/archiver/tid-127.html

Domaintools.com shows that the domain was registered to a Chinese individual from 2009-06-03 to 2011-08-22. After that WHOIS information was protected by a Privacy protection service. Here is WHOIS data at the time of domain registration:

```
Admin Name........... zheng wenlong
Admin Address........ tianjin jiefangdongjie 63hao
Admin Address........ yancheng
Admin Address........ 300560
Admin Address........ fujian
Admin Address........ CHINA
Admin Email.......... vydteam@yahoo.cn
Admin Phone.......... +86.13652452428
```

Please note, that +8613652452428 is a Chinese local cell phone number.

Domaintools.com has also preserved a screenshot of the website while it was online on 2010-02-25. It shows some of the team member names mentioned above.



最近没有论坛公告

RSS 标记已读 | 精华区 | 最新帖子   今日:0 | 昨日:0 | 最高日:69 | 主题:208 | 帖子:913 | 会员:19075 | 新会员:gu789967

» ..-== General ==-..   分栏版主：alexa

论坛	主题 / 文章	最后发表	版主
News Articles and Updates {新闻发布} Share all the latest News, infosec and hacking in this forum	25 / 86	谷歌将与美国国家 .. alexa [2010-02-07 12:38]	alexa
Tutorials {资料搜集} Anything related to tutorials papers or pdfs about, Malwares, Worms, bots, rootkits, rats, etc...	11 / 19	【zz】IP地址反查， .. alexa [2010-01-01 12:27]	alexa

» 技术交流   分栏版主：t-bag shalyse shine killer 0x001

论坛	主题 / 文章	最后发表	版主
公告	0 / 0		shalyse
web hacking	17 / 34	Re:whole SQL inj(enjoy .. it1806022 [2009-12-11 15:44]	shalyse
club Show Show ur u want to show~bugs shells ponits some pocs etcs..	32 / 281	Re:pchome.com.tw 猪肉 .. alexa [2010-02-07 12:23]	shalyse
Exploits, bugs,0days And POC's Made By everyone	29 / 140	Internet Explorer (6/7) .. t-bag [2010-02-16 11:03]	shalyse
Hacker's video	5 / 51	Re:Win Local Exp With NC: .. dhzhi777 [2010-02-04 15:32]	shine
Security Tools	7 / 35	Re:【zz】MSE简体中 .. it1806022 [2009-12-11 15:44]	alexa
wireless Security -=- Bluetooth Security -=- Routers	8 / 18	Compiz Fusion on BT 4 bet .. t-bag [2009-09-26 14:38]	alexa
Packers & Crypters & Protectors & Related sources Everything you need to hide your files and make them undetected by antivirus	3 / 5	Cyber-Sec Crypter t-bag [2010-01-09 21:09]	puppet alexa

» ..== Programming & Source Code ==..   分栏版主：0x001 killer shine

论坛	主题 / 文章	最后发表	版主
PHP	4 / 6	php control Registry with .. killer [2010-01-27 12:40]	t-bag
Perl & Python & Ruby	12 / 43	Oracle 11g Database passw .. t-bag [2010-02-03 14:35]	killer
C# & VB.NET	5 / 24	Re:GoolagScanner_1.0.0.41 ghij040 [2009-11-27 23:44]	0x001

Image Supplied By DomainTools.com

Ydteam website as it was in 2010

Another trace to the source of attack is based on email sender IP address. The emails were sent from 118.142.11.114. According to robtex.com, there are 2 domain names that share this IP:

pad62.com
ru.pad62.com

Pad62.com was created in 2011-06-05, on the date of registration if had non-protected WHOIS information, according to domaintools.com:

Registrant: ji shao
Xuan Die Xiao Jie 418 Kao
peng hu, xiang gang 064562
China

Registered through: GoDaddy.com, Inc.
Domain Name: PAD62.COM
Created on: 05-Jun-11
Expires on: 05-Jun-12
Last Updated on: 05-Jun-11

Administrative Contact:
shao, ji huisengaunr@sina.com
Xuan Die Xiao Jie 418 Kao
peng hu, xiang gang 064562
China
1-330-040-0367

We checked which other domains are associated with the WHOIS information above and found the following domain names:

100-d.com
sm08.com
cx-cx.com
6-pro.com
aohoe.info
besheo.info
dyyerre.info
jiaoyouliaotian.org
tao5178.info

One more route is to check the C&C of the initial dropper/downloader module. This was **tank.hja63.com.** Acccording to domain tools, hja63.com had non-protected WHOIS information in 2011:

Registrant: ji shao
Xuan Die Xiao Jie 418 Kao
peng hu, xiang gang 064562
China

Registered through: GoDaddy.com, Inc.
Domain Name: HJA63.COM
Created on: 05-Jun-11
Expires on: 05-Jun-12
Last Updated on: 05-Jun-11

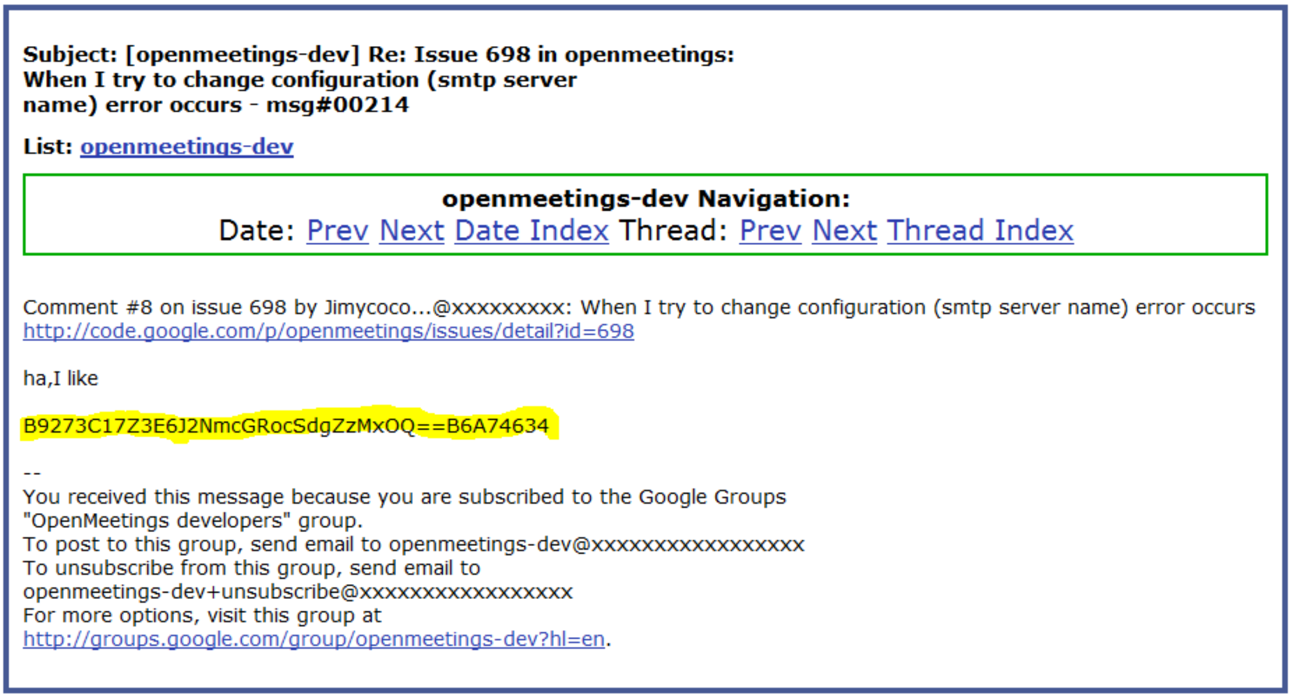
Administrative Contact:
shao, ji huisengaunr@sina.com
Xuan Die Xiao Jie 418 Kao
peng hu, xiang gang 064562
China
1-330-040-0367

When we checked, **tank.hja63.com** resolved to **173.234.184.45** (owned by DiaHosting Limited, USA), while hja63.com resolved to 68.178.232.100 (GoDaddy ISP server).

Bot Control Messages On Public Resources

Analysis of the file c_20100.nls revealed additional information leading to probable attackers. Looking for identifiers (used as message boundaries, or delimiters) ***B9273C17*** and ***B6A74634*** specified in this malicious file on Internet we found the following pages where the attackers left messages for the bots:

http://osdir.com/ml/openmeetings-dev/2011-10/msg00214.html
http://osdir.com/ml/openmeetings-dev/2011-10/msg00215.html
http://osdir.com/ml/openmeetings-dev/2011-10/msg00241.html



Subject: [openmeetings-dev] Re: Issue 698 in openmeetings: When I try to change configuration (smtp server name) error occurs - msg#00214

List: openmeetings-dev

openmeetings-dev Navigation:
Date: Prev Next Date Index Thread: Prev Next Thread Index

Comment #8 on issue 698 by Jimycoco...@xxxxxxxxx: When I try to change configuration (smtp server name) error occurs
http://code.google.com/p/openmeetings/issues/detail?id=698

ha,I like

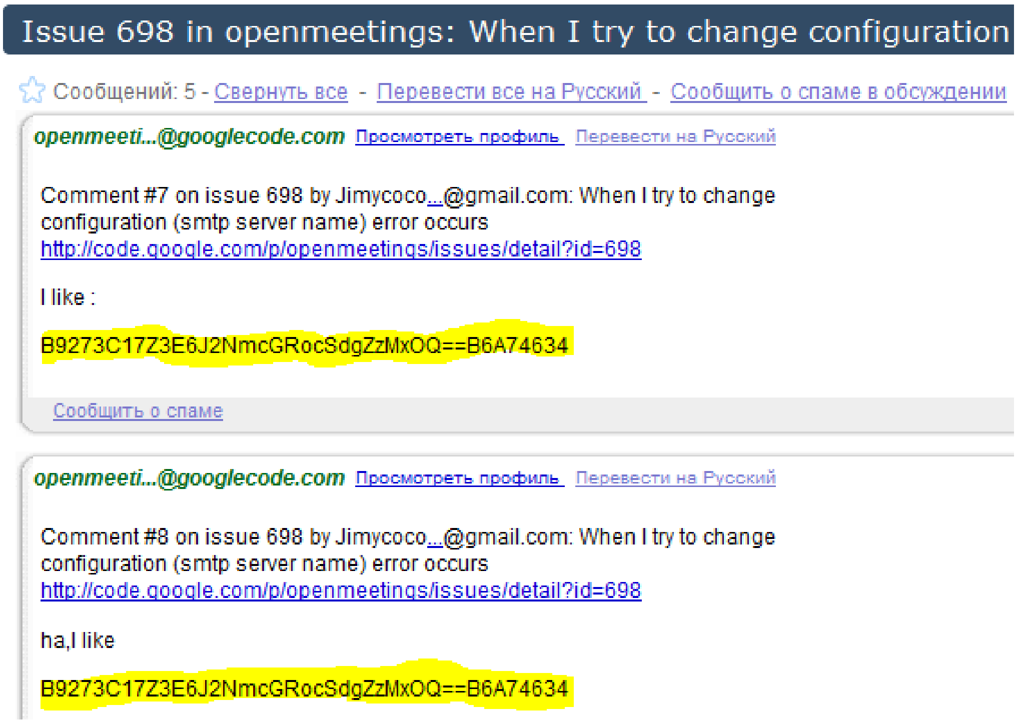
B9273C17Z3E6J2NmcGRocSdgZzMxOQ==B6A74634

--
You received this message because you are subscribed to the Google Groups
"OpenMeetings developers" group.
To post to this group, send email to openmeetings-dev@xxxxxxxxxxxxxxxx
To unsubscribe from this group, send email to
openmeetings-dev+unsubscribe@xxxxxxxxxxxxxxxx
For more options, visit this group at
http://groups.google.com/group/openmeetings-dev?hl=en.

An encoded C&C address for a bot on a public webpage

Another place of just mentioned forum thread:

https://groups.google.com/group/openmeetings-dev/browse_thread/thread/ccfeb8242a4f11ec/a700f22be192482a?show_docid=a700f22be192482a&pli=1
https://groups.google.com/group/openmeetings-dev/tree/browse_frm/month/2011-10/a8509400cef9a8ac?rnum=221&_done=%2Fgroup%2Fopenmeetings-dev%2Fbrowse_frm%2Fmonth%2F2011-10%3F



Issue 698 in openmeetings: When I try to change configuration

Сообщений: 5 - Свернуть все - Перевести все на Русский - Сообщить о спаме в обсуждении

openmeeti...@googlecode.com Просмотреть профиль Перевести на Русский

Comment #7 on issue 698 by Jimycoco...@gmail.com: When I try to change configuration (smtp server name) error occurs
http://code.google.com/p/openmeetings/issues/detail?id=698

I like :

B9273C17Z3E6J2NmcGRocSdgZzMxOQ==B6A74634

Сообщить о спаме

openmeeti...@googlecode.com Просмотреть профиль Перевести на Русский

Comment #8 on issue 698 by Jimycoco...@gmail.com: When I try to change configuration (smtp server name) error occurs
http://code.google.com/p/openmeetings/issues/detail?id=698

ha,I like

B9273C17Z3E6J2NmcGRocSdgZzMxOQ==B6A74634

Some more server addresses for the bot

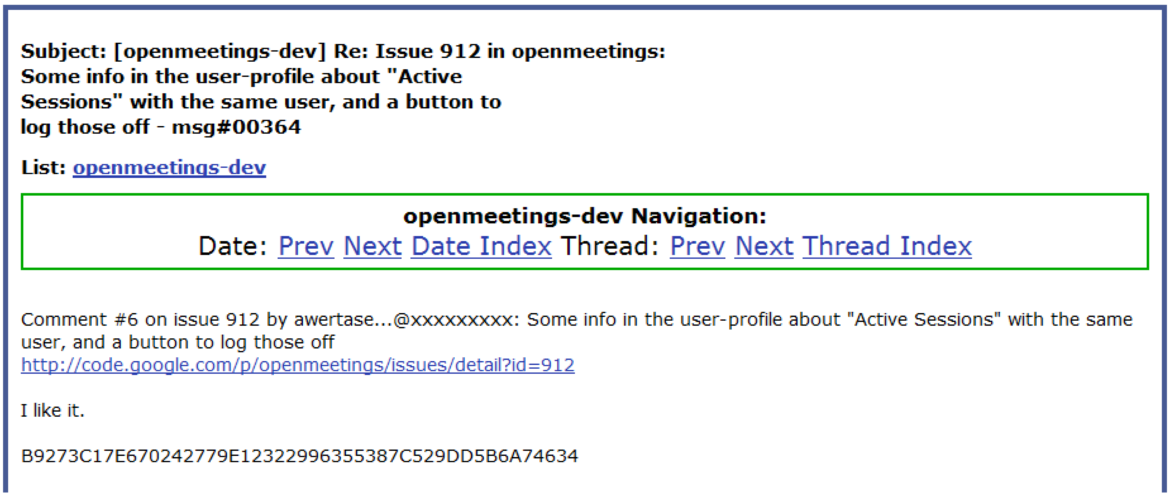
Here, we see these emails used as commenters' identifiers:

Jimycoco…@gmail.com
awertase…@yahoo.com

Jimycoco…@gmail.com most probably refers to *Jimycocowell* which is a username that pops up further.

Searching for "*awertase*" brought another forum thread where ciphered data for the same bot appeared:

http://osdir.com/ml/openmeetings-dev/2011-09/msg00364.html



Subject: [openmeetings-dev] Re: Issue 912 in openmeetings: Some info in the user-profile about "Active Sessions" with the same user, and a button to log those off - msg#00364

List: openmeetings-dev

openmeetings-dev Navigation:
Date: Prev Next Date Index Thread: Prev Next Thread Index

Comment #6 on issue 912 by awertase...@xxxxxxxxx: Some info in the user-profile about "Active Sessions" with the same user, and a button to log those off
http://code.google.com/p/openmeetings/issues/detail?id=912

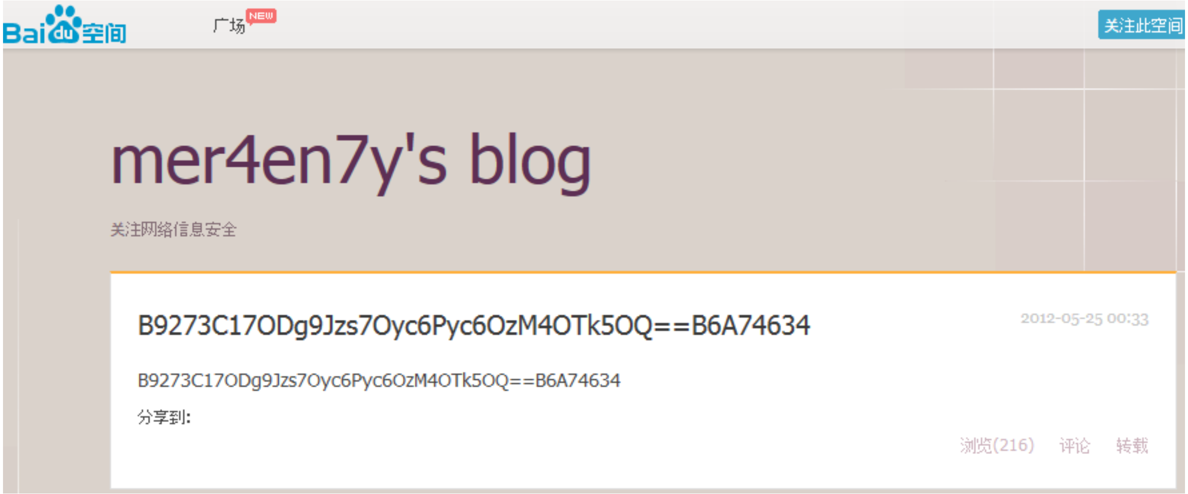
I like it.

B9273C17E670242779E12322996355387C529DD5B6A74634

Yet another message for bots from awertase...

The full email behind awertase...@xxxxxxxx seems to be awertasegfae@yahoo.com according to http://awertasegfae.blogspot.ru/2011/10/first-test.html

http://hi.baidu.com/alonecode/item/6936f85a3d98ce3533e0a9ed



mer4en7y's blog

关注网络信息安全

B9273C17ODg9Jzs7Oyc6Pyc6OzM4OTk5OQ==B6A74634

2012-05-25 00:33

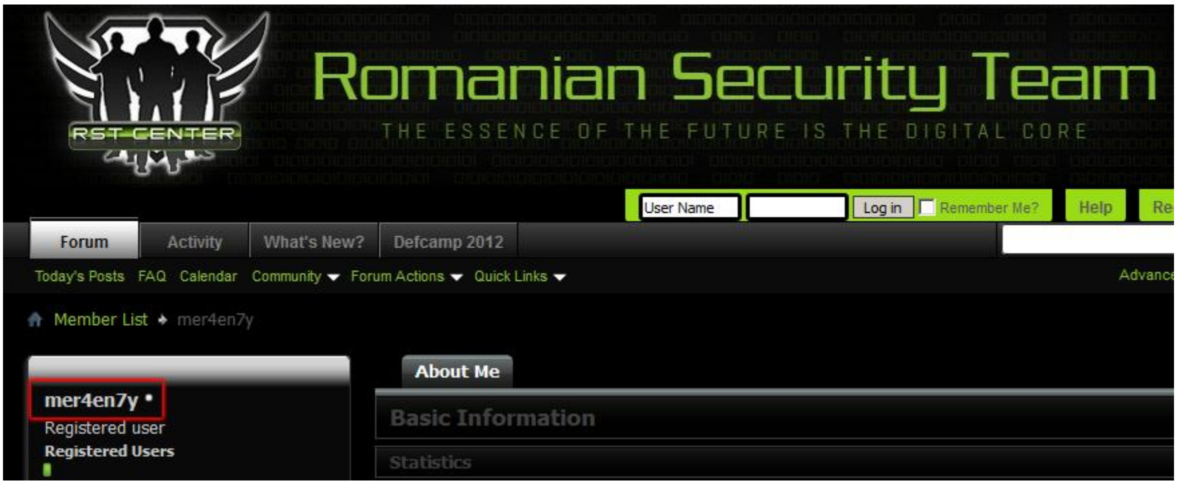
B9273C17ODg9Jzs7Oyc6Pyc6OzM4OTk5OQ==B6A74634

分享到:

Another webpage with message for bots

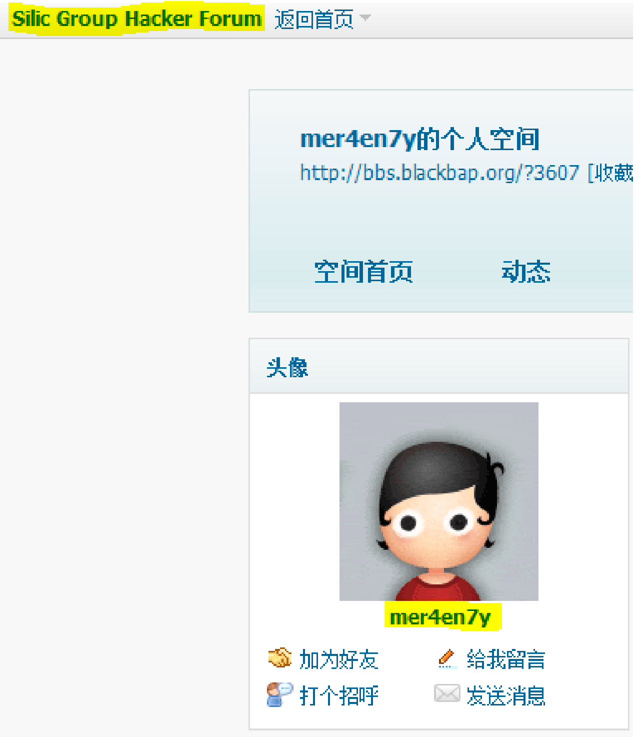
According to Figure 32, "*mer4en7y*" and "*alonecode*" (from the URL of the page) are nicknames which are related to the user of the Baidu blog platform where messages for a bot were left. Google Search for the nickname "*mer4en7y*" returned 5490 results. This is a very active user that posts messages for this type of bot. The first results lead to hacker forums and IT-security specific web-platforms. The same nickname has appeared on a well-known Romanian Security Team forum.

Mer4en7y Individual Activity



mer4en7y •
Registered user
Registered Users

mer4en7y username at Romanian hackers forum



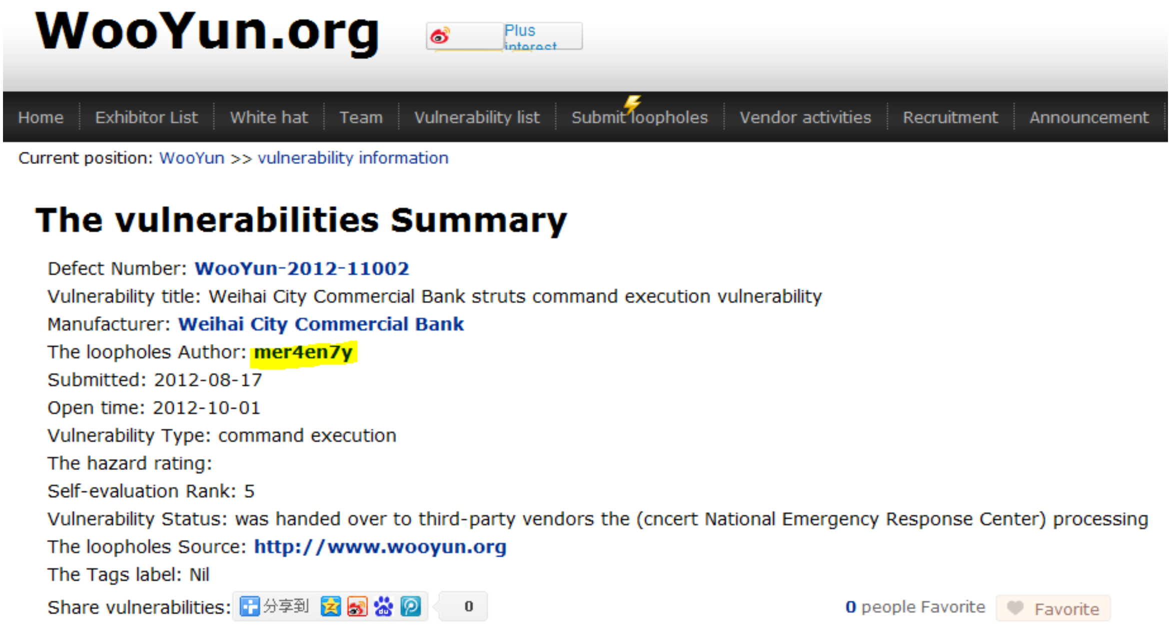
Silic Group Hacker Forum 返回首页

mer4en7y的个人空间
http://bbs.blackbap.org/?3607 [收藏

Mer4en7y at Silic Group Hacker Forum

According to the following, Mer4en7y submitted a vulnerability found in Weihai City Commercial Bank system:

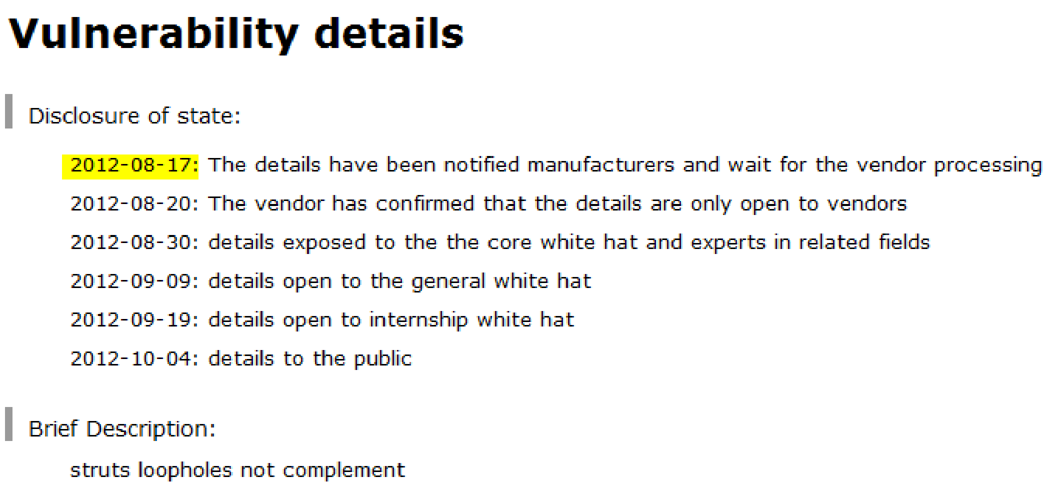
http://wooyun.org/bugs/wooyun-2010-011002



Current position: WooYun >> vulnerability information

The vulnerabilities Summary

Defect Number: WooYun-2012-11002
Vulnerability title: Weihai City Commercial Bank struts command execution vulnerability
Manufacturer: Weihai City Commercial Bank
The loopholes Author: mer4en7y
Submitted: 2012-08-17
Open time: 2012-10-01
Vulnerability Type: command execution
The hazard rating:
Self-evaluation Rank: 5
Vulnerability Status: was handed over to third-party vendors the (cncert National Emergency Response Center) processing
The loopholes Source: http://www.wooyun.org
The Tags label: Nil
Share vulnerabilities: 分享到 0



Vulnerability details

Disclosure of state:

2012-08-17: The details have been notified manufacturers and wait for the vendor processing
2012-08-20: The vendor has confirmed that the details are only open to vendors
2012-08-30: details exposed to the the core white hat and experts in related fields
2012-09-09: details open to the general white hat
2012-09-19: details open to internship white hat
2012-10-04: details to the public

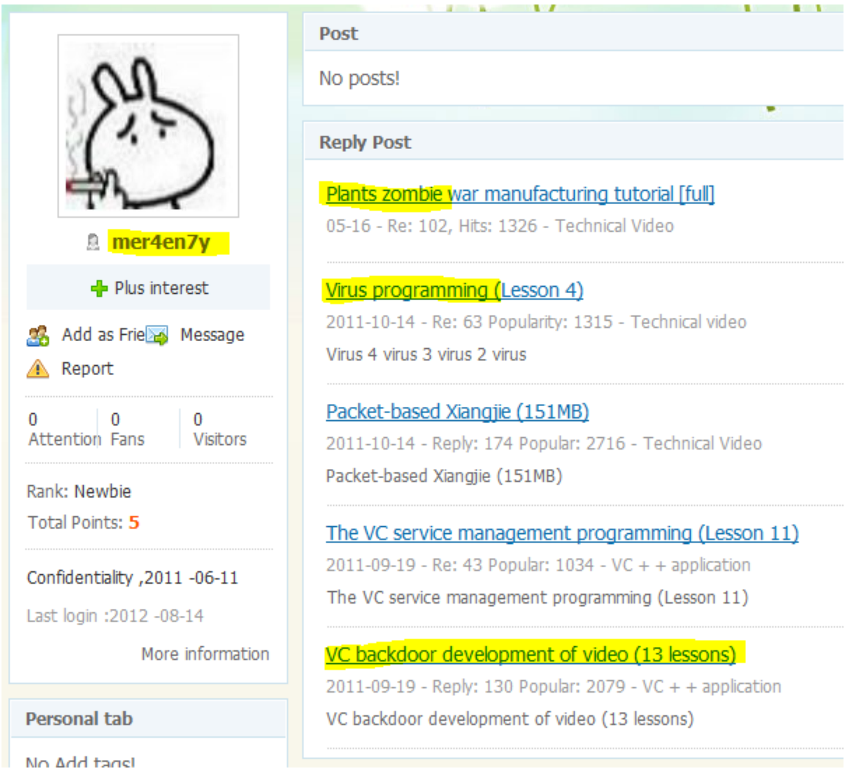
Brief Description:

struts loopholes not complement

Mer4en7y's activity on vulnerability research

Favorite videos and tutorials of Mer4en7y:

http://www.tdcqjslt.com/u.php?uid=1918



mer4en7y

0 Attention 0 Fans 0 Visitors

Rank: Newbie
Total Points: 5

Confidentiality ,2011 -06-11
Last login :2012 -08-14

Personal tab
No Add tags!

Post
No posts!

Reply Post

Plants zombie war manufacturing tutorial [full]
05-16 - Re: 102, Hits: 1326 - Technical Video

Virus programming (Lesson 4)
2011-10-14 - Re: 63 Popularity: 1315 - Technical video
Virus 4 virus 3 virus 2 virus

Packet-based Xiangjie (151MB)
2011-10-14 - Reply: 174 Popular: 2716 - Technical Video
Packet-based Xiangjie (151MB)

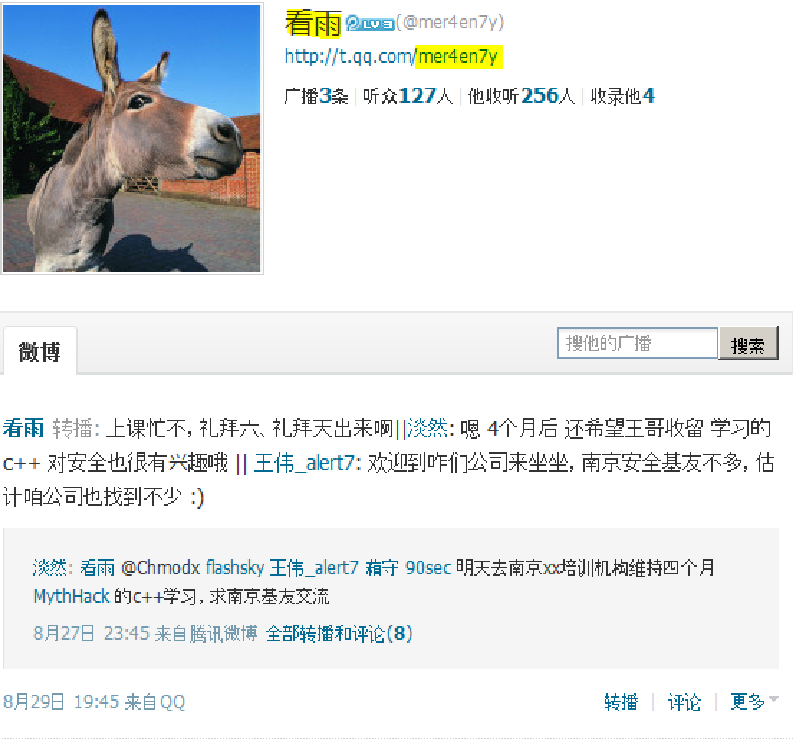
The VC service management programming (Lesson 11)
2011-09-19 - Re: 43 Popular: 1034 - VC + + application
The VC service management programming (Lesson 11)

VC backdoor development of video (13 lessons)
2011-09-19 - Reply: 130 Popular: 2079 - VC + + application
VC backdoor development of video (13 lessons)

Mer4en7y's favorites confirm malware-related activities

Mer4en7y's micro-blogging page at *t.qq.com:* http://t.qq.com/mer4en7y

Alias of that profile is translated as "*watching a rain*".



看雨 (@mer4en7y)
http://t.qq.com/mer4en7y
广播3条 | 听众127人 | 他收听256人 | 收录他4

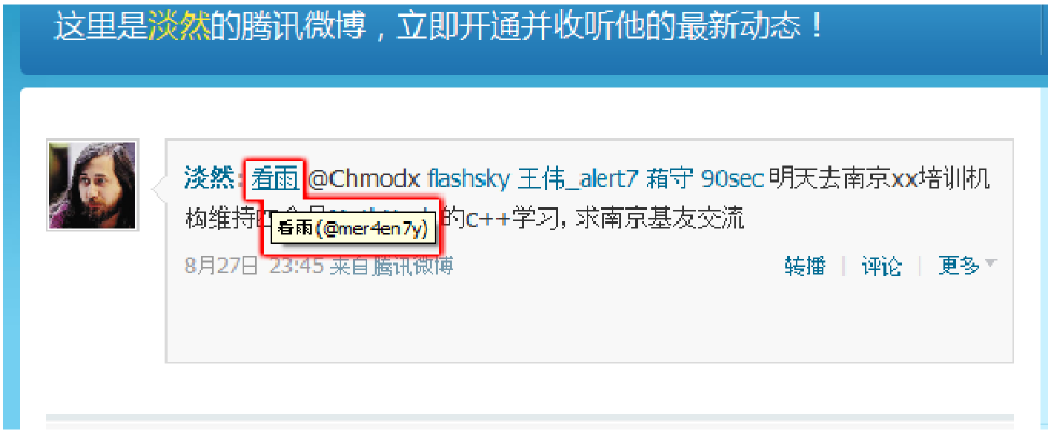
看雨 转播：上课忙不，礼拜六、礼拜天出来啊||淡然：嗯 4个月后 还希望王哥收留 学习的c++ 对安全也很有兴趣哦 || 王伟_alert7：欢迎到咋们公司来坐坐，南京安全基友不多，估计咱公司也找到不少 :)

淡然：看雨 @Chmodx flashsky 王伟_alert7 箱守 90sec 明天去南京xx培训机构维持四个月 MythHack 的c++学习，求南京基友交流
8月27日 23:45 来自腾讯微博 全部转播和评论(8)

8月29日 19:45 来自QQ

Mer4en7y's microblogging profile

A user with nickname “*d4nr4n*” (http://t.qq.com/d4nr4n) is posting a message where *mer4en7y* is mentioned:



这里是淡然的腾讯微博，立即开通并收听他的最新动态！

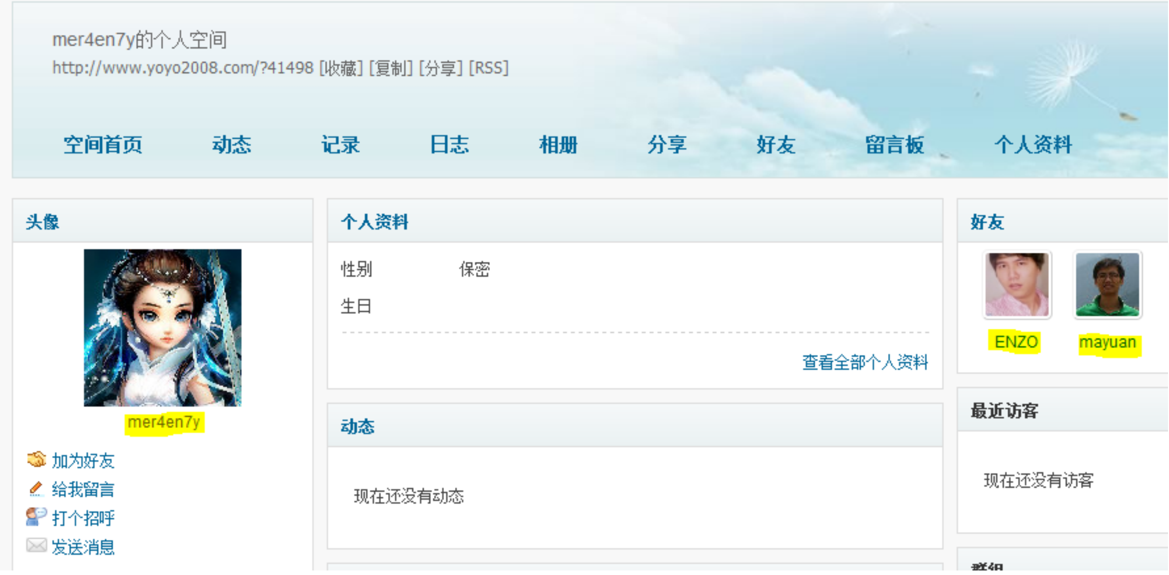
淡然：看雨 @Chmodx flashsky 王伟_alert7 箱守 90sec 明天去南京xx培训机构维持 看雨(@mer4en7y) 的c++学习，求南京基友交流
8月27日 23:45 来自腾讯微博

Mer4en7y's relation to Nanjing

Google translation: “*%mentioned individuals% go to Nanjing tomorrow xx training institutions to maintain four months … C++ learning, seeking Nanjing-based friends of the exchange*”

Mer4en7y at *yoyo2008.com*:

http://www.yoyo2008.com/home.php?mod=space&uid=41498


mer4en7y的个人空间
http://www.yoyo2008.com/?41498 [收藏] [复制] [分享] [RSS]

头像

mer4en7y

个人资料

性别	保密
生日	

查看全部个人资料

动态

现在还没有动态

好友

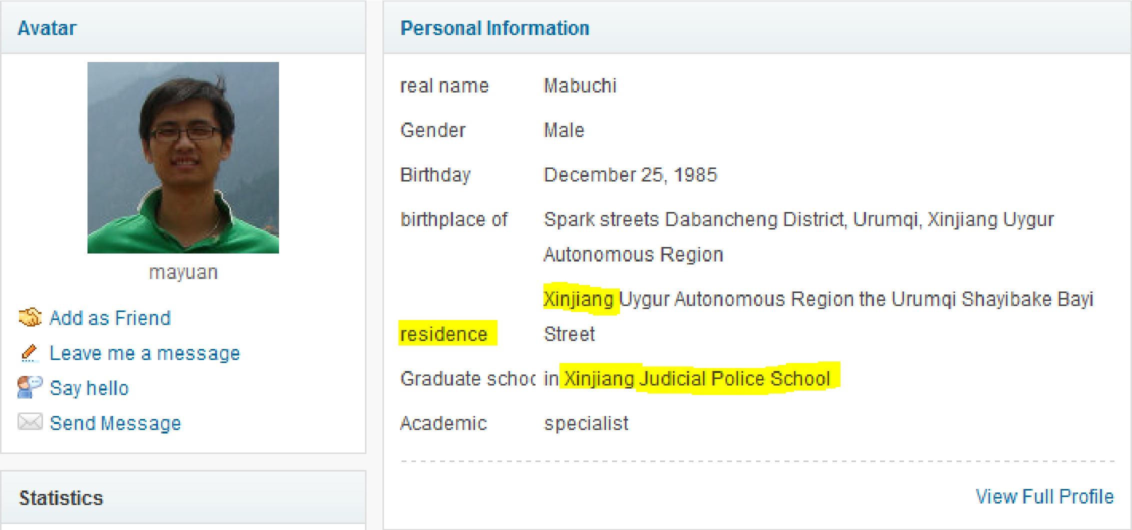
ENZO mayuan

最近访客

现在还没有访客

Mer4en7y profile at yoyo2008.com

One of two friends of **Mer4en7y** in yoyo2008 social network is a user named "*mayuan*" which seems to be from Xinjiang and a graduate of Judicial Police School according to shared private information out there:


Avatar

mayuan

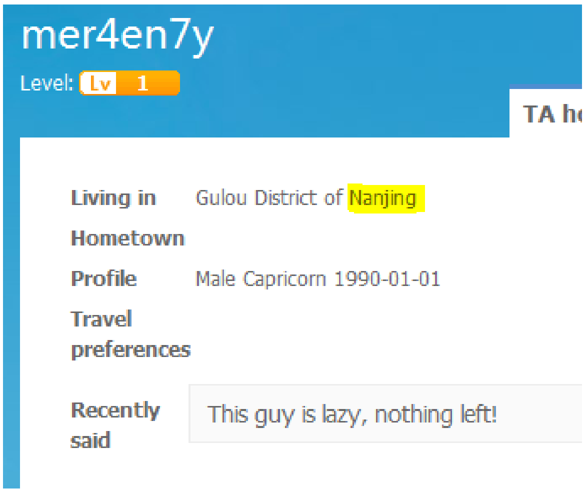
Personal Information

real name	Mabuchi
Gender	Male
Birthday	December 25, 1985
birthplace of	Spark streets Dabancheng District, Urumqi, Xinjiang Uygur Autonomous Region
residence	Xinjiang Uygur Autonomous Region the Urumqi Shayibake Bayi Street
Graduate schoc	in Xinjiang Judicial Police School
Academic	specialist

View Full Profile

Mer4en7y's contact profile at yoyo2008.com

http://u.pintour.com/uid-b1bf56e230cc42d9bfa003a7718888d2/



mer4en7y

Level: Lv 1

TA h

Living in Gulou District of Nanjing

Hometown

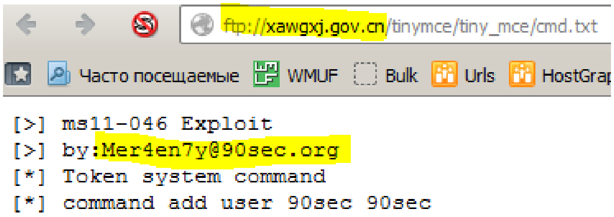
Profile Male Capricorn 1990-01-01

Travel preferences

Recently said This guy is lazy, nothing left!

Another Mer4en7y profile show Nanjing as a hometown

Mer4en7y's exploit has been involved in the penetration of public radio service ftp server (according to WHOIS information this domain belongs to Xi'an Municipal Bureau of Radio and Television).



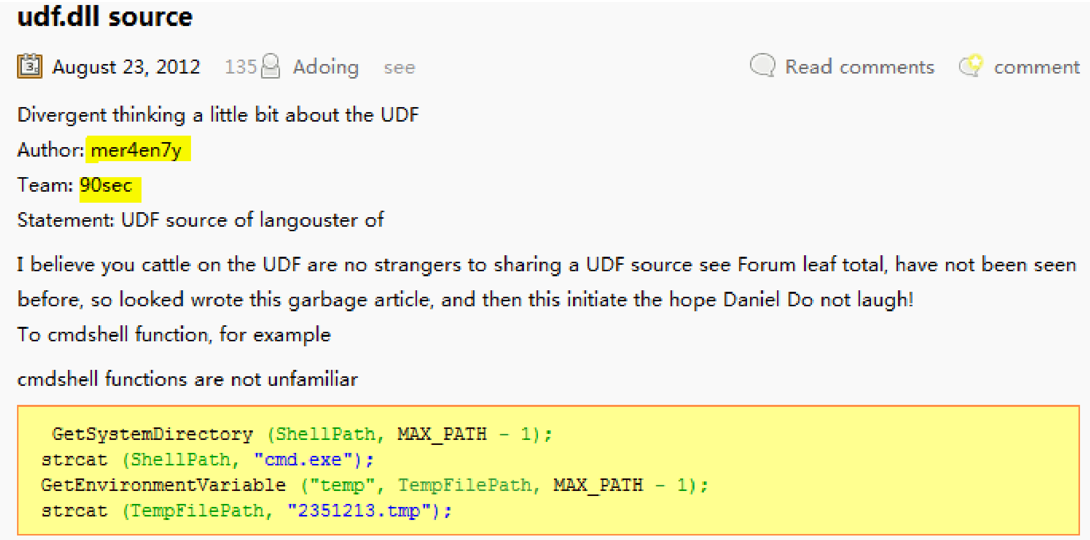
```
[>] ms11-046 Exploit
[>] by:Mer4en7y@90sec.org
[*] Token system command
[*] command add user 90sec 90sec
```

A trace of cyberattack based on Mer4en7y's code

As we can see here Mer4en7y had an email address associated with 90sec hackers team.

Another reference on the net shows that Mer4en7y is after sourcecode of proprietary products (probably udf.dll from Roxio Inc):

http://www.uedbox.com/udf-dll-source/



udf.dll source

August 23, 2012 135 Adoing see Read comments comment

Divergent thinking a little bit about the UDF

Author: mer4en7y

Team: 90sec

Statement: UDF source of langouster of

I believe you cattle on the UDF are no strangers to sharing a UDF source see Forum leaf total, have not been seen before, so looked wrote this garbage article, and then this initiate the hope Daniel Do not laugh!

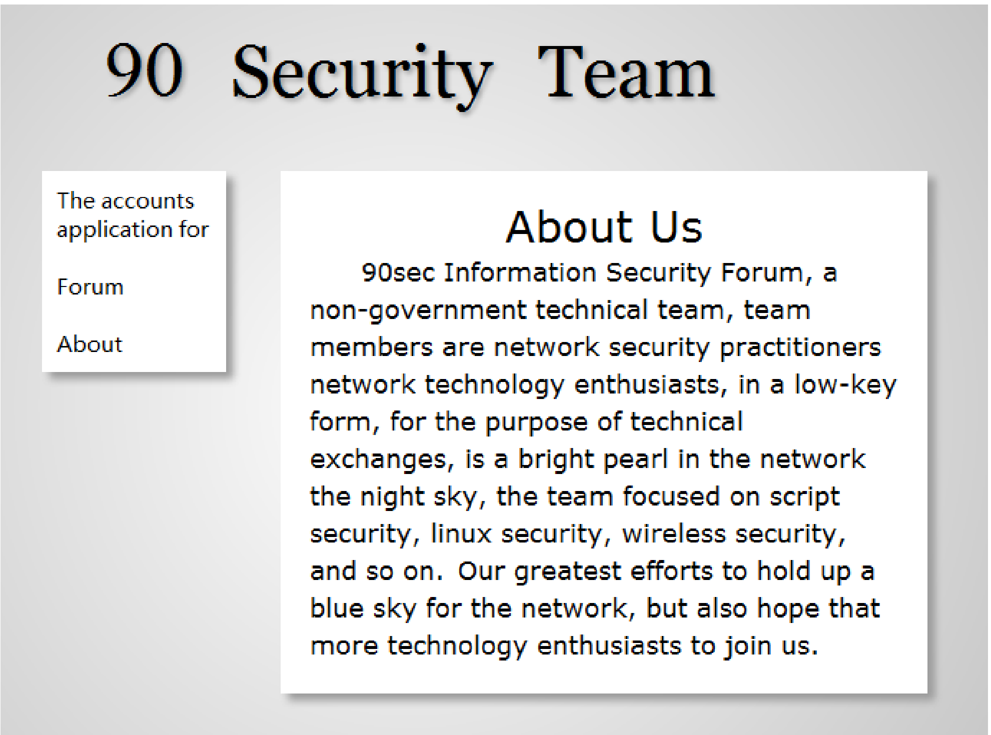
To cmdshell function, for example

cmdshell functions are not unfamiliar

```
 GetSystemDirectory (ShellPath, MAX_PATH - 1);
strcat (ShellPath, "cmd.exe");
GetEnvironmentVariable ("temp", TempFilePath, MAX_PATH - 1);
strcat (TempFilePath, "2351213.tmp");
```

Mer4en7y discussing udf.dll source-code and cmdshell

The following confirms that **Mer4en7y is a member of 90sec group.** The group website is located at http://www.90sec.org/:



90 Security Team

The accounts application for

Forum

About

About Us

90sec Information Security Forum, a non-government technical team, team members are network security practitioners network technology enthusiasts, in a low-key form, for the purpose of technical exchanges, is a bright pearl in the network the night sky, the team focused on script security, linux security, wireless security, and so on. Our greatest efforts to hold up a blue sky for the network, but also hope that more technology enthusiasts to join us.

90Sec team about-page

Mer4en7y replies on job offer posted at *90sec* forum (someone wanted to hire computer exerts with **very special knowledge**):

https://forum.90sec.org/viewthread.php?action=printable&tid=2012

Rough translation of job offer from Chinese:

"Subject: Looking for information security researcher
From: Southland sword

Time: 2012-04-06 00:38

Subject: Security researcher job
Responsibilities:

1. *Full target penetration alone or with a team depending on available resources;*
2. *Penetration testing report and recommendations*

Technical requirements:

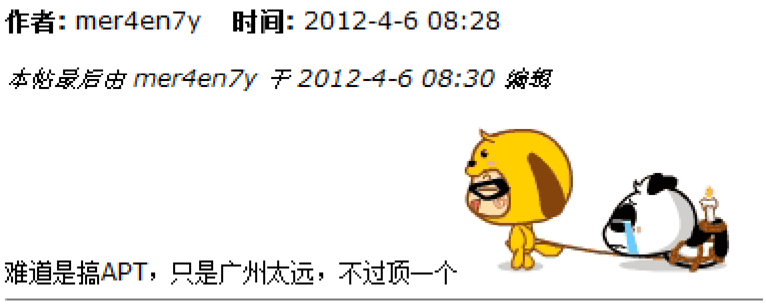
1. *Knowledge of penetration testing, methods, processes, proficiency in a variety of penetration testing tools;*
2. *Knowledge of common Web development languages (asp, php, jsp), experience with SQL-injection, XSS, common websecurity exploits and patches;*
3. *Experience with all kinds of operating systems and databases for common security vulnerabilities;*
4. *Good verbal and written language skills ;*
5. *Be able to work in a team; individuals who lose trust, do not listen to the teamleader and not accepting the rules will be kicked out;*

Work Location: Guangdong (OR Guangzhou Shenzhen)

Baochibaozhu package, Relatively free playing time.
Salary: monthly allocation of the total amount of work and cooperation share more than 1W.
Vacancies: 5 people
For candidates: first contact me (preferably work resume), after my check the resume will be passed to the head coordinator for arranging a personal meeting.
Salary: free meal and apartments, office location is in a senior villa suite of 200 square meters, computers are available but please bring your own hard drive with environment and tools you are familiar with. Even a single completed project will provide you with money for your monthly expenses.
Powerful background. No comments!

Tho who are competent, please contact:
Email: Infosec@cntv.cn QQ: admin@inessus.com"

And *Mer4en7y*'s replied to this job offer:



作者: mer4en7y **时间:** 2012-4-6 08:28

本帖最后由 mer4en7y 于 2012-4-6 08:30 编辑

难道是搞APT，只是广州太远，不过顶一个

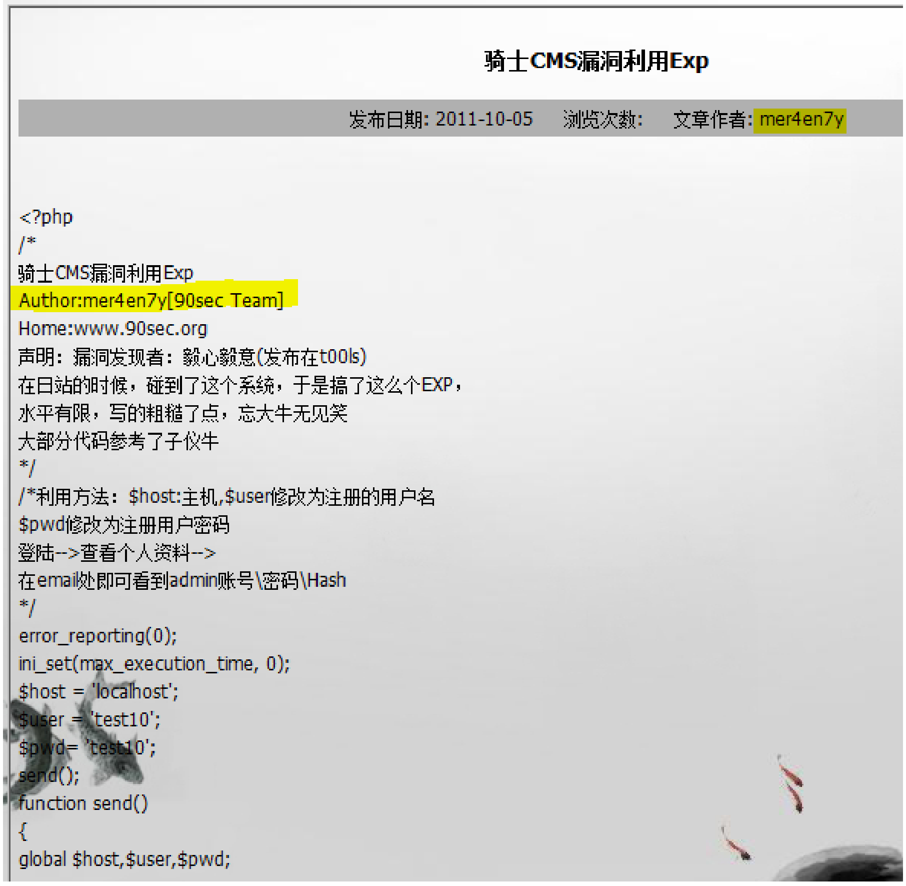
Mer4en7y's comment about job offer

Which can be translated as: "*Aren't you recruiting people for APT? Guangzhou is too far, but anyway I support it*".

There are some interesting comments in the mentioned forum thread regarding reference "Powerful background" in job offer. People in the thread speculated that it could mean the work is supported by the government.

Mer4en7y is publishing an exploit:

http://www.hackqing.com/index.asp?FoxNews=129.html



骑士CMS漏洞利用Exp

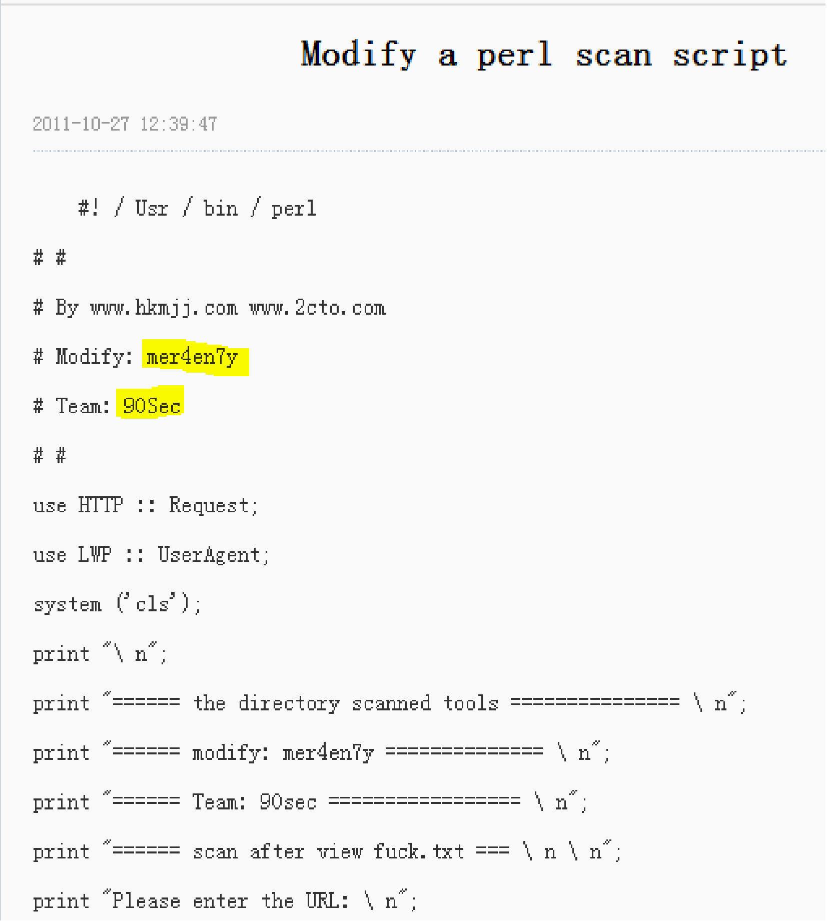
发布日期: 2011-10-05　浏览次数:　文章作者: mer4en7y

```
<?php
/*
骑士CMS漏洞利用Exp
Author:mer4en7y[90sec Team]
Home:www.90sec.org
声明：漏洞发现者：毅心毅意(发布在t00ls)
在日站的时候，碰到了这个系统，于是搞了这么个EXP，
水平有限，写的粗糙了点，忘大牛无见笑
大部分代码参考了子仪牛
*/
/*利用方法：$host:主机,$user修改为注册的用户名
$pwd修改为注册用户密码
登陆-->查看个人资料-->
在email处即可看到admin账号\密码\Hash
*/
error_reporting(0);
ini_set(max_execution_time, 0);
$host = 'localhost';
$user = 'test10';
$pwd= 'test10';
send();
function send()
{
global $host,$user,$pwd;
```

Mer4en7y's exploit code in PHP

Mer4en7y published a modified Perl script for network scan:

http://www.2cto.com/kf/201110/109200.html



Modify a perl scan script

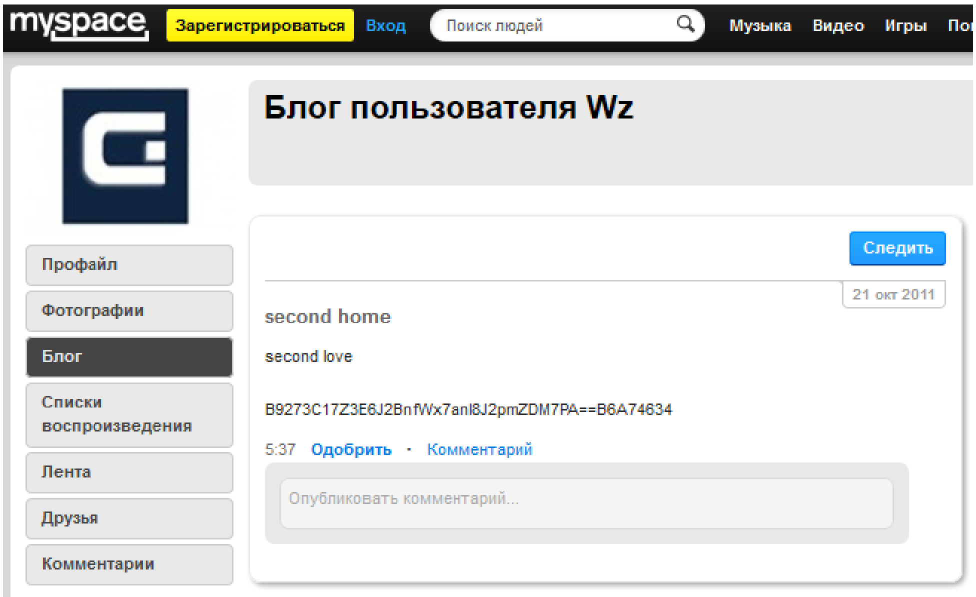
2011-10-27 12:39:47

```
   #! / Usr / bin / perl
# #
# By www.hkmjj.com www.2cto.com
# Modify: mer4en7y
# Team: 90Sec
# #
use HTTP :: Request;
use LWP :: UserAgent;
system ('cls');
print ″\ n″;
print ″====== the directory scanned tools =============== \ n″;
print ″====== modify: mer4en7y ============== \ n″;
print ″====== Team: 90sec ================= \ n″;
print ″====== scan after view fuck.txt === \ n \ n″;
print ″Please enter the URL: \ n″;
```

Mer4en7y's network scanner on Perl

Jimmycocowell Individual Activity

Let's continue with other places where delimiters have been found: https://www.myspace.com/574064782/blog



Блог пользователя Wz

21 окт 2011

second home

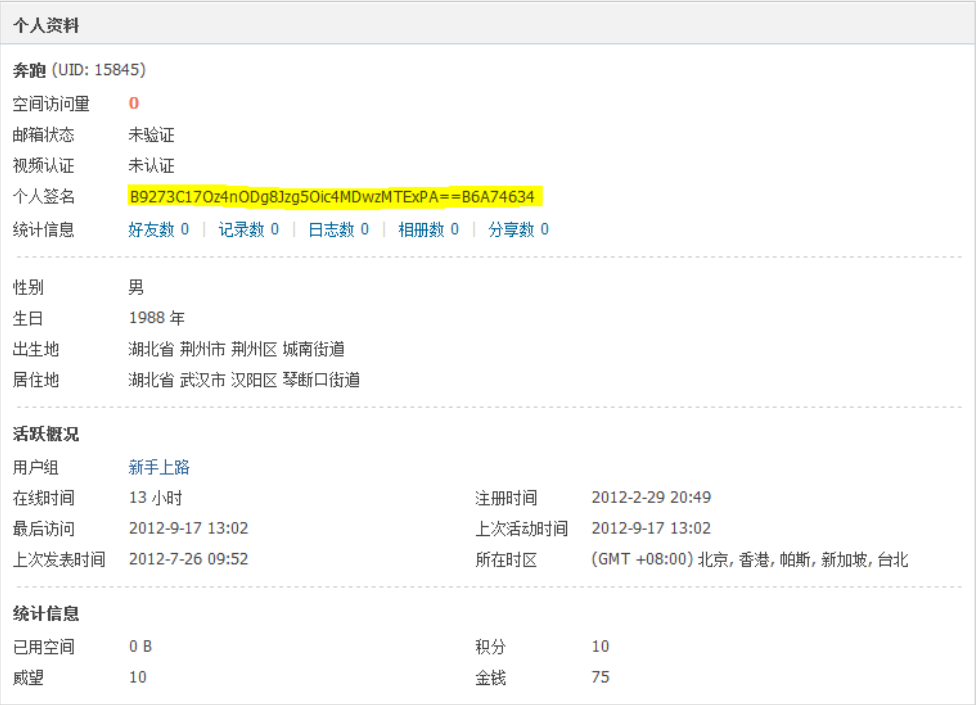
second love

B9273C17Z3E6J2BnfWx7anl8J2pmZDM7PA==B6A74634

5:37

Another bot control message by Wz

http://www.wuhanbike.net/home.php?mod=space&uid=15845&do=profile



个人资料

奔跑 (UID: 15845)

空间访问量	0
邮箱状态	未验证
视频认证	未认证
个人签名	B9273C17Oz4nODg8Jzg5Oic4MDwzMTExPA==B6A74634
统计信息	好友数 0 | 记录数 0 | 日志数 0 | 相册数 0 | 分享数 0

性别	男
生日	1988 年
出生地	湖北省 荆州市 荆州区 城南街道
居住地	湖北省 武汉市 汉阳区 琴断口街道

活跃概况

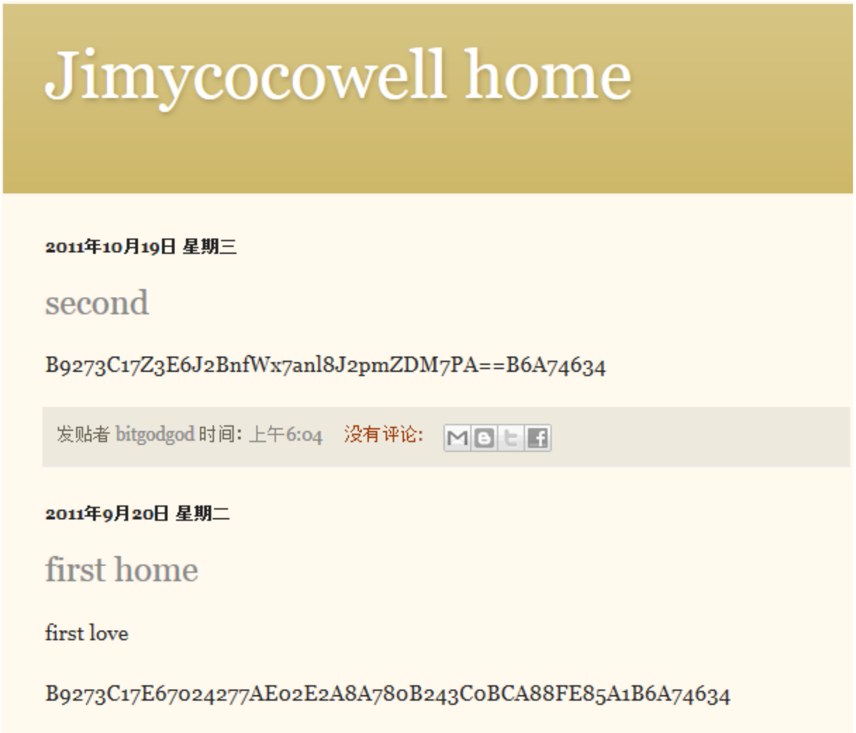
用户组	新手上路		
在线时间	13 小时	注册时间	2012-2-29 20:49
最后访问	2012-9-17 13:02	上次活动时间	2012-9-17 13:02
上次发表时间	2012-7-26 09:52	所在时区	(GMT +08:00) 北京, 香港, 帕斯, 新加坡, 台北

统计信息

已用空间	0 B	积分	10
威望	10	金钱	75

Another bot control message by 奔跑 (Run)

http://jimycocowell.blogspot.ru/



Jimycocowell home

2011年10月19日 星期三

second

B9273C17Z3E6J2BnfWx7anl8J2pmZDM7PA==B6A74634

发贴者 bitgodgod 时间: 上午6:04 没有评论:

2011年9月20日 星期二

first home

first love

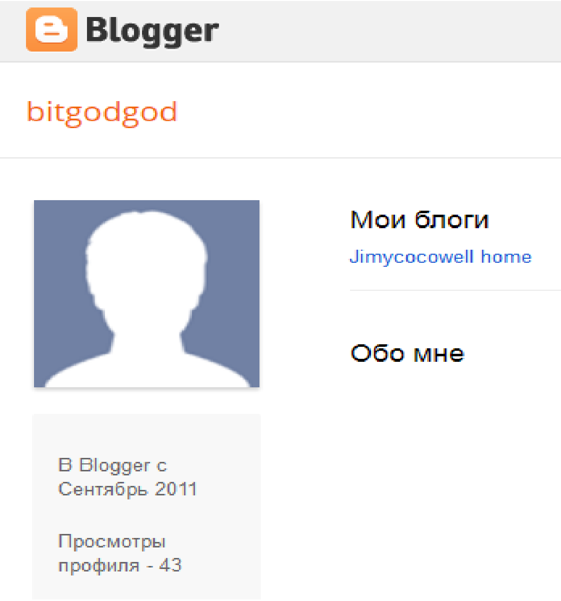
B9273C17E67024277AE02E2A8A780B243C0BCA88FE85A1B6A74634

Another bot control message by Jimmycocowell

The attacker left two messages. The very first one is labeled as "*first home*"/ "*first love*" and contains a ciphered C&C domain as described above, i.e. C&C domain is encrypted with RC4 algorithm and its hex binary value is presented in text format between delimiters.

But the next message dubbed "*second*" contains a ciphered C&C domain too but it is encoded in another way: The initial C&C domain is XORed with fixed byte value and the resulted data is transformed using BASE64 encoding. The resulted text is inserted between **the same** delimiters. By all appearances this method is used in the next version of the backdoor which is the subject of current research (see *c_20100.NLS)*. It is also possible that programs with support of either this or that encryption could be used simultaneously in the frame of one attack. Between all found messages for the bot the second type of messages (BASE64) is significantly prevalent.

A link to this "*Jimycocowell home*" is also present at following place of "*bitgodgod*" user: http://www.blogger.com/profile/06442609461818597659



Blogger

bitgodgod

Мои блоги

Jimycocowell home

Обо мне

В Blogger с
Сентябрь 2011

Просмотры
профиля - 43

Jimmycocowell registration date and alias

Bitgodgod and Bitbugbug

We have located one sample of Winnti malware with a hardcoded C&C: **mail.7niu.com**.
Domaintools information about the domain:

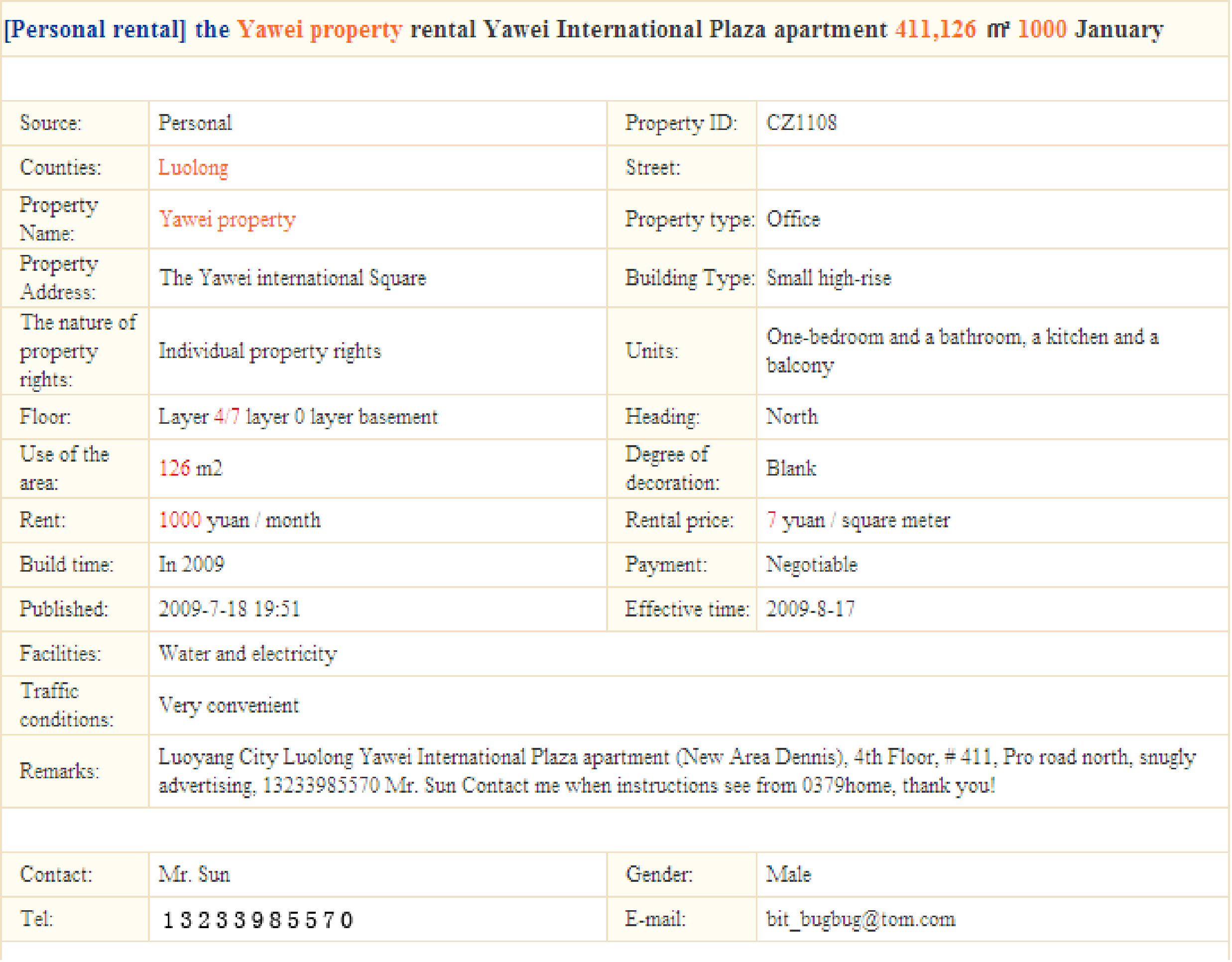
```
Domain Name     : 7niu.com
PunnyCode       : 7niu.com
Creation Date   : 2006-06-11 00:00:00
Updated Date    : 2012-01-27 21:35:57
Expiration Date : 2016-06-11 00:00:00

  Registrant:
  Organization   : qi tou niu
  Name           : xibei jiao
  Address        : beijing
  City           : beijing
  Province/State : Beijing
  Country        : CN
  Postal Code    : 100000
```

Administrative Contact:
Name : xibei jiao
Organization : qi tou niu
Address : beijing
City : beijing
Province/State : Beijing
Country : beijing
Postal Code : 100000
Phone Number : 86--1321333333
Fax : 86--010555555
Email : bit_bugbug@tom.com

Technical Contact:
Name : xibei jiao
Organization : qi tou niu
Address : beijing
City : beijing
Province/State : Beijing
Country : CN
Postal Code : 100000
Phone Number : 86--1321333333
Fax : 86--010555555
Email : rain@etang.com

You can see how similar “bitbugbug” and “bitgodgod”. Both are directly related to Winnti activity.
The email address “bit_bugbug@tom.com” also can be found on Chinese websites about home rentals:

http://oldhouse.0379home.com/RentView-1108.html



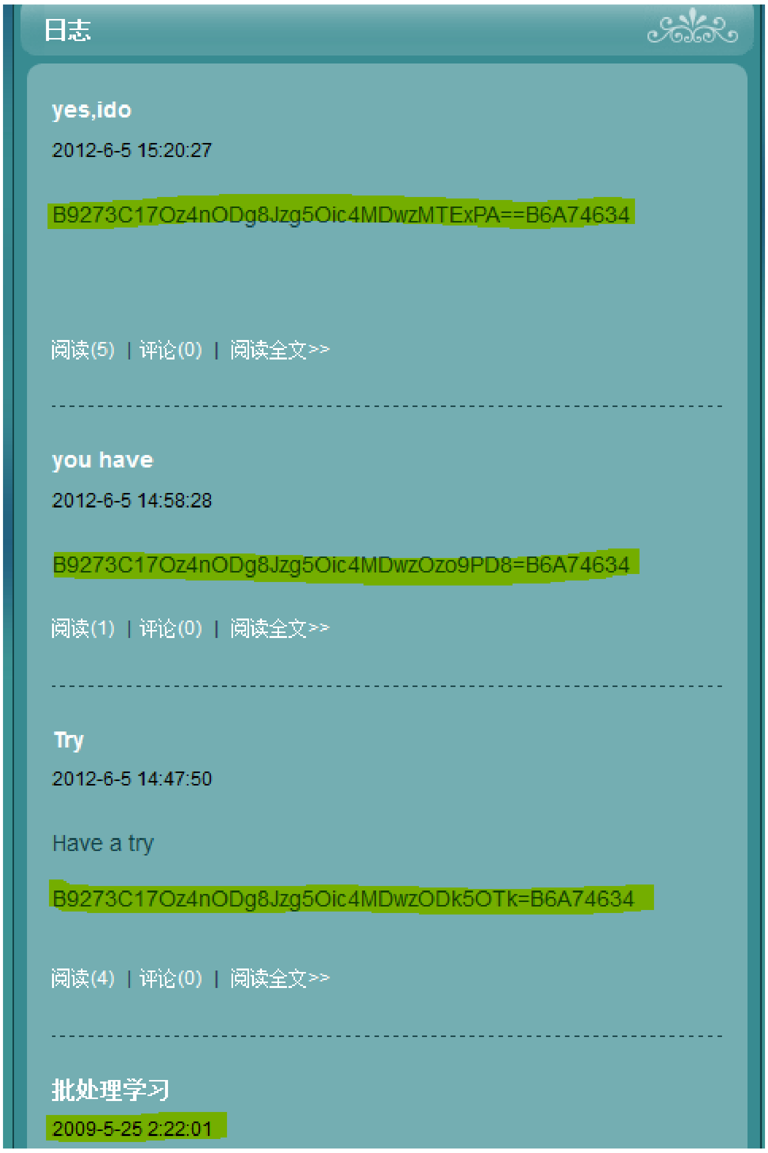
[Personal rental] the Yawei property rental Yawei International Plaza apartment 411,126 ㎡ 1000 January

Source:	Personal	Property ID:	CZ1108
Counties:	Luolong	Street:	
Property Name:	Yawei property	Property type:	Office
Property Address:	The Yawei international Square	Building Type:	Small high-rise
The nature of property rights:	Individual property rights	Units:	One-bedroom and a bathroom, a kitchen and a balcony
Floor:	Layer 4/7 layer 0 layer basement	Heading:	North
Use of the area:	126 m2	Degree of decoration:	Blank
Rent:	1000 yuan / month	Rental price:	7 yuan / square meter
Build time:	In 2009	Payment:	Negotiable
Published:	2009-7-18 19:51	Effective time:	2009-8-17
Facilities:	Water and electricity		
Traffic conditions:	Very convenient		
Remarks:	Luoyang City Luolong Yawei International Plaza apartment (New Area Dennis), 4th Floor, # 411, Pro road north, snugly advertising, 13233985570 Mr. Sun Contact me when instructions see from 0379home, thank you!		
Contact:	Mr. Sun	Gender:	Male
Tel:	13233985570	E-mail:	bit_bugbug@tom.com

Yang Individual Activity

We have located another individual calling himself Yang. He distributed bot control commands and was quite active on the internet as well.

http://yang8559420.blog.163.com/



日志

yes,ido

2012-6-5 15:20:27

B9273C17Oz4nODg8Jzg5Oic4MDwzMTExPA==B6A74634

阅读(5) | 评论(0) | 阅读全文>>

you have

2012-6-5 14:58:28

B9273C17Oz4nODg8Jzg5Oic4MDwzOzo9PD8=B6A74634

阅读(1) | 评论(0) | 阅读全文>>

Try

2012-6-5 14:47:50

Have a try

B9273C17Oz4nODg8Jzg5Oic4MDwzODk5OTk=B6A74634

阅读(4) | 评论(0) | 阅读全文>>

批处理学习

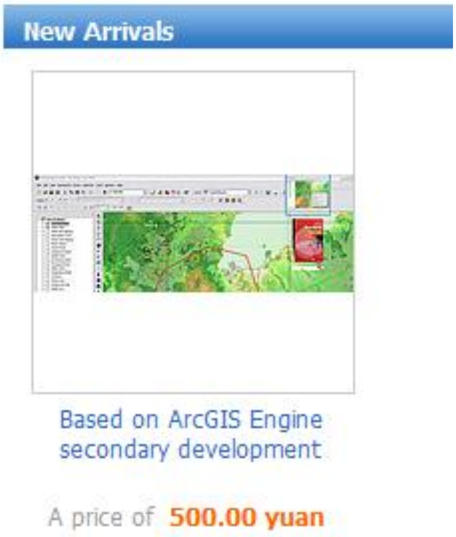
2009-5-25 2:22:01

Yang8559420 blog

Search for "*yang8559420*" brought some results:

Yang is a distributor of resources (maps or programs) for applications based on ArcGIS Engine (http://www.esri.com/software/arcgis/arcgisengine)

http://shop65775432.taobao.com/?spm=a1z0b.7.2-2442034955.3.rfLsIS



New Arrivals

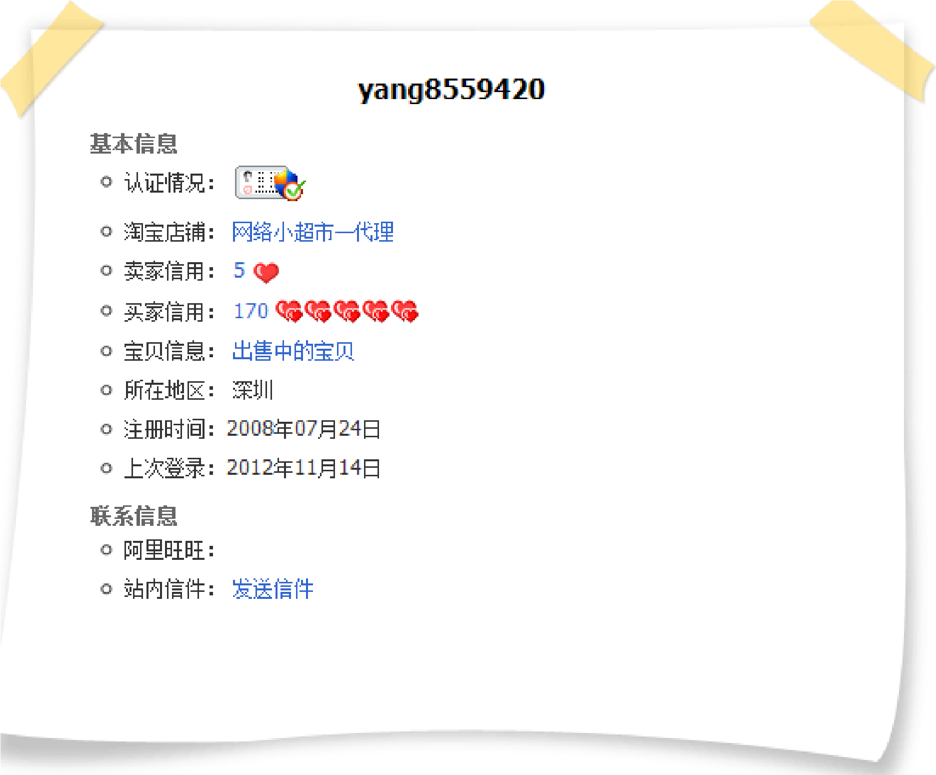
Based on ArcGIS Engine secondary development

A price of 500.00 yuan

Yang offered ArcGIS engine sourcecode for sale

Information about the seller:

http://shop65775432.taobao.com/view_page-74445421.htm



yang8559420

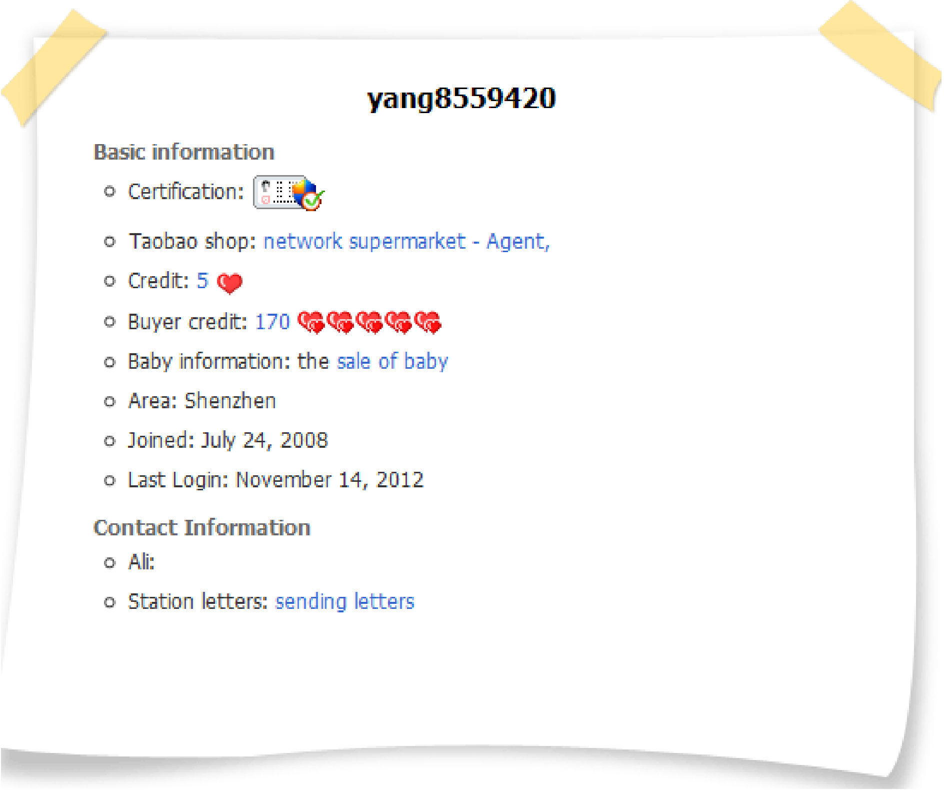
基本信息

- 认证情况：
- 淘宝店铺：网络小超市一代理
- 卖家信用：5 ❤
- 买家信用：170
- 宝贝信息：出售中的宝贝
- 所在地区：深圳
- 注册时间：2008年07月24日
- 上次登录：2012年11月14日

联系信息

- 阿里旺旺：
- 站内信件：发送信件

Yang8559420 trader profile (Chinese)



yang8559420

Basic information

- Certification:
- Taobao shop: network supermarket - Agent,
- Credit: 5
- Buyer credit: 170
- Baby information: the sale of baby
- Area: Shenzhen
- Joined: July 24, 2008
- Last Login: November 14, 2012

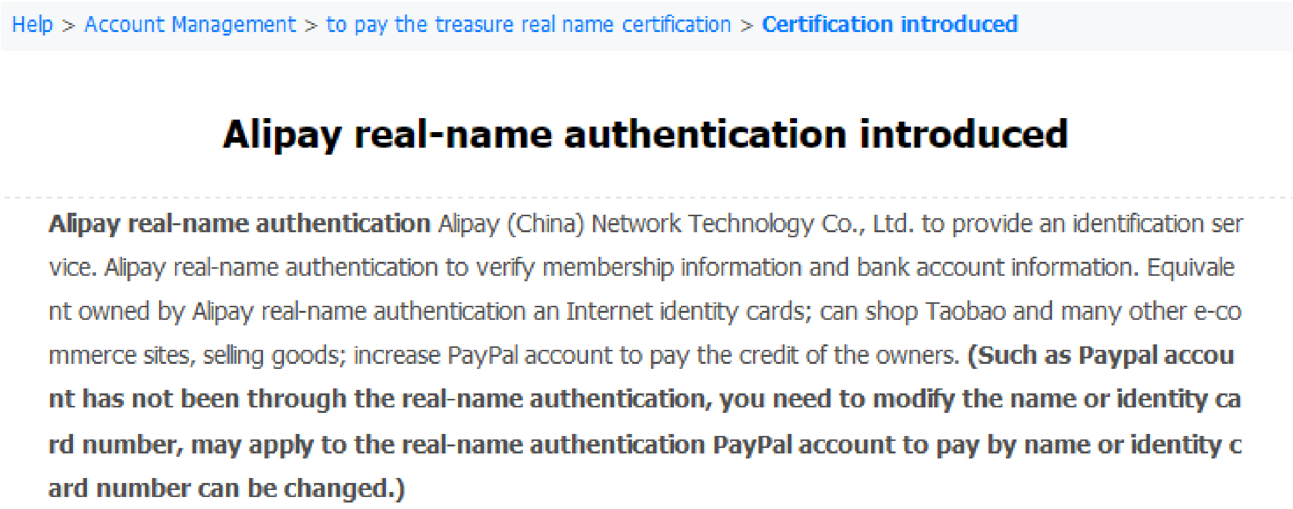
Contact Information

- Ali:
- Station letters: sending letters

Yang8559420 trader profile (English Google-translation)

Yang is certified at alipay.com (see field “Certification” above):

http://help.alipay.com/lab/help_detail.htm?help_id=211779



Help > Account Management > to pay the treasure real name certification > **Certification introduced**

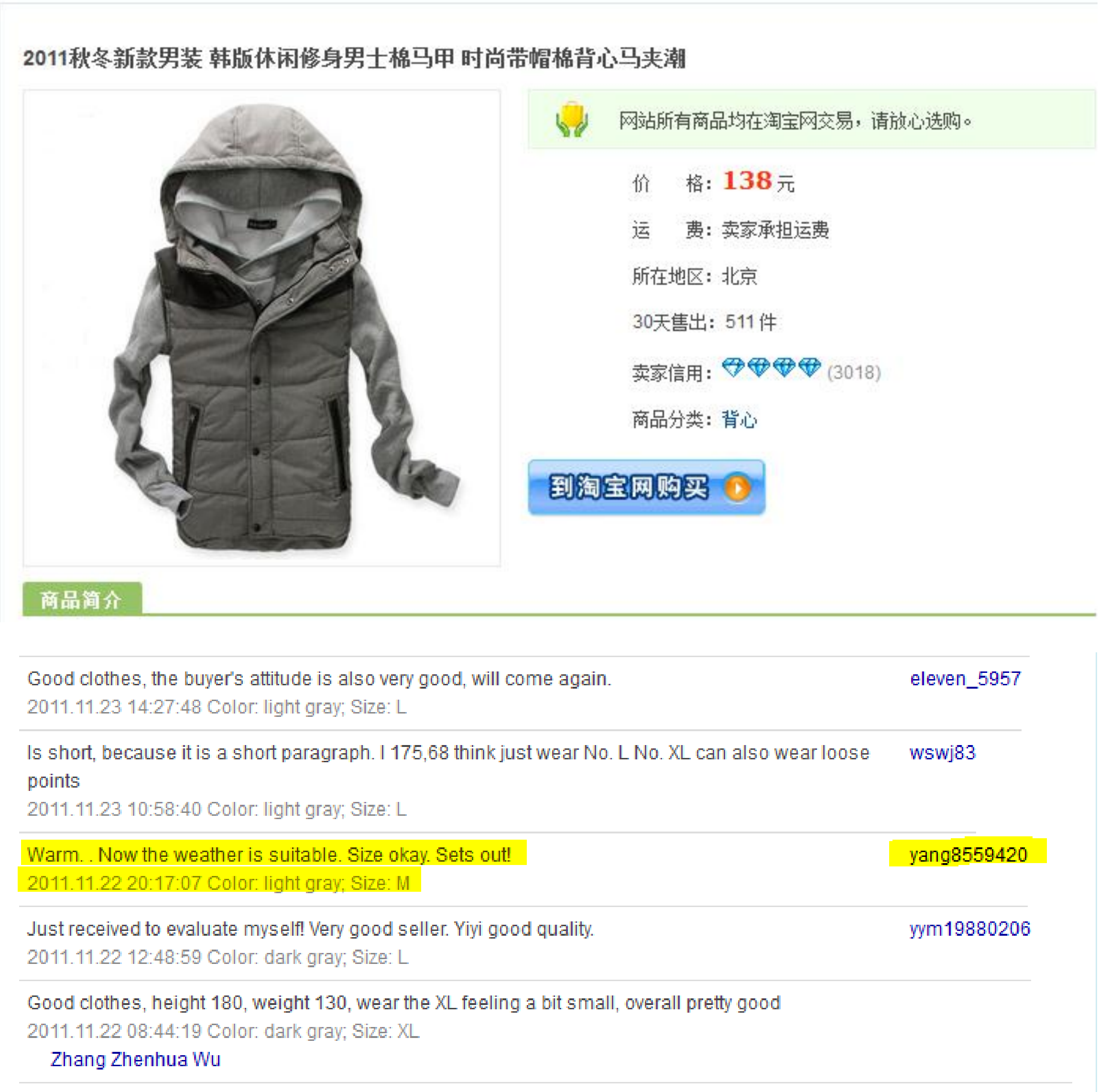
Alipay real-name authentication introduced

Alipay real-name authentication Alipay (China) Network Technology Co., Ltd. to provide an identification service. Alipay real-name authentication to verify membership information and bank account information. Equivalent owned by Alipay real-name authentication an Internet identity cards; can shop Taobao and many other e-commerce sites, selling goods; increase PayPal account to pay the credit of the owners. **(Such as Paypal account has not been through the real-name authentication, you need to modify the name or identity card number, may apply to the real-name authentication PayPal account to pay by name or identity card number can be changed.)**

Alipay certification

Yang left some feedback about a coat:

http://www.yifa8.com/4/766/770/763311.html



2011秋冬新款男装 韩版休闲修身男士棉马甲 时尚带帽棉背心马夹潮

网站所有商品均在淘宝网交易，请放心选购。

价　格：138元

运　费：卖家承担运费

所在地区：北京

30天售出：511件

卖家信用：◆◆◆◆ (3018)

商品分类：背心

Good clothes, the buyer's attitude is also very good, will come again. eleven_5957
2011.11.23 14:27:48 Color: light gray; Size: L

Is short, because it is a short paragraph. I 175,68 think just wear No. L No. XL can also wear loose points wswj83
2011.11.23 10:58:40 Color: light gray; Size: L

Warm. . Now the weather is suitable. Size okay. Sets out! yang8559420
2011.11.22 20:17:07 Color: light gray; Size: M

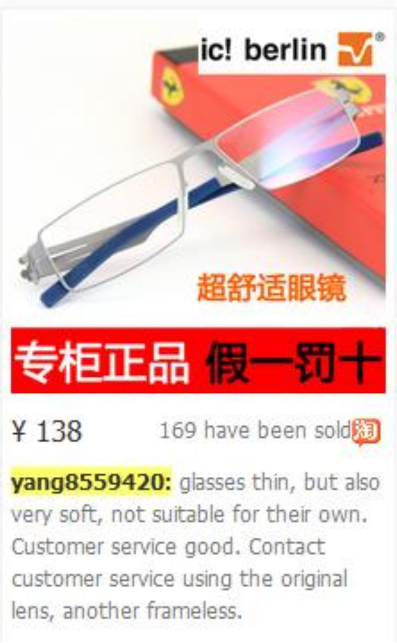
Just received to evaluate myself! Very good seller. Yiyi good quality. yym19880206
2011.11.22 12:48:59 Color: dark gray; Size: L

Good clothes, height 180, weight 130, wear the XL feeling a bit small, overall pretty good
2011.11.22 08:44:19 Color: dark gray; Size: XL
Zhang Zhenhua Wu

Yang comments on the internet (private life related)

Yang is selling glasses:

http://webcache.googleusercontent.com/search?q=cache:susBSuR_5zoJ:re.taobao.com/search%3Frefpid%3Dmm_16823808_2252954_8791633%26keyword%3D%2525D5%2525E6%2525CB%2525BF%252520%2525C1%2525AC%2525D2%2525C2%2525C8%2525B9%252520%2525C7%2525E5%2525B2%2525D6%26back%3Dlo1%25253D0%252526lo2%25253D0%252526nt%25253D1%26isinner%3D1%26yp4p_page%3D3%26posid%3D7+%22yang8559420%22&cd=14&hl=ru&ct=clnk&gl=ru



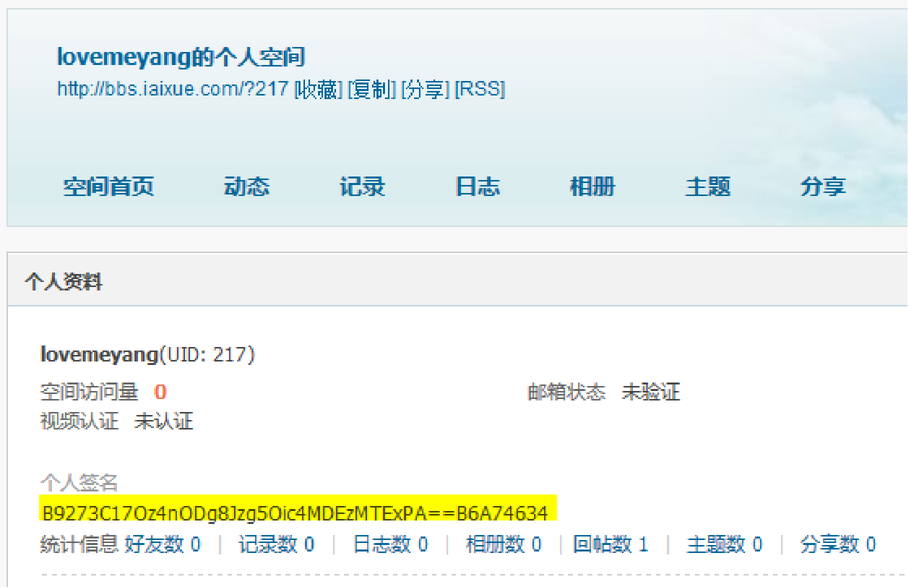
¥ 138 169 have been sold

yang8559420: glasses thin, but also very soft, not suitable for their own. Customer service good. Contact customer service using the original lens, another frameless.

Glasses for sale by Yang

http://bbs.iaixue.com/home.php?mod=space&uid=217&do=profile

User: *lovemeyang* (probably related to Yang). Signature is a message for a bot:



lovemeyang的个人空间

http://bbs.iaixue.com/?217 [收藏] [复制] [分享] [RSS]

lovemeyang(UID: 217)

空间访问量 0 邮箱状态 未验证

视频认证 未认证

个人签名

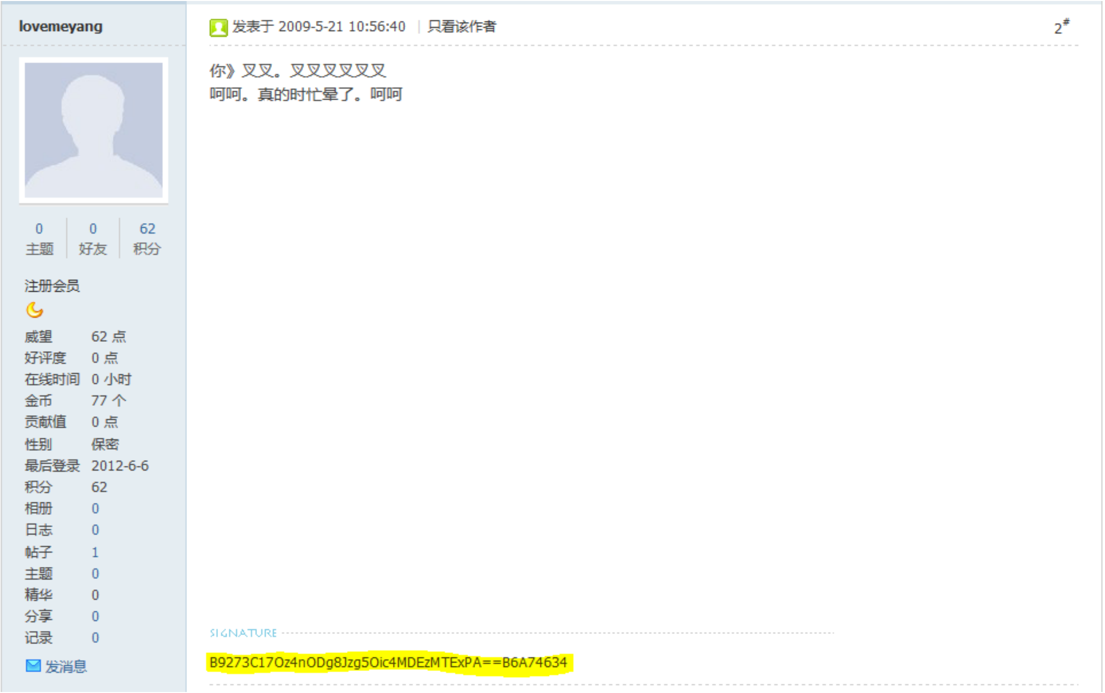
B9273C17Oz4nODg8Jzg5Oic4MDEzMTExPA==B6A74634

统计信息 好友数 0 | 记录数 0 | 日志数 0 | 相册数 0 | 回帖数 1 | 主题数 0 | 分享数 0

Another message for bot by lovemeyang

So, both Yang8559420 and Lovemeyang messages go with signature:

http://bbs.iaixue.com/forum.php?mod=viewthread&tid=261



lovemeyang

0 主题 | 0 好友 | 62 积分

注册会员

威望 62 点
好评度 0 点
在线时间 0 小时
金币 77 个
贡献值 0 点
性别 保密
最后登录 2012-6-6
积分 62
相册 0
日志 0
帖子 1
主题 0
精华 0
分享 0
记录 0

发表于 2009-5-21 10:56:40 | 只看该作者 2#

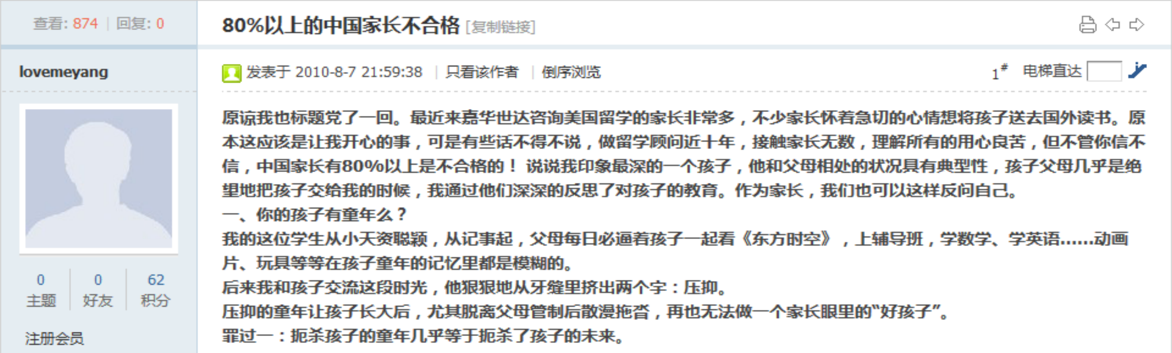
你》叉叉。叉叉叉叉叉叉
呵呵。真的时忙晕了。呵呵

SIGNATURE

B9273C17Oz4nODg8Jzg5Oic4MDEzMTExPA==B6A74634

Same signature used by Yang8559420 and Lovemeyang

http://bbs.iaixue.com/forum.php?mod=viewthread&tid=612



查看: 874 | 回复: 0

80%以上的中国家长不合格 [复制链接]

lovemeyang

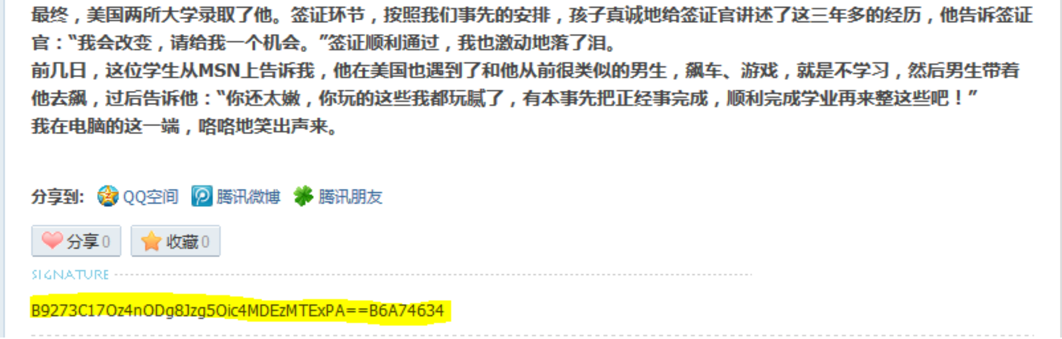
发表于 2010-8-7 21:59:38 | 只看该作者 | 倒序浏览 1# 电梯直达

0 主题 | 0 好友 | 62 积分

注册会员

原谅我也标题党了一回。最近来嘉华世达咨询美国留学的家长非常多，不少家长怀着急切的心情想将孩子送去国外读书。原本这应该是让我开心的事，可是有些话不得不说，做留学顾问近十年，接触家长无数，理解所有的用心良苦，但不管你信不信，中国家长有80%以上是不合格的！ 说说我印象最深的一个孩子，他和父母相处的状况具有典型性，孩子父母几乎是绝望地把孩子交给我的时候，我通过他们深深的反思了对孩子的教育。作为家长，我们也可以这样反问自己。
一、你的孩子有童年么？
我的这位学生从小天资聪颖，从记事起，父母每日必逼着孩子一起看《东方时空》，上辅导班，学数学、学英语……动画片、玩具等等在孩子童年的记忆里都是模糊的。
后来我和孩子交流这段时光，他狠狠地从牙缝里挤出两个字：压抑。
压抑的童年让孩子长大后，尤其脱离父母管制后散漫拖沓，再也无法做一个家长眼里的"好孩子"。
罪过一：扼杀孩子的童年几乎等于扼杀了孩子的未来。

…



最终，美国两所大学录取了他。签证环节，按照我们事先的安排，孩子真诚地给签证官讲述了这三年多的经历，他告诉签证官："我会改变，请给我一个机会。"签证顺利通过，我也激动地落了泪。
前几日，这位学生从MSN上告诉我，他在美国也遇到了和他从前很类似的男生，飙车、游戏，就是不学习，然后男生带着他去飙，过后告诉他："你还太嫩，你玩的这些我都玩腻了，有本事先把正经事完成，顺利完成学业再来整这些吧！"
我在电脑的这一端，咯咯地笑出声来。

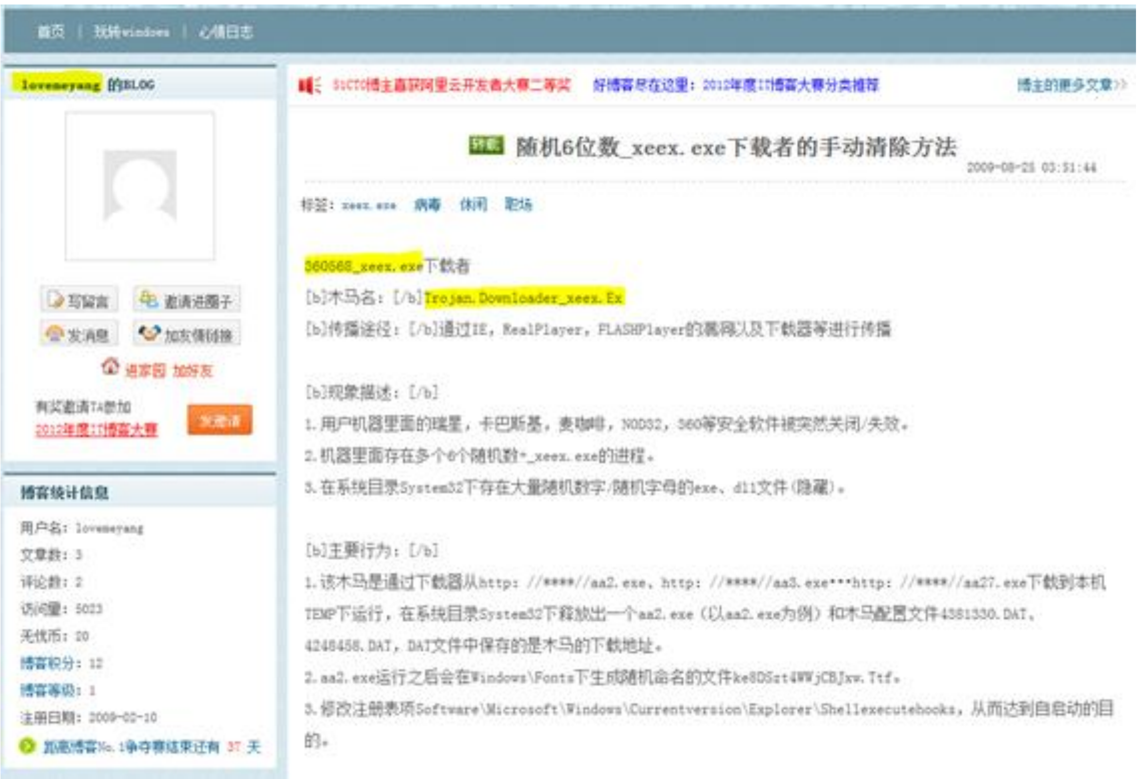
SIGNATURE

B9273C17Oz4nODg8Jzg5Oic4MDEzMTExPA==B6A74634

Signature by Lovemeyang

Search for "*lovemeyang*" returned too much data, making it difficult tofilter out those identifying possible attackers – false positives are highly-probable. However, it's worth mentioning that the following link refers to an account with the “*lovemeyang*” username and the user has earlier posted blogs relating to IT-security, so possibly the user is that Yang who is involved in the attack:

http://lovemeyang.blog.51cto.com/659880/195451



随机6位数_xeex.exe下载者的手动清除方法

2009-08-25 03:51:44

标签：xeex.exe 病毒 休闲 职场

360568_xeex.exe下载者

[b]木马名：[/b]Trojan.Downloader_xeex.Ex

[b]传播途径：[/b]通过IE，RealPlayer，FLASHPlayer的漏洞以及下载器等进行传播

[b]现象描述：[/b]

1.用户机器里面的瑞星，卡巴斯基，麦咖啡，NOD32，360等安全软件被突然关闭/失效。

2.机器里面存在多个6个随机数+_xeex.exe的进程。

3.在系统目录System32下存在大量随机数字/随机字母的exe、dll文件(隐藏)。

[b]主要行为：[/b]

1.该木马是通过下载器从http：//****//aa2.exe、http：//****//aa3.exe***http：//****//aa27.exe下载到本机TEMP下运行，在系统目录System32下释放出一个aa2.exe（以aa2.exe为例）和木马配置文件4381330.DAT，4248458.DAT，DAT文件中保存的是木马的下载地址。

2.aa2.exe运行之后会在Windows\Fonts下生成随机命名的文件ke8DSzt4WWjCBJxw.Ttf。

3.修改注册表项Software\Microsoft\Windows\Currentversion\Explorer\Shellexecutehooks，从而达到自启动的目的。

博客统计信息

用户名：lovemeyang

文章数：3

评论数：2

访问量：5023

无忧币：20

博客积分：12

博客等级：1

注册日期：2009-02-10

距离博客No.1争夺赛结束还有 37 天

Yang and relation to a malware

Conclusions

Our research revealed long-term oriented large scale cyber-espionage campaign of a criminal group with Chinese origins. These attacks are not new, many other security researchers have published details of various cybercriminal groups coming from China. However, the current hacking group has distinguishable features that make it stand out among others:

- Massive abuse of digital signatures; the attackers used digital signatures of one victim company to attack other companies and steal more digital certificates;
- Usage of kernel level 64-bit signed rootkit;
- Abusing great variety of public Internet resources to store control commands for the malware in an encrypted form;
- Sharing/selling stolen certificates to other groups that had different objectives (attacks against Uyghur and Tibetan activists);
- Stealing source code and other intellectual property of software developers in online gaming industry.

The Winnti hacking group is not the first and not the last. By making our research paper available to the public, we hope that it will not only spread the knowledge among security researchers but also will help system administrators and security officials in all type of organizations around the world to learn the tactics and tools of the perpetrators. We hope that our shared knowledge will help to better protect IT infrastructure. We also hope that our message will reach Chinese law enforcement agencies. If the current research is not enough to initiate criminal investigation, we hope that it will be enough at least to make some checks and probably prevent other malicious activity from reaching out foreign countries and business within China.

Appendix

Winnti MD5s:

Winnti 1.0

Win32 samples
006c4561499da562a4e337e2c146cf1a
024CC9872D9F413292D0F952920547CA
0613d67070679fb97ddefc5973c4d604
0630a443bd0102647ca1707cdf7f8c35
0751ca6f8b652cae6f2b650f0cf9036a
095a6a3b6eba996d2786b5ec919b1a7e
0af3761919bffa0019e7899333846b27
0f3c15de074f934499f5bbc095d5557f
11ed89f0ab17cf3973e2bf970879661a
128cb2a5de0d0422d69bab6d23ebb0aa
17c72e0cde2e4019a6b885f8188ac410
18813863417608b4ad14babebcafcb57
1a5da850993681e685893547d1aa2eaf
1ab7360a9438fb816f01ac00c17c9da4
1d688ca3148df378a15796f43242b77c
2128b6c7ec7848b73aeb6f211cef7615
296220a85742a8722b1335977dd98251
379251974ebcd5c397f92ca45bb9620d
38fb6993c3c94ea6df01235f44be4e77
3c722f0bea82e5bb8958f7fab012c911
3ecbc145dd593ec431145dd84e1e50cb
4038fb208d4b50e1f5f765811fdac174
41ff77ea7d4960c75d272a6a6fc31e7c
4402db68df6682bfe3e1e855a2474444
4722c665196fb6c7450980eafde6ac86
4e8f1c053dbe449c93f04e11d4afa352
4f213f9f187a65ce437157a3e7d253c0
50635147a579a8c8859a49c609f9d3d2
50678adefc49735a4f236e06e83c089d
5156bc9f1dd8ef1c1055933bb9c89c91
516fe9d2fe8b047fa8ba993692f44482
5171b030750f364a3459d5de22bc875d
5a93c03ddfe3edeb2573b72d12ebe0e5
5db7ba6e771cef48c623ae48fbb4740b
629c0a9d3d0f471005c87d06aed45113
64d225a757686db6263e5df919e9dfd6
6db0e662dad6407f666aa0ea4b995e7f
7460f35e3b24db9b92bc4cccb6c3f3ac
7529e41a101170eadb83bcb77bf29e65
814001293e4a50d12cf55563e0b95ffe
81b27822a6619a7c78eebbd6dc4b889d
9251ff253c38c437bad4926378981ad0
9a575f37ffa684d56d1f5ffebc24b8f3

a2c3fa86d43eca498c2b6ee8b5ecafb1
a62afe6d59ae1ac32e8afbb88345ba03
a91f69fc4b353d4228990464ca791705
ada3fb277229d6a12df364fd856f00c3
b01145e9d0c0f9d2822a250df95d888e
b28a68036b34e5d74672b289591aefa4
babd625bb2284d58a9c1884a80f07bdd
bb79348412e72e77a8254fc289244829
bc3ffe2761d210fa05dde9ced4ed4869
be8b2bf704a1165d5b8b4e26fff4180c
c050c1ca31e8509f7b12824824ba2ddd
c181065a366ea6f8c6791fd87fcb86d6
c248c15622cfb0985fb421c29771d6ae
c2ac3d2f0299633e2c588d2fa43d0d63
c2c2eb5f0762db8068bd4031bd6b59bc
c35180bd2138fd81469805d8eb3480bf
ca69ffc76e74e9d17f26f5f5b20a1db7
d202ca2b2e04b2b730c43e5a13927096
d8e289fba6a22cb853d737676ab1545d
e0df537f91f3bc3713a5ec5cf41f9e2d
e2e314cbdcf493bcd14cea9cdd887786
e464e0d0893add9d71bb951502ae738a
e58c7b9b2576c63ac60743a99310664b
eda0eb9e5c08729f12ddb64f6ec7ae2f
f06ec81a1f416812ffcc47fd5f709b50
f39fda34f2e332ddb1363f5e0e541c26
faa77eacaa7de27b0f04c3139066d73c
01f1204f54c645a13368e1ba54179779
099116c83c9b95ea71e75e1760fced28
2ad67673a4facf2b493ca5989839d8e3
2ec43703cc80323ae32fed751bedfff1
4a02ce3d6c6696ddda2a673298870e16
4b8fd1ee47f17164e61194f6b2dbfa40
508f0af84d83e093bf6910dbab45421f
5c865404f27f5e5b83b6fcfd94068118
8a0a00b1676c3b65b3c56dab7f8feb99
91ae694e565f4a2f52d5f792d8353fcd
95DF76F2ABDB9B133003D4DB637DC67B
be594ee2a7e4b11878de020cf724205f
ce3f94fea7f57ce5a9a5a26e51b617fb
d07f8aa768f7886400bb725c23fd2421
d9792b5f7bf497a3584d0c0d388f6b16
efdda5d0a14810ff86e60a70c5baa6b0
f975d016b83880c898b334714c1291b0
fc293476226d1471c8de65ab65af7b2f

Win64 samples

24c846e935d1efdd090469a69e01da65
604c8b4f2f82e016cff74ebc4a359e34

624db864fe644bc08c16cdbdb8f4bdfb
677c3236b3acac70f528de8b4cf62539
6e83c0e6739a2782ce385632f5e982c3
6e927175a6224add534a6072bc6a6170
7ea57ad96cee3db9baf5a36b43ba9abc
92fd35efabf8d774cf5bb4c2be8b733c
9642c7ee5819f5f8f3f8354da0845190
a00c66d502453524a7fe411ce7bbfea4
b062063cf2d5b7fcc4abd8390e4f0090
c9e55d71b7d8f05324c3ad041a943103
c9e9b8103077d9a9bb21e563f14ef738
ce3eecc1cc27e753b3eeae50074c3edd
d194316fc5a7f7b433d26ed9da09b249
de1ea8d6c20d8ecdd1c29219e30d4984
e5338b89c4721482df24f9aa5a3c6389
ec6d53e1a030e166acbc6f357362c195
66de2aaad67446aabbe5adeb873b4b24
8505e92a2c3812ec298acd6bb20437a2
9f5b4f39699fda67ffa65f98086f7451
B8F03B556AE4255BA8D828B6D9909B08
efb16a33a0c9da12a71ef44e7d688233

Drivers
5ce790274b7507740e9983d2efe69c17 679ba94211a4e027c2b56b959e62c8e3 6b4ab6ca6808e955a6fd11ae5ffea1f6 6f5a10edc2c7319b8d7abc0a606e5ce6 ca04aa367e6f090903018131245296ce e8e1f133ef1a303e2e901e59329af1dd 4591d01a291b700efbc5b263c67a266c

Winnti 1.1

Win32 samples
1014374a0b4972adec93a015df6e4558 582f84b21978cab7d190aef663a268ea 2d0950f69e206486c5272f2b0fc3aa22 a374be9091ed1791424fc236144e9d81 e867dba9d96acae55552777a8729a45a f809eea8170afacd2dfe2c45ba86861e

Drivers
07a18ad4d859c67f208ccb76a7e6a184 0996b71f1364acde317881810c5912f0 97f64270b59b0f6b83ec93efc41543fd

Droppers
509c562db69f8332b9fc3298236e8ffa
130a799edeb0753164cdb76ccf8fd64c
5654424ea88de69d5c6031f7009f0428

Winnti 1.2

Samples
0393eebedbde6e5ee868f81ac024b401
36711896cfeb67f599305b590f195aec
43da75e7f8e7e1893dce276bd5b2e680
535ede2d69a7e07a097ef6648b12e417
8acb42de94427141f7caffed74f9fc43
a0a96138b57ee24eed31b652ddf60d4e
d350ae5dc15bcc18fde382b84f4bb3d0
e252d9ec48bca3d261f5acdd33bfd1cb
f454ba447eef28f96dafe3398df82a7e
011815cb37f49a1d14d3db895a5e705f
115dc2627483aba7119ad4ceab1e042a
18677c3a2af1476aa8cbc73cfb74d8c1
1b0753f717d7a33defc389e399b20d57
29525be71ba4846739e553a0835ab460
2989b78ac3a752bf6792ac9ac606fdf0
2ffc739a927b62d4b7096e636951b77d
3047ed57acac30c2327e74070b3864b7
3d107d5bdf554c6ae8d05c886080a18d
4197499923ab6125e2ee5e950b21ec91
453021b8cc10f9077fa80d60d09c631d
4732d2056060c66f46caded82954836e
4d028c7a47c1b0d00e894ad351a61996
6e9b47f2ae1f9e7260b8793f35fbbd3a
8a1d1965b2d8501e692394bb801f58ca
a0629962c34ed9594b18493f459560a7
ada515709be09e495bc9c1206069e796
bfcd3417b513a6c3fed4b5466055d939

Droppers
60bd5a9ab78f6c614b824ddcb47dfd7c
8f54cf08ee45a8d5eb31d05dbab4b561
15d6249e0e7e03b3e00cc3917431cf64
4fbb502ba8c7e8d81ec98a5974b9001a
5618bc41af50c790c8e8680ba30030ed
7d51ea0230d4692eeedc2d5a4cd66d2d
961954bbc411d4eafd72efad94a6e160
c206992f7c6836ec6a227a6e29ae7609

Winnti 2.0

Samples
06d8b1468f09d10aa5c4b115544ccc6e
0cd07490fc02e2a602781bb939d0bc3d
2d0950f69e206486c5272f2b0fc3aa22
3358c54a22d186ec9de0f15bc4bb2698
35bdc5a2acf35bdf9fb9169e1a47d3e7
5778178a1b259c3127b678a49cd23e53
6dfcdc4c8edc77642f15592143f34569
9a83cd3f8e619c8b1b38b0b5ceeea357
afe4ec9a88f84fbf9c1eb0f3ff47a12b
B0BD6C215A7C20B23FD23D77FA26F3BA
bbbb9bb5c7a59b98f18b06344ac8980f
d23237edbdcc4118b538454b45c00021
d4a2060a5086c56f7ff65eaa65de81ff
dc22d742a15f8d6d8edf49d1c8cc8be9
e7e5c5c991e6d66fca16c988c891e10f
f4c9bc4f045b90c496df4b75398dfa5c

Drivers
04f3fbaaaf5026df29e0d7d317194043
07e40089cdf338e8d1423b3d97332a4d
0b105cd6ecdfe5724c7db52135aa47ef
7024ea8285cee098829ac8f2b1de4455

Compromised certificates

Company	Serial number
ESTsoft Corp	30 d3 fe 26 59 1d 8e ac 8c 30 66 7a c4 99 9b d7
Kog Co., Ltd.	66 e3 f0 b4 45 9f 15 ac 7f 2a 2b 44 99 0d d7 09
LivePlex Corp	1c aa 0d 0d ad f3 2a 24 04 a7 51 95 ae 47 82 0a
MGAME Corp	4e eb 08 05 55 f1 ab f7 09 bb a9 ca e3 2f 13 cd
Rosso Index KK	01 00 00 00 00 01 29 7d ba 69 dd
Sesisoft	61 3e 2f a1 4e 32 3c 69 ee 3e 72 0c 27 af e4 ce
Wemade	61 00 39 d6 34 9e e5 31 e4 ca a3 a6 5d 10 0c 7d
YNK Japan	67 24 34 0d db c7 25 2f 7f b7 14 b8 12 a5 c0 4d
Guangzhou YuanLuo	0b 72 79 06 8b eb 15 ff e8 06 0d 2c 56 15 3c 35
Fantasy Technology Corp	75 82 f3 34 85 aa 26 4d e0 3b 2b df 74 e0 bf 32
Neowiz	5c 2f 97 a3 1a bc 32 b0 8c ac 01 00 59 8f 32 f6

Winnti C&Cs

Winnti 1.0
newpic.dyndns.tv
update.ddns.net
nd.jcrsoft.com
cc.nexoncorp.us
98.126.36.202
kr.zzsoft.info
as.cjinternet.us
ca.zzsoft.info
sn.jcrsoft.com
lp.apanku.com
sshd.8866.org
ftpd.6600.org
tcpiah.googleclick.net
rss.6600.org
lp.zzsoft.info
lp.gasoft.us
eya.jcrsoft.com
ftpd.9966.org
kr.xxoo.co
wi.gcgame.info
tcp.nhntech.com
ka.jcrsoft.com
my.zzsoft.info
jp.jcrsoft.com
su.cjinternet.us
vn.gcgame.info
ap.nhntech.com
ru.gcgame.info
kr.jcrsoft.com
wm.ibm-support.net
fs.nhntech.com
docs.nhnclass.com
rh.jcrsoft.com
wm.nhntech.com
wm.myxxoo.com
ka.zzsoft.info
ad.jcrsoft.com
my.gasoft.us

Winnti, all, unsorted
gunz.gcgame.info
dell-support.org
t3.jcrsoft.com
kr.hja63.com
dbo.gcgame.info
2m.reegame.net
ns1.msftncsl.com
update.reegame.net
pop.hja63.com
imap.gasoft.us
dns.naverpulic.com
pda.zzsoft.info
pop.cjinternet.us
bar.gasoft.us
hja63.com
god.zzsoft.info
goqc.xxoo.co
apps.mynetav.net
ns3.nhnclass.com
tug.mynetav.net
vip-webmail.com
mail.7niu.com
game.joymax.in
tho.hja63.com
zb.mynetav.net
vtc.gasoft.us
tv3.mynetav.net
hk.hja63.com
ad.gasoft.us
ns5.msftncsl.com
ftp.zzsoft.info
sm.gcgame.info
eudb.reegame.net
tech.ibm-support.net
gm.gcgame.info
winlogon.net
iyy.conimes.com
ru.gcgame.info
oa.nexoncorp.us
cjinternet.us
wm.ibm-support.net
hp-supports.com
pass1.hangame.co.uk
mail.cjinternet.us
tt.xxoo.co
e.jcrsoft.com
gamenow.8800.org
googlefiles.net
ns4.msftncsl.com

gf.jcrsoft.com
sg.xxoo.co
ns3.nhnclub.com
wog.zzsoft.info
ssl.msftncsl.com
ns7.msftncsl.com
udp.nhntech.com
ad.jcrsoft.com
ns6.msftncsl.com
ibm-support.net
gh.zzsoft.info
kerberos.dnsalias.com
ns1.nhnclub.com
imap.zzsoft.info
gongyi.co
jcrsoft.com
uni.vip-webmail.com
smtp.jcrsoft.com
cc.nexoncorp.us
imm.conimes.com
mail.hja63.com
pass2.googletrait.com
club.cjinternet.us
mail.nexoncorp.us
as.cjinternet.us
service.dell-support.org
service.googlefiles.net
ftp.nexoncorp.us
e.gcgame.info
hansoft.sunsb.net
www.jcrsoft.com
ftpd.6600.org
sshd.8866.org
cpu.4pu.com
nx2.joymax.in
av.gcgame.info
dl-adobe.com
cj.jcrsoft.com
ro.myxxoo.com
rh.gcgame.info
cc.xxoo.co
swordwind.net
lp.xxoo.co
brqc.xxoo.co
ava.apanku.com
wi.gcgame.info
zm.gasoft.us
as.xxoo.co
gh.gasoft.us
baesystems.conimes.com
ns2.nhnclub.com

intercpu.com
e.hja63.com
pda.gasoft.us
wsafelogin.com
mail.nexongame.net
smtp.cjinternet.us
wm.nhntech.com
www.gcgame.info
ix.xxoo.co
support.dell-support.org
han.zzsoft.info
imap.hja63.com
nhntech.com
qc.xxoo.co
ip.xxoo.co
sl.myxxoo.com
mail.joymax.in
help.googleclick.net
www.nexoncorp.us
conimes.com
usa.xxoo.co
my.reegame.net
login.joymax.in
hsb.mynetav.net
docs.naverpulic.com
fax.nexoncorp.us
mail.jcrsoft.com
guys.mynetav.net
google.x3322.org
jc.nhntech.com
roqc.xxoo.co
ws.gcgame.info
xss.gongyi.co
new.java-ssl.com
ava.zzsoft.info
eya.jcrsoft.com
gn.xxoo.co
crl.nhntech.com
tah.xxoo.co
dns.nhnclass.com
zzsoft.info
nx.xxoo.co
ns2.naverpulic.com
pop.zzsoft.info
on.xxoo.co
pwd.nhntech.com
ftp.gcgame.info
nx2.hangame.co.uk
he.xxoo.co
hk.zzsoft.info
nhnclass.com

nexoncorp.us
w.gasoft.us
kr-mail.com
ns1.nhnclass.com
smtp.nexoncorp.us
xv.apanku.com
imap.nexoncorp.us
stmp.msftncsl.com
nx3.hangame.co.uk
msftncsl.com
soft.hja63.com
bcc.hja63.com
wm.myxxoo.com
ns3.msftncsl.com
us.msftncsl.com
dns--google.com
t3.myxxoo.com
au.msftncsl.com
support.nexononline.com
sg.java-ssl.com
l53.xxoo.co
udp.myxxoo.com
q.gasoft.us
nx2.interdriver.net
a.gcgame.info
mg.zzsoft.info
jp.xxoo.co
ros.zzsoft.info
x64.reegame.net
versiontt.no-ip.org
imap.cjinternet.us
rf.gcgame.info
ca.zzsoft.info
pda.hja63.com
tw.java-ssl.com
java-ssl.com
sn.jcrsoft.com
service.interdriver.net
db.nexongame.net
id.java-ssl.com
perl.mynetav.net
osk.jcrsoft.com
mini.googletrait.com
mail.gcgame.info
nc.feelids.com
tcpiah.googleclick.net
googleclick.net
pop.hangame.co.uk
www.gasoft.us
nxeu.jcrsoft.com
eya.zzsoft.info

sellsads.sells-it.net
wapqq.3322.org
kr.reegame.net
nt.nexoncorp.us
tcp.nhntech.com
www.hja63.com
aion.reegame.net
su.cjinternet.us
get.java-ssl.com
eudb.nexongame.net
nsqc.xxoo.co
mail.gasoft.us
kr.jcrsoft.com
ads01.mynetav.net
gm.gasoft.us
a1.reegame.net
smtp.gcgame.info
pda.jcrsoft.com
kor.xxoo.co
ns9.msftncsl.com
nx.jcrsoft.com
nexon.hangame.co.uk
smtp.gasoft.us
ns2.java-ssl.com
alta.apanku.com
nexon.joymax.in
my.gasoft.us
dns2.msftncsl.com
ckts.mynetav.net
pass1.googletrait.com
dns.nhnclub.com
kr.zzsoft.info
mir.reegame.net
jrun.hja63.com
wm.googleclick.net
bot.dongevil.info
mail.zzsoft.info
nexononline.com
tv.mynetav.net
e.gasoft.us
xy.hja63.com
www.apanku.com
usa.nexongame.net
ftp.gasoft.us
ogp.reegame.net
kog.jcrsoft.com
www.joymax.in
br.xxoo.co
ftp.cjinternet.us
qc.zzsoft.info
pay.gcgame.info

hangame.co.uk
test.reegame.net
gs.xxoo.co
xx.hja63.com
ap.myxxoo.com
cg.apanku.com
ns1.naverpulic.com
ree.reegame.net
jp.jcrsoft.com
interdriver.net
ns1.java-ssl.com
www.googletrait.com
www.zzsoft.info
qs.nexongame.net
nx3.joymax.in
a1.nexongame.net
wi.zzsoft.info
mx.hja63.com
ga.nhntech.com
nx.cjinternet.us
ftp.jcrsoft.com
fm.hja63.com
lftv.mynetav.net
e.zzsoft.info
udp.ibm-support.net
nx3.intercpu.com
wh.jcrsoft.com
zz.xxoo.co
shoes.sellClassics.com
ar.apanku.com
ka.zzsoft.info
jjevil.com
nexongame.net
est.gcgame.info
imc.zzsoft.info
newpic.dyndns.tv
mini.reegame.net
update.ddns.net
js.nexoncorp.us
nd.jcrsoft.com
ed.xxoo.co
also.msftncsl.com
support.interdriver.net
ru.cjinternet.us
smtp.zzsoft.info
pda.gcgame.info
th.xxoo.co
nhnclub.com
www.cjinternet.us
ssh.joymax.in
tvads01.dyndns.tv

pp.ibm-support.net
blog.mynetav.net
ijj.conimes.com
tank.hja63.com
lp.gasoft.us
nx3.googlefiles.net
pass1.nexongame.net
gcqc.xxoo.co
br.reegame.net
ftpd.9966.org
kr.xxoo.co
offices.dyndns-office.com
hansoft.does-it.net
gasoft.us
docs.nhnclub.com
sf.cjinternet.us
pass2.nexongame.net
updata-microsoft.com
ka.jcrsoft.com
us.xxoo.co
myav.mynetav.net
w53.myxxoo.com
isatap.dyndns.org
tt.conimes.com
vn.gcgame.info
ap.nhntech.com
bot.jgame.in
l.xxoo.co
ftp.hja63.com
mail.msftncsl.com
dns01.dyndns-work.com
service.hp-supports.com
ns2.nhnclass.com
fax.cjinternet.us
nx2.intercpu.com
windows.doomdns.com
btg.mynetav.net
xxoo.co
mynetav.net
mini.msftncsl.com
pass2.hangame.co.uk
webadmin.dnsdojo.net
imap.gcgame.info
joymax.in
udp.jjevil.com
www.reegame.net
myxxoo.com
iss.conimes.com
ads01.dyndns-web.com
www.mynetav.net
dns.msftncsl.com

pop.jcrsoft.com
ball.reegame.net
lyto.zzsoft.info
rw.nhntech.com
els.jcrsoft.com
a1.googletrait.com
googletrait.com
w80.xxoo.co
scvhosts.com
nexon.nexongame.net
pic.4pu.com
q.gcgame.info
dbo.jcrsoft.com
ns2.msftncsl.com
ynk.xxoo.co
tw.hja63.com
pass1.reegame.net
my.zzsoft.info
www2.mynetav.net
www.nexongame.net
id.naverpulic.com
roap.myxxoo.com
openhost.webhop.net
mir2.nexongame.net
imap.jcrsoft.com
pop.gasoft.us
bar.zzsoft.info
game.nexongame.net
fs.nhntech.com
osk.zzsoft.info
docs.nhnclass.com
t3.nhntech.com
ahn.gasoft.us
officess.dyndns-office.com
new.nexoncorp.us
dbo.zzsoft.info
w.zzsoft.info
lp.gcgame.info
ro.hja63.com
gcgame.info
xl.apanku.com
web-games.us
sl.xxoo.co
login.hangame.co.uk
ro.xxoo.co
dbo.gasoft.us
moon.reegame.net
egi.mynetav.net
vn.jcrsoft.com
ftp.mynetav.net
us.nhntech.com

masternow.webhop.net
file.googlefiles.net
holleword.3322.org
est.zzsoft.info
apanku.com
help.ibm-support.net
tw.reegame.net
est.gasoft.us
mg.jcrsoft.com
lp.apanku.com
smtp.hja63.com
xnews.myPicture.info
lp.zzsoft.info
nx3.interdriver.net
rss.6600.org
fn.hja63.com
usp.xxoo.co
ads01.dyndns-pics.com
oky.mynetav.net
pop.nexoncorp.us
naverpulic.com
pop.gcgame.info
ap.googleclick.net
haj.mynetav.net
ac.xxoo.co
mini.nexongame.net
udp.googleclick.net
nd.xxoo.co
new.myxxoo.com
rh.jcrsoft.com
wm.xxoo.co
dns.java-ssl.com
wyqc.xxoo.co
q.zzsoft.info
pass1.joymax.in
item.ItemDB.com
reegame.net
mailes.dyndns-mail.com
nd.gasoft.us
a.zzsoft.info
w53.xxoo.co