Flancrest Enterprises Inc.

Major Incident Report

December 12th 2019

Contents

Executive Summary

On November 15th 2019, Flancrest Enterprises administrators received notifications from an employee stating they had observed their local file system become quickly encrypted by an unknown program. This employee downloaded the toolkit designed to aid in the development of Excel macros from a remote web site hosted by a 3rd party outside the scope of Flancrest Enterprises vendor management efforts. Initially the employee was not aware that the software downloaded was malicious, and had no malicious intentions in downloading this program, however once they observed their file system becoming encrypted at an alarming rate, they immediately notified the IT administration team for assistance. IT administrator Todd Sanders began investigating and quickly alerted the IT Security team of a possible incident in progress. IT Security lead Rod Sanders began investigating anti-virus system logs for other signs of infection. The initial infection did not trigger any anti-virus alerts, however subsequent infections within the Flancrest infrastructure were quickly noticed by the company's Spylance monitoring environment, however proactive efforts to stem the infection and encryption routines automatically of the malicious software were unsuccessful due to the nature of the malware and the delay in signature production for the Spylance software suite.

Once signatures were developed and loaded into the tool, the malware was quickly identified across the organization, and subsequently quarantined to prevent further infection. After thorough investigation, Rod uncovered 25 workstations and 4 servers which had encrypted files and indicators of malware infection. User machines were segmented to an isolated network, and re-imaged from a known good backup, which would have eliminated the malware and any attached methods of persistence.

The servers affected were FLAN-SQL-BAK01, FLAN-FS01, FLAN-PS01, and FLAN-SQL01. These servers were also isolated to separate networks, and the file server as well as the print server were restored from known good backups and no indicators of compromise were observed. The remaining database server (FLAN-SQL01) was not imaged from last known backup, as the previous four backups were corrupted by the malware affecting the database backup server (FLAN-SQL-BAK01). Additionally backups previous to these did not contain very important data required for the new E-commerce platform rollout that was set to complete in early Q1.

Incident managers determined the best course of action was to retain a security company in the local area by the name of Gravey and Jobriath Security LLC (hereafter referred to as "G&J Security"), who assisted with a forensic investigation of the remaining affected servers for potential indicators of compromise and analysis of potential persistence methods left behind by the ransomware infection. A suspicious binary file was found by analysts, and a sample was extracted for further investigation by G&J Security's malware analysis team.

At the time of this report, the infection is resolved, files are restored, systems patched, and security tools updated to minimize the impact of a similar event in the future.

Introduction

On November 15th, 2019, a user in the Finance and Accounting department browsed to a website detailing how to create Excel macros in spreadsheets for faster accounting and analysis efforts. This website included a free toolset available for download, which the user opted to download and install on their corporate system. While this is not against the acceptable use policy, the tool they downloaded was from an unknown source and was not subject to any analysis prior to download and installation efforts. As a result, the program was a Trojan horse style binary that contained a payload of ransomware code designed to encrypt the user's local files, and spread through attached file shares on the local system.

The user notified IT administrators quickly, who in turn notified IT security of an event affecting the user's workstation and underlying data. After some investigation by the IT security team, it was determined the infection had spread to the entire Finance and Accounting user machine network, as well as some production servers that were attached to the network for business purposes. It was at this point an incident was declared, and the incident response process engaged.

After these machines were contained to an isolated environment, further investigation revealed no indicators of compromise through the remaining elements of the infrastructure. Signatures were created and provided by the Spylance AV team to assist with detection and quarantine efforts throughout the environment.

User workstations were quickly re-imaged and no indicators of compromise were detected on the machines after imaging efforts were completed. The affected servers were also re-imaged, with the exception of the e-commerce database server FLAN-SQL01 and its attached backup server FLAN-SQL-BAK01. Once it was confirmed these servers could not be re-imaged, a security organization was retained to assist in the forensic investigation and analysis of suspicious files left behind by the infection.

The contents of this report below detail the scope of the infection, initial causes, and contents of the analyzed binary that was flagged as suspicious by IT security administration and G&J Security analysts.

Scope

The following departments and system owners were affected by this security breach:

- Finance and Accounting
- IT Administration

The following corporate network elements were affected by this incident:

- VLAN01 – Users workstations
- VLAN14 – Finance to database connection

The servers listed below were confirmed infected during this incident:

- FLAN-FS01: File server used by accounting and finance team
- FLAN-PS01: Print server used by accounting and finance team
- FLAN-SQL01: Database server containing backend data for the upcoming deployment of the new e-commerce platform due for release in early Q1
- FLAN-SQL-BAK01: SQL01 database backup server specifically for maintaining backups for the e-commerce system.

Business Impact

The initial impact to the business was minimal as the files encrypted were backed up recently, and user machines were quickly imaged back to a known good state which allowed the employees to continue working with minimal downtime. It is estimated that no more than 4 hours of user work time was lost due to this infection, as the imaging process was conducted in assembly line fashion with automated tools and effective workflows.

The server infections resulted in high impact, as those attached machines contained large amounts of data and additional time was required to eliminate the infection and restore the affected files. Identification and server restore was completed in less than 5 hours, however data restoration required approximately 12 hours to fully restore once the servers were deemed safe. Total time spent recovering the server environment is 20 hours, when accounting for investigation and initial analysis efforts.

Analysis of the suspicious file left behind on the SQL server was an additional 20 hours including report creation, and review time.

Total labor time associated with this breach and subsequent activities is between 40 and 50 hours total.

Due to the nature of ransomware, it is not believed that the data has been copied and stolen by the attacker, as the payload is designed to elicit payment, rather than compromising files. Ongoing investigations on the dark web, as well as other open source intelligence mechanisms is recommended as a good security practice.

Suspicious File Analysis Results

This sample is an Agobot variant ransomware. It was executed by running the Trojan file excel Macro free toolset. This executable is a botnet that requires little or no programming knowledge to use. It is a multi-thread program and written in Visual C++.

When this malware is running, it establishes a connection to an IRC server(irc.foxlink.net). It is a standalone file that copies itself to the Windows System folder and creates a Registry key to start that file during every Windows session. After connecting to an IRC server, it creates a bot in a specific channel on the IRC server. The compromised host acts as a backdoor server interface. And join the IRC channel as a client. A bot master can send commands to a bot using the IRC interface.

This sample utilizes the RSA encryption algorithm. RSA is an asymmetric cryptographic algorithm to encrypt files. Asymmetric means that there are two different keys which are a public key and private key. When activated, the malware encrypted file system on localhost and mounted network drives using RSA public-key cryptography, with the private key sent back to the bot master and stored on the malware's control servers.

As an Agobot, it has various features. It has functionalities of scanning the local network and port. Even though this malware disables DDoS function, it has an HTTP command to do DDoS attack. Also, It has an SMTP command for spam. There is a function to execute programs and commands as well.

For persistence, It implies file-based persistence by copying itself to a hidden file. Modifying registry run keys is another typical persistence technique for malware. This technique allows the actual payload to execute when a user logs in.

It is hard for anti-virus to detect this sample because it utilized upx to pack itself. What is more, it has the ability to detect and terminate running anti-virus processes. By changing firewall settings of certain registry keys, it could launch an auto-scan local network and create an internet connection for receiving a command from bot master.

It behaves totally different running in user mode then running with administrator privileges. It requires administrator privileges to create and copy itself or connect back to the bot master. It obtains administrator privileges through issuing a brute force attack user accounts. This attack is carried out using a simple wordlist.

Recommendations and Conclusion

Infections of this nature are common place in organizations of any size, and can happen with the same methods as Flancrest experienced in this breach. Early identification and quick work by incident responders ensured the infection was kept to a minimum, and recovery efforts while long were mostly automated and subjected to constant supervision by administrators. At the time of reporting the conclusion is that the infection is eradicated from the environment, and all systems are restored to their previous functions.

We recommend 6 simple methods to keep our system safe.

1. End-user Education
As with all malware, the biggest problem is end-user education, as long as people click on executables from untrustworthy sources we will continue to see these threats. Therefore, we recommend the HR department arrange information security training for employees.

2. Use a strong password and change it regularly.
Some malware will use a brute-force attack using word lists to access the account. To prevent this attack and make harder to break the password, we recommend creating a standard for password with numbers, capital letters and special characters, so that employees should follow the standard. Also, we recommend system administrator change the password regularly for a critical system.

3. Keep the Antivirus up to date.
While Av engineers have the time and skills to break this obfuscation and get into the original executables, keeping your AV product up to date with the latest versions of each of these unpackers would require regular updates similar to how virus signatures are currently distributed.

4. Back up the computer
It's very important to get a habit to back up the computer whenever you get important data in order to recover even malware encrypts every file on the computer.

5. Keep Checking the unspecific traffic from the network.
Agobots are different from your average Trojan, in that a well-educated Network Administrator will be able to spot infected machines on his network with reasonable ease. With the regular inspection of network flows you will quickly notice any irregular activity like heavy IRC traffic to non-standard ports, or NetBios network scans looking for weak passwords.

6. Close all uncommonly used ports.
If there are lots of opened ports which are not commonly used, it can be a target from an attacker and they
will use that port to communicate.

It's hard to prevent unexpected attacks, however, if we follow the methods above, we might reduce the risk of infection.

Appendix 1: Malware Analysis Process Documentation

1. Basic information of malware.

Date	Dec. 12. 2019
Time	13:00
Analyst Name(s)	Jun Wang, Modae Kang
Binary Hash Value (SHA-256)	e652d7c27f43ac16a2ec6ee5491166674882d458e49dd667f97b97992e912d1c
Binary Hash Value (SHA-1)	cc591268d617fff39d92b3a7ab0d3e052a251f66
Binary Hash Value (MD5)	c8ae98259a3d11aaaad08bffd40d6228

[Static Analysis]

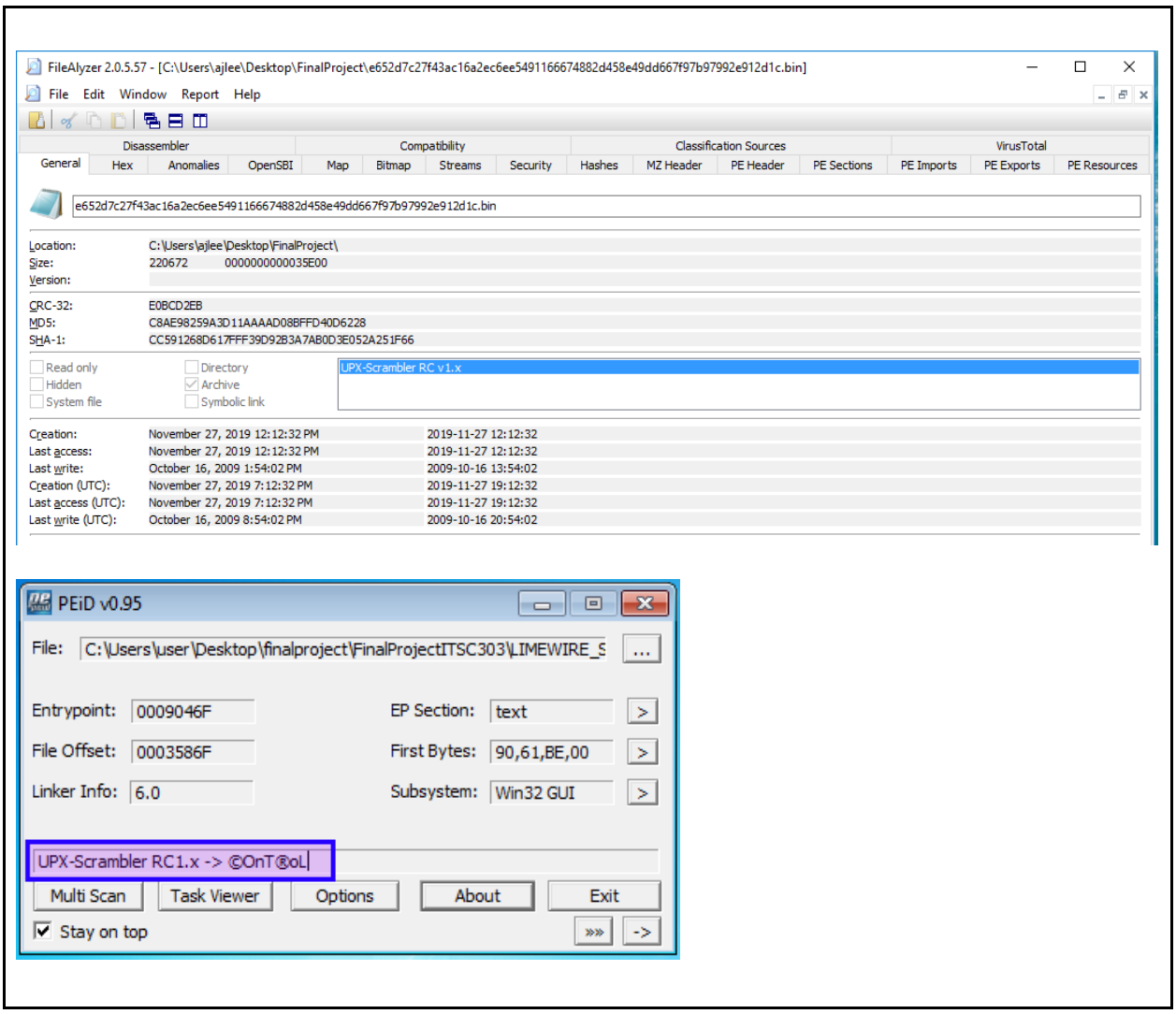
2. Strings Analysis

Description of String Usage: There are only a few strings which is readable and meaningful.

Strings Found	Offset Address in Hex	Description of String Usage
DnsQuery_A	0x00035C0D	provides application developers with a DNS query resolution interface.
WNetAddConnection2W	0x00035C16	makes a connection to a network resource and can redirect a local device to the network resource.
??1out_of_range@std@@UAE@XZ	0x00035C2A	The error handling method to operate some specific operations, when enumerating through processes.
EnumProcesses	0x00035C5A	Retrieves the process identifier for each process object in the system.
ShellExecuteA	0x00035C6A	Performs an operation on a specified file.
wsprintfA	0x00035C7A	Writes formatted data to the specified buffer

3. Packed or Obfuscated Identification

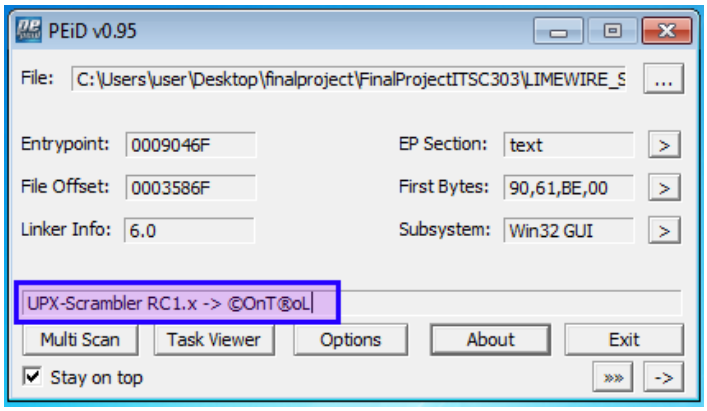
Based on the result from FileAlyzer and PEiD, it's packed by UPX-Scrambler. If it's packed, it's hard to do static analysis. In this case, there are two cases. First is move on dynamic analysis and second is try to unpack it.



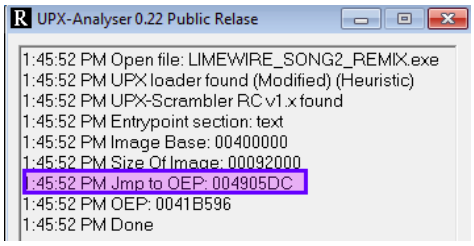
4. Unpacking

before do dynamic analysis, try to do more static analysis after unpacking it.

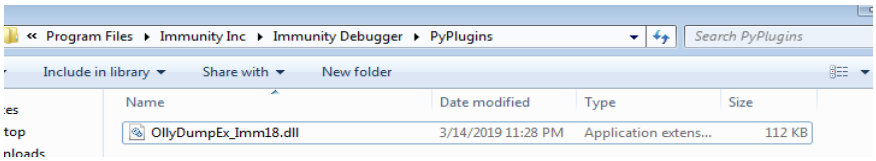
1) Check the file is packed or not.



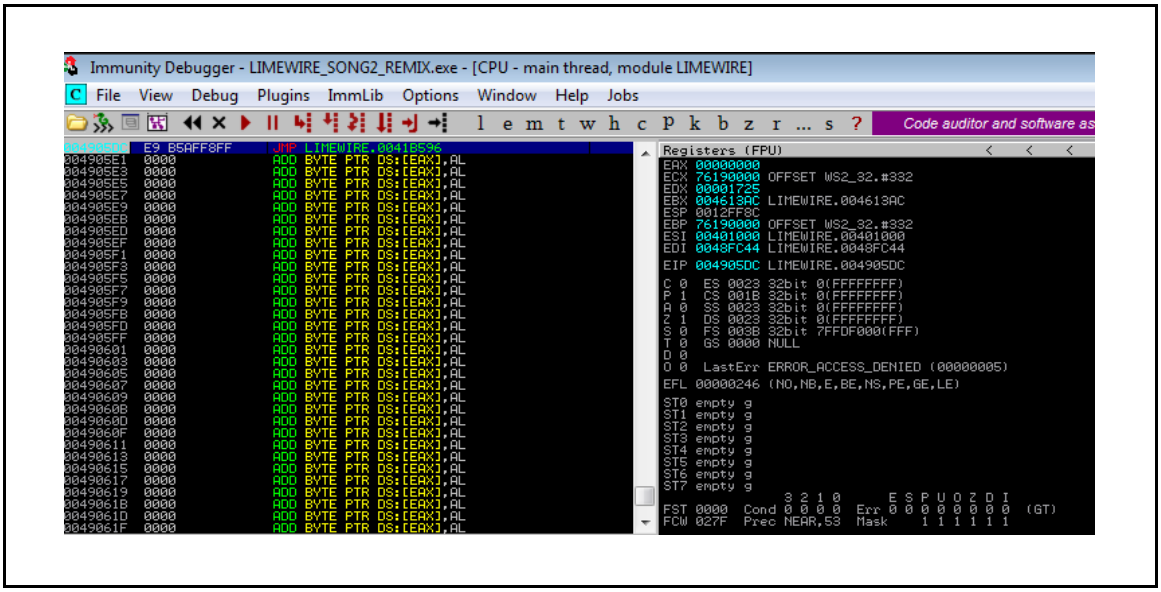
2) Find OEP using UPX-Analyser.



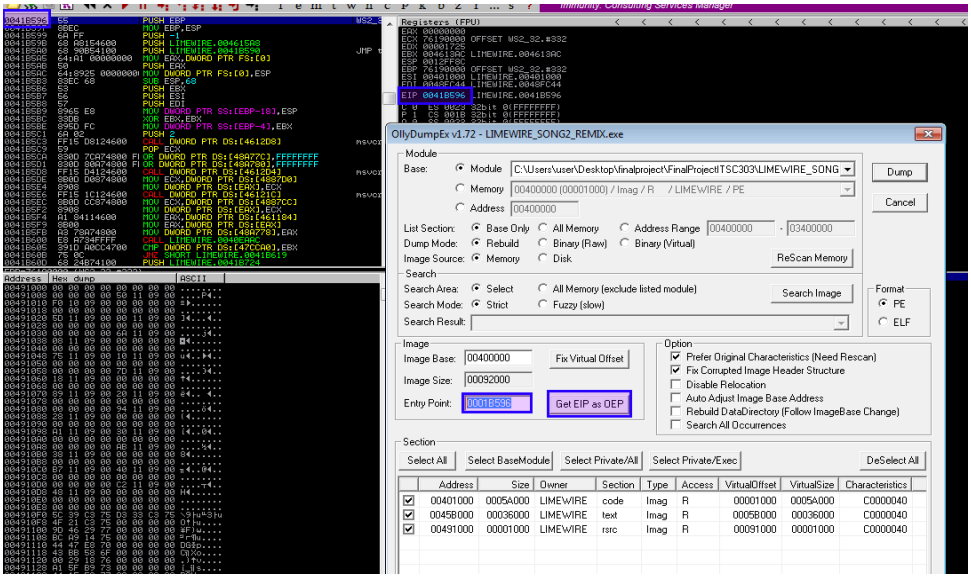
3) Need a plugin for dumping the code. Download OllyDumpEx and extract it. Put the OllyDumpEx_Imm18.dll to C:\Program Files\Immunity Inc\Immunity Debugger\PyPlugins



4) Open the sample in Immunity Debugger. Ctrl+G find the OEP 004905DC. Set a breakpoint(F2) and run(F9) until reach this break point.

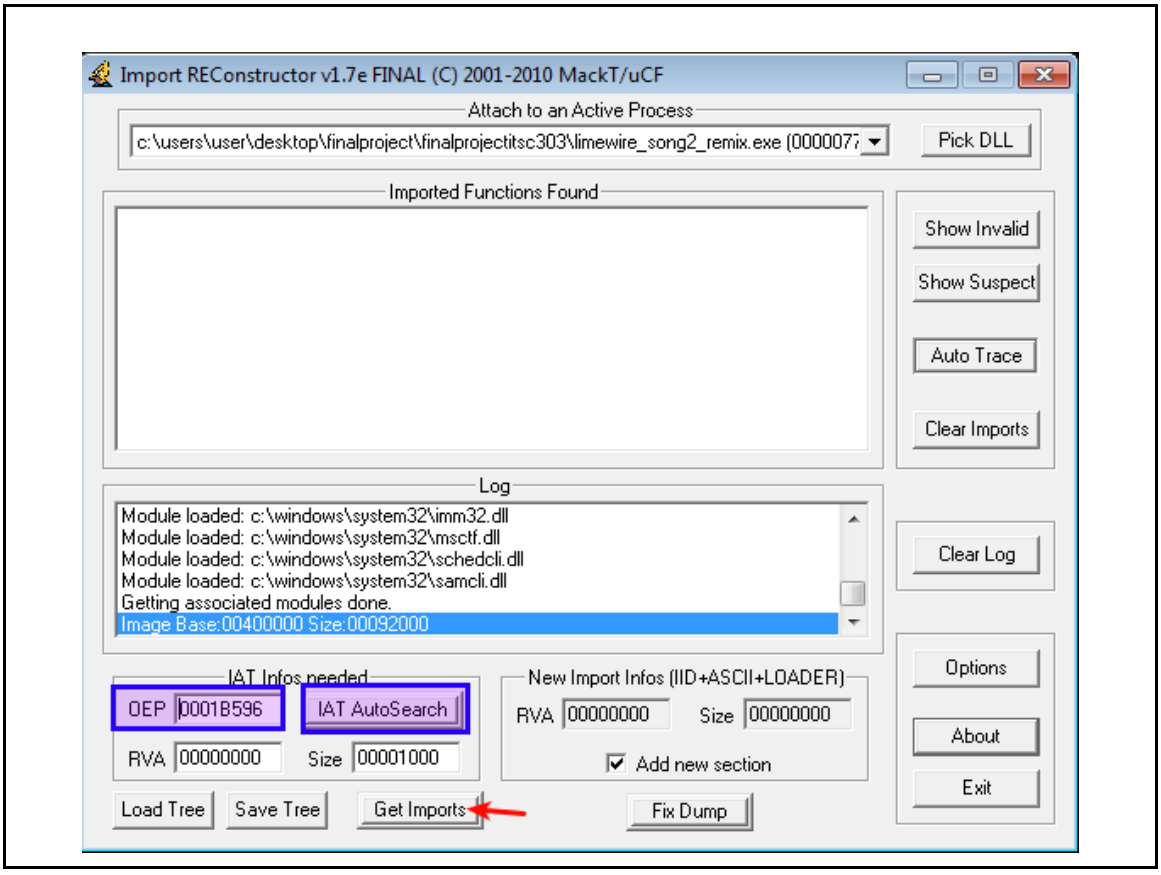


5) Step in to this jump. It will be the real OEP.
Right click on the first instruction, and click on OllyDump.
Click on Get EIP as OEP. It will replace the Entry Point with EIP (real OEP).
Click on Dump. It will generate a dmp file for the decompressed code.

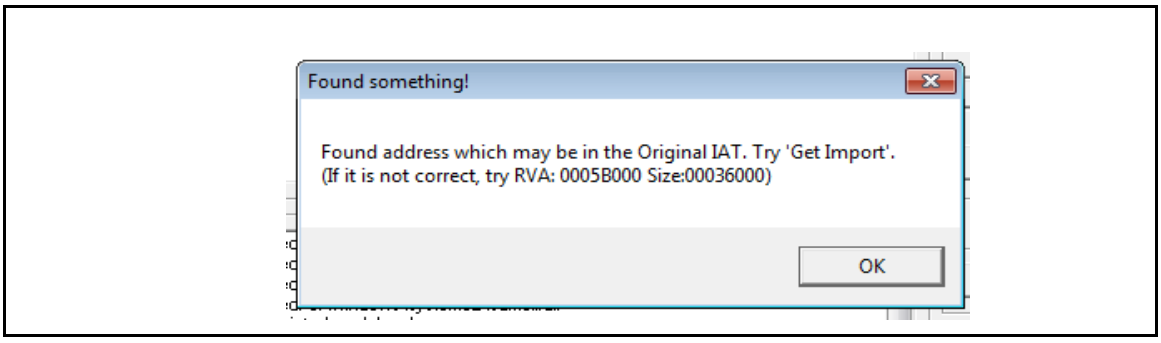


6) Open Import REConstructor, Attach to active process -> the malware sample running on immunitydbg.

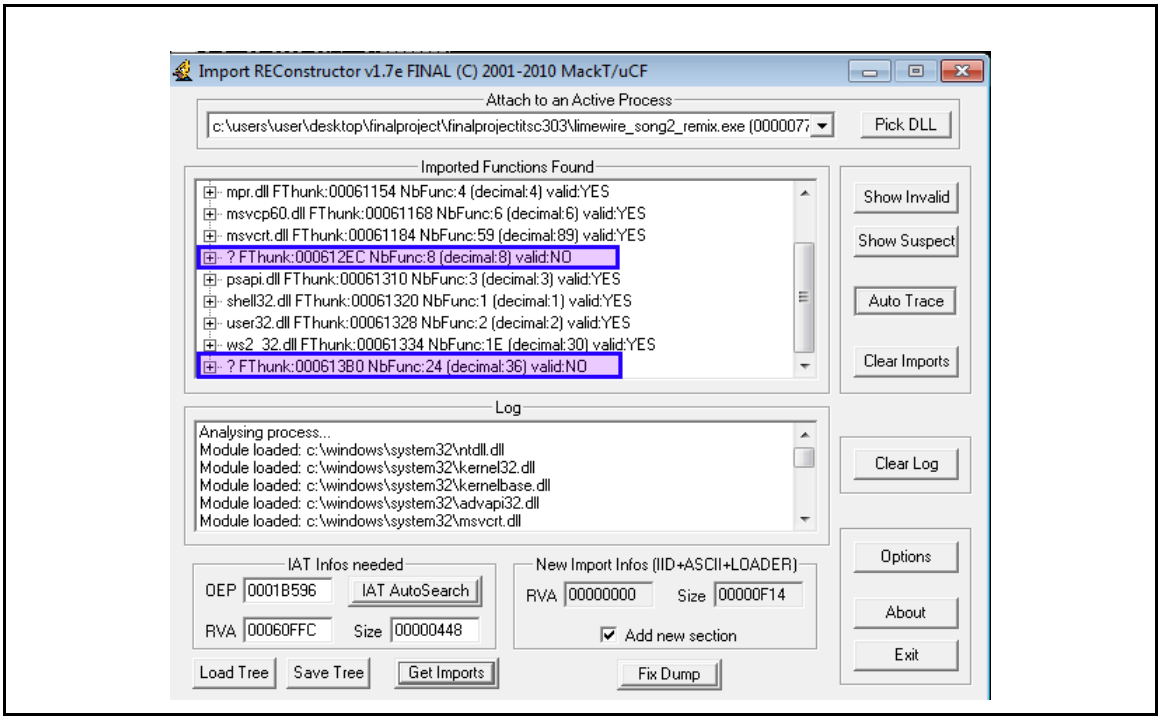
7) Change the OEP to the one we find in Immunity Debugger , 0001B596.



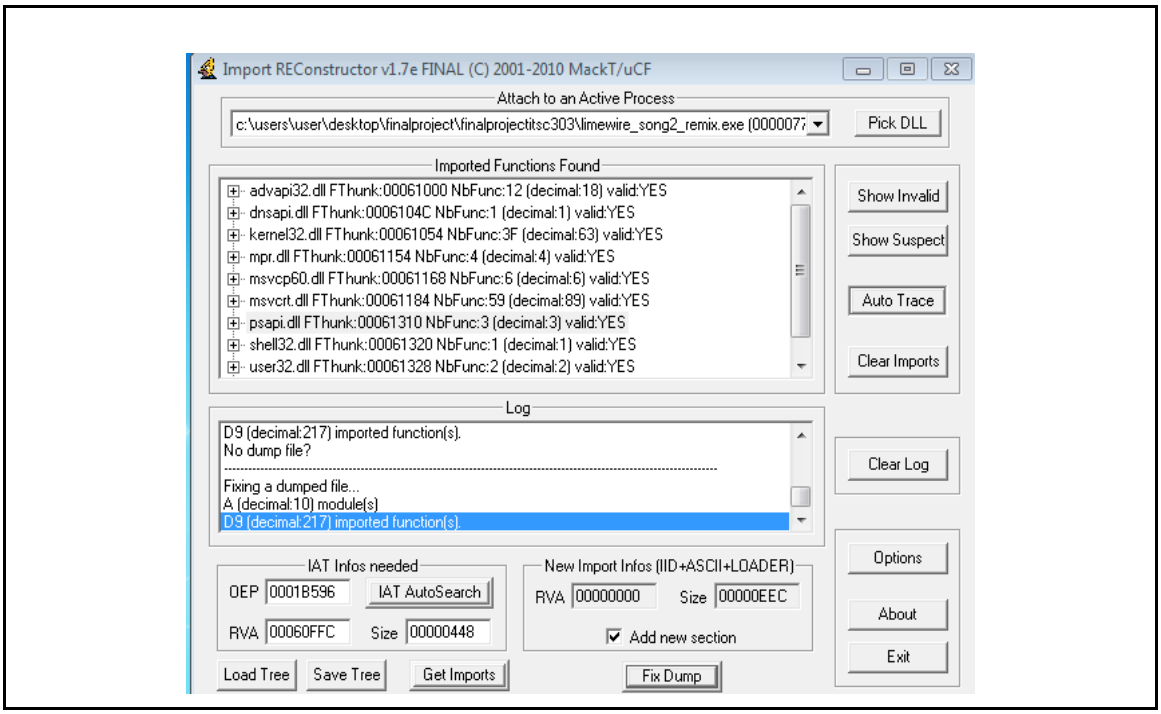
8) Click on IAT AutoSearch



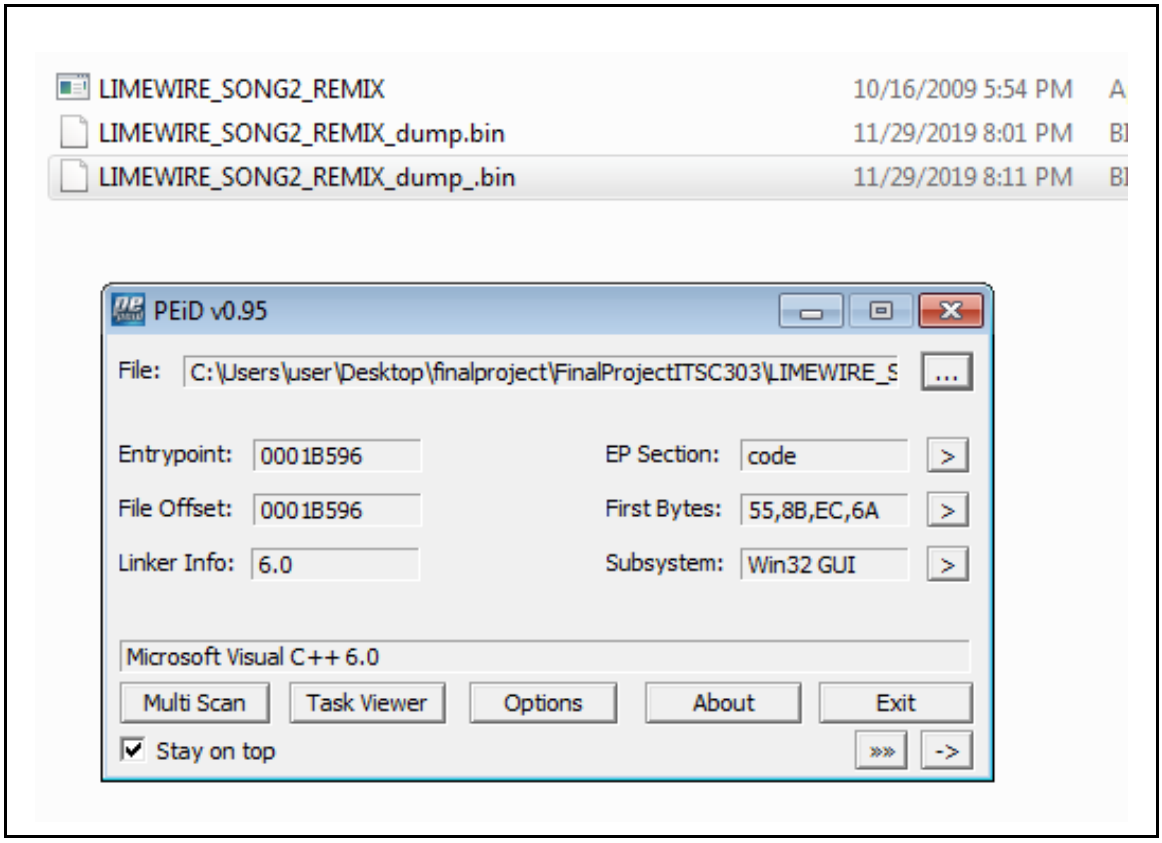
9) Click on Get Imports. Delete the NO valid Thunks.



10) Click on FixDump -> open the previous dump generated by OllyDumpEx.



11) This will generate a new dump file. Named as a LIMEWIRE_SONG2_REMIX.dump after the previous dump file. This is the decompressed executable file.



Following these steps above, we can get an unpacked file. Also, based on the PEiD, it is apparent that this malware is coded in Microsoft Visual C++ 6.0.

Now we will analyze it deeply using Ghidra.

5. Libraries

There are same libraries between original packed sample and unpacked sample. However, there are more Functions from unpacked sample.

Toolset used	Ghidra	
Packed/Unpacked	**Original Packed Sample**	**Unpacked Sample**
Libraries	KERNEL32.DLL ADVAPI32.dll DNSAPI.dll MPR.dll MSVCP60.dll MSVCRT.dll NETAPI32.dll PSAPI.DLL SHELL32.dll USER32.dll WS2_32.dll	KERNEL32.DLL ADVAPI32.dll DNSAPI.dll MPR.dll MSVCP60.dll MSVCRT.dll NETAPI32.dll PSAPI.DLL SHELL32.dll USER32.dll WS2_32.dll
Functions	**(SERVICE)** KERNEL32.DLL::LoadLibraryA KERNEL32.DLL::GetProcAddress KERNEL32.DLL::ExitProcess **(Registry Modification)** ADVAPI32.dll::RegCloseKey (**CreateProcess)** USER32.DLL::wsprintfA KERNEL32.DLL:ExitProcess PSAPI.DLL::EnumProcesses SHELL32.DLL::ShellExecuteA	**(SERVICE)** ADVAPI32.DLL::CreateService ADVAPI32.DLL::StartServiceA ADVAPI32.DLL::CreateServiceA ADVAPI32.DLL::StartServiceCtrlDispatcherA ADVAPI32.DLL::OpenServiceA ADVAPI32.DLL::ControlService ADVAPI32.DLL::DeleteService **(Registry Modification)** ADVAPI32.dll::RegSetValueExA ADVAPI32.dll::RegOpenKeyExA ADVAPI32.dll::RegCreateKeyExA ADVAPI32.dll::RegDeleteValueA ADVAPI32.dll::RegQueryValueExA ADVAPI32.dll::RegisterServiceCtrlHandlerA (**CreateProcess)** KERNEL32.DLL::CreateProcessA KERNEL32.DLL::OpenSCManagerA KERNEL32.DLL::OpenProcess KERNEL32.DLL::CreateEventA

	(Thread Operation) None **(File Operation)** None. **(Connection)** MPR.DLL::WNetAddConnection2W NETAPI32.DLL::NetUseDel	**(Thread Operation)** KERNEL32.DLL::GetCurrentThread KERNEL32.DLL::GetThreadPriority KERNEL32.DLL::SuspendThread KERNEL32.DLL::SetThreadContext KERNEL32.DLL::ResumeThread **(File Operation)** KERNEL32.DLL::CreateFileA KERNEL32.DLL::WriteFile KERNEL32.DLL::DeleteFileA KERNEL32.DLL::CopyFileA KERNEL32.DLL::ReadFile **(Connection)** MPR.DLL::WNetAddConnection2A MPR.DLL::WNetAddConnection2W DNSAPI.dll::DnsQuery_A NETAPI32.DLL::NetUseDel
Noteworthy Functions	PSAPI.DLL::EnumProcesses SHELL32.DLL::ShellExecuteA MPR.DLL::WNetAddConnection2W NETAPI32.DLL::NetUseDel	KERNEL32.DLL::CreateFileA KERNEL32.DLL::WriteFile KERNEL32.DLL::CopyFileA MPR.DLL::WNetAddConnection2A MPR.DLL::WNetAddConnection2W DNSAPI.dll::DnsQuery_A NETAPI32.DLL::NetUseDel

6. Strings analysis using unpacked sample

Strings related to command: There are a lot of strings related to command.

bot commands	
Command	**Description**
bot.about	displays the info the author wants you to see
bot.dns	resolves ip/hostname by dns
bot.execute	makes the bot execute an .exe, exe is hidden when visibility is 0. note that visibility has no effect on gui programs that dont honor the visibility parameter WinMain gets.
bot.flushdns	flushes the bots dns cache
bot.id	displays the bots id which is used to identify which version is running, and only update the bots that need it during an update
bot.longuptime	If uptime > 7 days then bot will respond
bot.nick	changes the nickname of the bot
bot.open	makes the bot open any file using ShellExecuteA or similar functions (in Linux) to open any file that is a registered file type
bot.quit	quits the bot
bot.remove	completely removes the bot from the system
bot.removeallbut	same as bot.remove, but skips bots that have the specified id
bot.rndnick	assigns a new random nickname to the bot
bot.secure	Makes the bot secure by deleting shares and disabling dcom
bot.status	causes the bot to display its status
bot.sysinfo	causes the bot to display system information
bot.unsecure	Makes the unsecure by creating shares and enabling dcom

http commands	
Command	**Description**
http.download	makes the bot download a file from http to the specified directory. supports environment variable expansions.
http.execute	makes the bot download a file from http to the specified directory and execute it. supports environment variable expansions.
http.update	makes the bot download a file from http to the specified directory and update to it if the id doesn't match. supports environment variable expansions.
http.visit	visits an url with a specified referrer

irc commands	
Command	**Description**
irc.action	lets the bot perform an action
irc.disconnect	disconnects the bot from irc
irc.getedu	prints netinfo when the bot is .edu
irc.gethost	prints netinfo when host matches
irc.join	makes the bot join a channel
irc.mode	lets the bot perform a mode change
irc.netinfo	prints netinfo
irc.part	makes the bot part a channel
irc.privmsg	sends a privmsg
irc.quit	quits the bot
irc.raw	sends a raw message to the irc server
irc.reconnect	reconnects to the server
irc.server	changes the server the bot connects to

ddos commands	
Command	**Description**
ddos.httpflood	starts a HTTP flood, can also be used as .visit replacement
ddos.pingflood	starts a Ping flood
ddos.spudpflood	starts a spoofed UDP flood
ddos.stop	stops all ddoses running
ddos.synflood	starts a spoofed SYN flood
ddos.udpflood	starts an UDP flood

harvest commands	
Command	**Description**
harvest.aol	makes the bot get aol stuff
harvest.cdkeys	makes the bot get a list of cdkeys
harvest.emailshttp	makes the bot get a list of emails via http
harvest.emails	makes the bot get a list of emails

redirect commands	
Command	**Description**
redirect.stop	stops all redirects running
redirect.socks	starts a socks4 proxy
redirect.https	starts a https proxy
redirect.http	starts a http proxy
redirect.gre	starts a gre redirect
redirect.tcp	starts a tcp port redirect

scan commands	
Command	**Description**
scan.dcom	AutoScanner
scan.dcom2	scans for dcom2 exploit
scan.locator	scans for locator exploit
scan.netbios	scans weak netbios passwords
scan.stats	stats for working scanners
scan.stop	stops all scans running asap
scan.webdav	scans for iis/webdav exploit
scan.wkssvc	scans for workstation exploit

PC control commands	
Command	**Description**
pctrl.kill	kills a process
pctrl.list	lists all processes

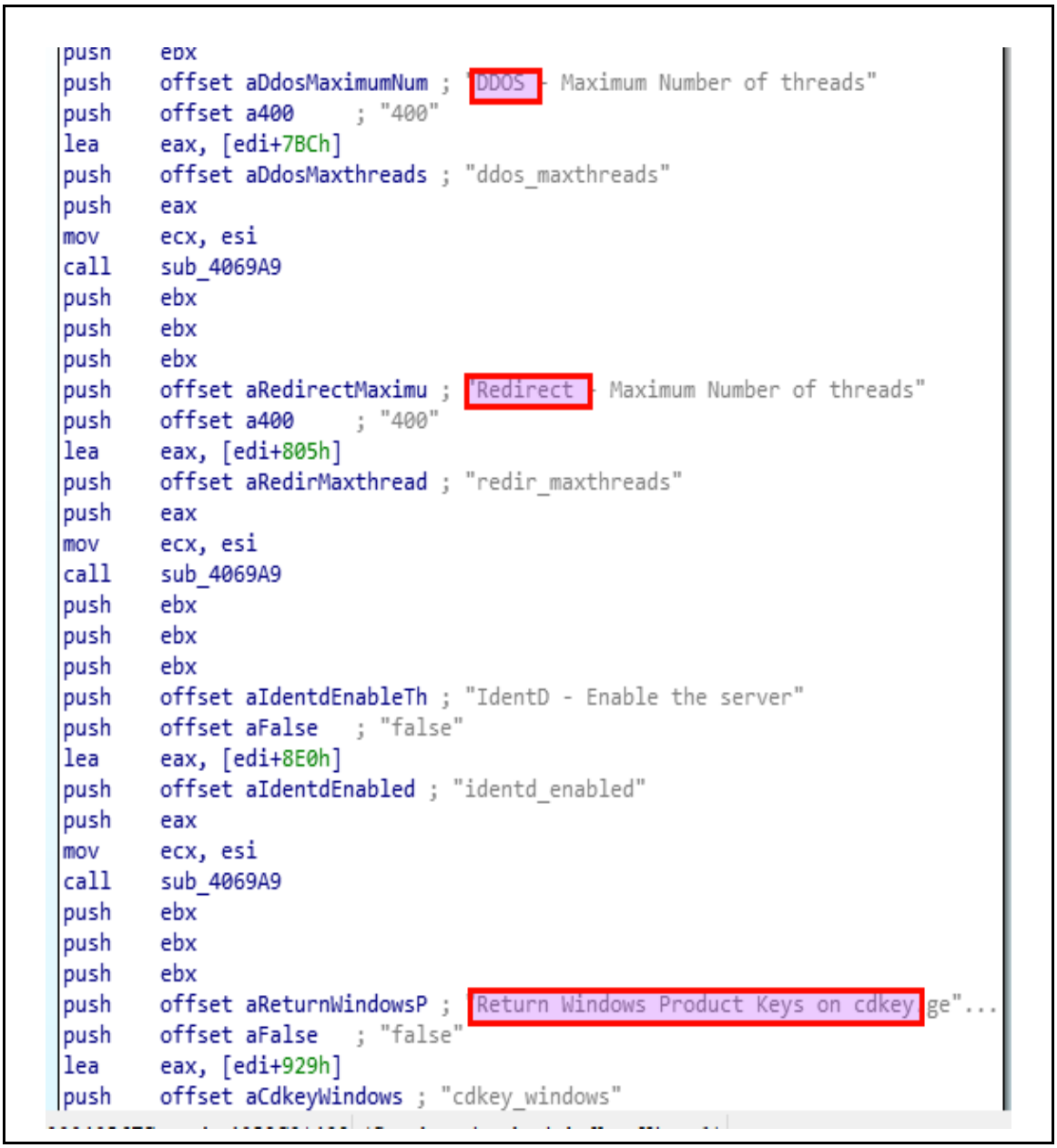
Inst commnads	
Command	**Description**
inst.svcdel	deletes a service from scm
inst.svcadd	adds a service to scm
inst.asdel	deletes an autostart entry
inst.asadd	adds an autostart entry

7. Disassemble & Decompilation

Analysis Disassemble and Decompilation using ghidra and IDA free

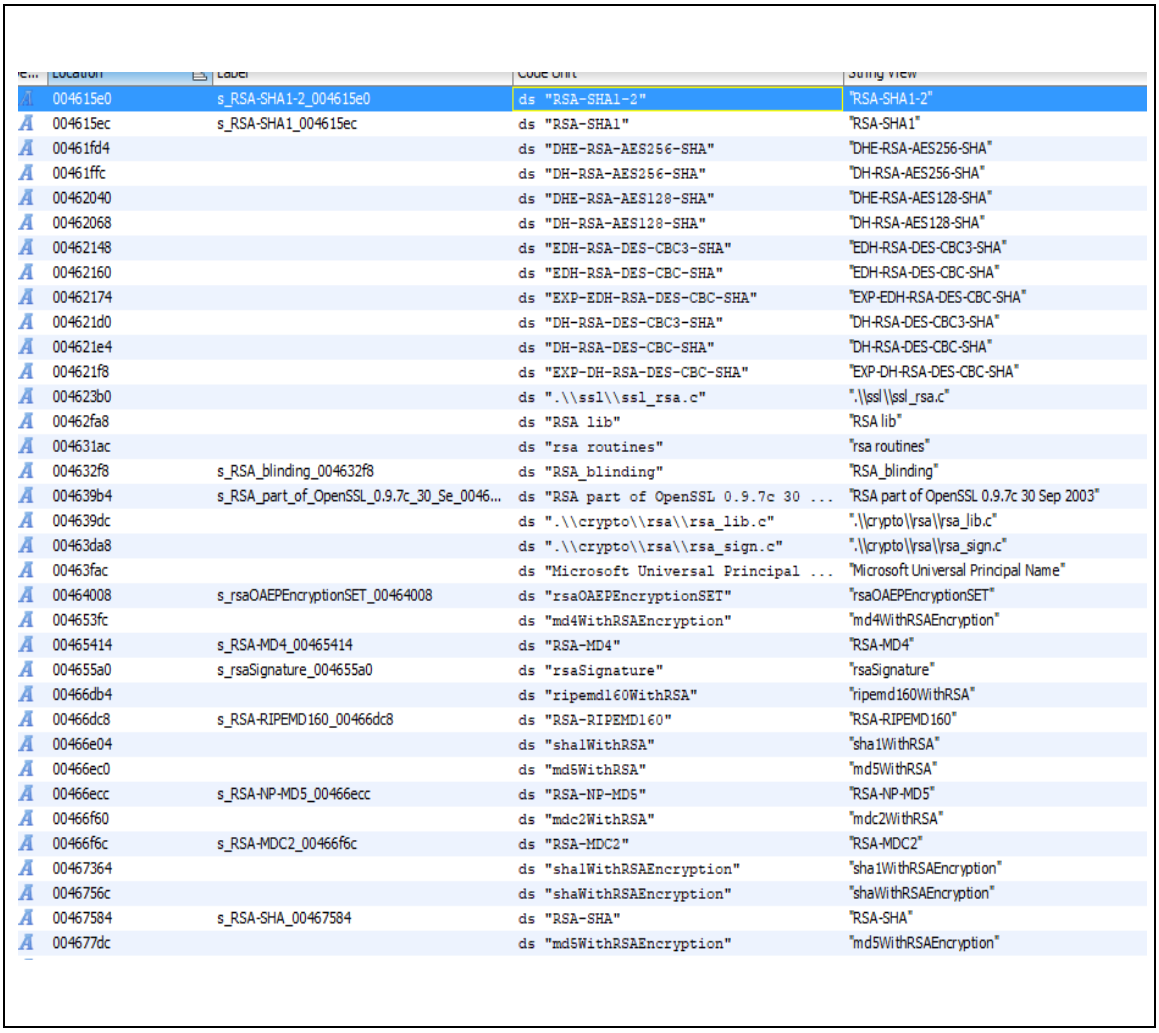
1) Entry Point: 0x0041B596

2) Main Features: Fun_004052c3
Control the irc channel and config for various attacks(DDOS, oal email spam==false, stealing product key==false, scanning local network, sending file)



```
push    ebx
push    offset aScannerMaximum ; "Scanner - Maximum Number of threads"
push    offset a400     ; "400"
lea     eax, [edi+6E1h]
push    offset aScanMaxthreads ; "scan_maxthreads"
push    eax
mov     ecx, esi
call    sub_4069A9
push    ebx
push    ebx
push    ebx
push    offset aScannerAutosca ; "Scanner - Autoscan local network"
push    offset aTrue    ; "true"
lea     eax, [edi+72Ah]
push    offset aScanAuto ; "scan_auto"
push    eax
mov     ecx, esi
call    sub_4069A9
push    ebx
push    ebx
push    ebx
push    offset aScannerAutosca_0 ; "Scanner - Autoscan LAN for NetBIOS"
push    offset aTrue    ; "true"
lea     eax, [edi+773h]
push    offset aScanAutoNb ; "scan_auto_nb"
push    eax
mov     ecx, esi
call    sub_4069A9
push    ebx
push    ebx
push    ebx
push    offset aCsendfileShowC ; "CSendFile - Show connections to the por"...
push    offset aTrue    ; "true"
lea     eax, [edi+972h]
push    offset aCsendfileShow ; "csendfile_show"
push    eax
mov     ecx, esi
call    sub_4069A9
push    ebx
push    ebx
push    ebx
push    offset aAolSpamChannel ; "AOL Spam - Channel name"
push    offset aAolspam ; "#aolspam"
lea     eax, [edi+84Eh]
push    offset aSpamAolChannel ; "spam_aol_channel"
push    eax
mov     ecx, esi
call    sub_4069A9
push    ebx
push    ebx
push    ebx
push    offset aAolSpamChannel ; "AOL Spam - Channel name"
push    offset aFalse   ; "false"
push    offset aSpamAolEnabled ; "spam_aol_enabled"
add     edi, 897h
mov     ecx, esi
push    edi
call    sub_4069A9
push    ebx
mov     ecx, ebp
```

3) Encryption methods (RSA)



Location	Label	Code Unit	String View
004615e0	s_RSA-SHA1-2_004615e0	ds "RSA-SHA1-2"	"RSA-SHA1-2"
004615ec	s_RSA-SHA1_004615ec	ds "RSA-SHA1"	"RSA-SHA1"
00461fd4		ds "DHE-RSA-AES256-SHA"	"DHE-RSA-AES256-SHA"
00461ffc		ds "DH-RSA-AES256-SHA"	"DH-RSA-AES256-SHA"
00462040		ds "DHE-RSA-AES128-SHA"	"DHE-RSA-AES128-SHA"
00462068		ds "DH-RSA-AES128-SHA"	"DH-RSA-AES128-SHA"
00462148		ds "EDH-RSA-DES-CBC3-SHA"	"EDH-RSA-DES-CBC3-SHA"
00462160		ds "EDH-RSA-DES-CBC-SHA"	"EDH-RSA-DES-CBC-SHA"
00462174		ds "EXP-EDH-RSA-DES-CBC-SHA"	"EXP-EDH-RSA-DES-CBC-SHA"
004621d0		ds "DH-RSA-DES-CBC3-SHA"	"DH-RSA-DES-CBC3-SHA"
004621e4		ds "DH-RSA-DES-CBC-SHA"	"DH-RSA-DES-CBC-SHA"
004621f8		ds "EXP-DH-RSA-DES-CBC-SHA"	"EXP-DH-RSA-DES-CBC-SHA"
004623b0		ds ".\\ssl\\ssl_rsa.c"	".\\ssl\\ssl_rsa.c"
00462fa8		ds "RSA lib"	"RSA lib"
00463lac		ds "rsa routines"	"rsa routines"
004632f8	s_RSA_blinding_004632f8	ds "RSA_blinding"	"RSA_blinding"
004639b4	s_RSA_part_of_OpenSSL_0.9.7c_30_Se_0046...	ds "RSA part of OpenSSL 0.9.7c 30 ...	"RSA part of OpenSSL 0.9.7c 30 Sep 2003"
004639dc		ds ".\\crypto\\rsa\\rsa_lib.c"	".\\crypto\\rsa\\rsa_lib.c"
00463da8		ds ".\\crypto\\rsa\\rsa_sign.c"	".\\crypto\\rsa\\rsa_sign.c"
00463fac		ds "Microsoft Universal Principal ...	"Microsoft Universal Principal Name"
00464008	s_rsaOAEPEncryptionSET_00464008	ds "rsaOAEPEncryptionSET"	"rsaOAEPEncryptionSET"
004653fc		ds "md4WithRSAEncryption"	"md4WithRSAEncryption"
00465414	s_RSA-MD4_00465414	ds "RSA-MD4"	"RSA-MD4"
004655a0	s_rsaSignature_004655a0	ds "rsaSignature"	"rsaSignature"
00466db4		ds "ripemd160WithRSA"	"ripemd160WithRSA"
00466dc8	s_RSA-RIPEMD160_00466dc8	ds "RSA-RIPEMD160"	"RSA-RIPEMD160"
00466e04		ds "sha1WithRSA"	"sha1WithRSA"
00466ec0		ds "md5WithRSA"	"md5WithRSA"
00466ecc	s_RSA-NP-MD5_00466ecc	ds "RSA-NP-MD5"	"RSA-NP-MD5"
00466f60		ds "mdc2WithRSA"	"mdc2WithRSA"
00466f6c	s_RSA-MDC2_00466f6c	ds "RSA-MDC2"	"RSA-MDC2"
00467364		ds "sha1WithRSAEncryption"	"sha1WithRSAEncryption"
0046756c		ds "shaWithRSAEncryption"	"shaWithRSAEncryption"
00467584	s_RSA-SHA_00467584	ds "RSA-SHA"	"RSA-SHA"
004677dc		ds "md5WithRSAEncryption"	"md5WithRSAEncryption"

4) Bot Control Function: sub_4022B0

```
sub_4022B0 proc near
push    ebx
push    esi
mov     esi, ecx
push    edi
mov     ebx, esi
mov     edi, offset dword_486210
neg     ebx
lea     eax, [esi+11h]
mov     ecx, edi
sbb     ebx, ebx
and     ebx, eax
lea     eax, [esi+67h]
push    ebx             ; int
push    offset aDisconnectsThe ; "disconnects the bot from irc"
push    offset aIrcDisconnect ; "irc.disconnect"
push    eax             ; int
call    sub_404FC4
push    ebx             ; int
push    offset aLetsTheBotPerf ; "lets the bot perform an action"
lea     eax, [esi+95h]
push    offset aIrcAction ; "irc.action"
push    eax             ; int
mov     ecx, edi
call    sub_404FC4
push    ebx             ; int
push    offset aPrintsNetinfoW ; "prints netinfo when the bot is .edu"
lea     eax, [esi+0C3h]
push    offset aIrcGetedu ; "irc.getedu"
push    eax             ; int
mov     ecx, edi
call    sub_404FC4
push    ebx             ; int
push    offset aPrintsNetinfoW_0 ; "prints netinfo when host matches"
lea     eax, [esi+0F1h]
push    offset aIrcGethost ; "irc.gethost"
push    eax             ; int
```

5) Bot Control function: FUN_004035bc

```
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xaa1),s_bot.about_00474778,
             s_displays_the_info_the_author_wan_00474784,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xa45),s_bot.die_0047475c,
             s_terminates_the_bot_00474764,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xb87),s_bot.dns_00474738,
             s_resolves_ip/hostname_by_dns_00474740,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xbe3),s_bot.execute_0047470c,
             s_makes_the_bot_execute_a_.exe_00474718,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xa73),s_bot.id_004746e0,
             s_displays_the_id_of_the_current_c_004746e8,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xb2b),s_bot.nick_004746b4,
             s_changes_the_nickname_of_the_bot_004746c0,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xb59),s_bot.open_00474690,
             s_opens_a_file_(whatever)_0047469c,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0x9bb),s_bot.remove_00474674,
             s_removes_the_bot_00474680,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xbb5),s_bot.removeallbut_00474638,
             s_removes_the_bot_if_id_does_not_m_0047464c,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xa17),s_bot.rndnick_00474600,
             s_makes_the_bot_generate_a_new_ran_0047460c,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0x9e9),s_bot.status_004745e4,
             s_gives_status_004745f0,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xacf),s_bot.sysinfo_004745bc,
             s_displays_the_system_info_004745c8,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xafd),s_bot.longuptime_00474580,
             s_If_uptime_>_7_days_then_bot_will_00474590,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xc11),s_bot.quit_00474574,
             s_quits_the_bot_00474lcc,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xc3f),s_bot.flushdns_00474548,
             s_flushes_the_bots_dns_cache_00474558,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xc6d),s_bot.secure_004745lc,
             s_delete_shares_/_disable_dcom_00474528,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xc9b),s_bot.unsecure_004744f0,
             s_enable_shares_/_enable_dcom_00474500,extraout_ECX);
FUN_00404fc4(&DAT_00486210,(void *)(extraout_ECX + 0xcc9),s_bot.command_004744c4,
             s_runs_a_command_with_system()_004744d0,extraout_ECX);
FUN_004061c9();
```

6) C2 server: irc.foxlink.net

```
                    s_irc.foxlink.net_004752d0

004752d0 69 72 63       ds          "irc.foxlink.net"
```

7) Execute ShellCommand: FUN_00404c9d

```
void FUN_00404c9d(void)

{
  uchar *_Source;
  size_t _Count;
  uchar *_Dest;
  uint uVar1;
  int iVar2;
  char *_Source_00;
  int extraout_ECX;
  undefined4 *puVar3;
  int unaff_EBP;
  undefined4 *in_FS_OFFSET;

  FUN_0041b220();
  if (*(void **)(extraout_ECX + 0x110) != (void *)0x0) {
    operator_delete(*(void **)(extraout_ECX + 0x110));
  }
  _Count = strlen(*(char **)(unaff_EBP + 8));
  _Dest = (uchar *)operator_new(_Count + 1);
  _Source = *(uchar **)(unaff_EBP + 8);
  *(uchar **)(extraout_ECX + 0x110) = _Dest;
  _mbscpy(_Dest,_Source);
  *(undefined *)(extraout_ECX + 4) = 0;
  *(undefined *)(extraout_ECX + 9) = 0;
  *(undefined *)(extraout_ECX + 10) = 0;
  *(undefined *)(extraout_ECX + 0xb) = 0;
  *(undefined4 *)(extraout_ECX + 5) = 0;
  FUN_004059d9((void *)(unaff_EBP + -0x2c),*(char **)(extraout_ECX + 0x110));
  *(undefined4 *)(unaff_EBP + -4) = 0;
  *(undefined4 *)(unaff_EBP + -0x14) = 0;
  FUN_00405fe4();
  *(undefined *)(unaff_EBP + -4) = 1;
  uVar1 = FUN_00405bb8();
  *(bool *)(unaff_EBP + 0xb) = uVar1 != 0;
  *(undefined *)(unaff_EBP + -4) = 0;
  FUN_00405a6f((undefined4 *)(unaff_EBP + -0x59));
  if (*(char *)(unaff_EBP + 0xb) != '\0') {
    *(undefined4 *)(unaff_EBP + -0x10) = 1;
```

8) CreateService: FUN_00407849

```
uint FUN_00407849(void)

{
  SC_HANDLE hService;
  LPCSTR lpServiceName;
  uint uVar1;
  BOOL BVar2;
  undefined4 extraout_ECX;
  int unaff_EBP;
  undefined4 *in_FS_OFFSET;
  bool bVar3;
  LPCSTR lpDisplayName;
  DWORD dwDesiredAccess;
  DWORD dwServiceType;
  DWORD dwStartType;
  DWORD dwErrorControl;
  LPCSTR lpBinaryPathName;
  LPCSTR lpLoadOrderGroup;
  LPDWORD lpdwTagId;
  LPCSTR lpDependencies;
  LPCSTR lpServiceStartName;
  LPCSTR lpPassword;

  FUN_0041b220();
  bVar3 = false;
  *(undefined4 *)(unaff_EBP + -0x14) = extraout_ECX;
  hService = OpenSCManagerA((LPCSTR)0x0,s_ServicesActive_004757a4,0xf003f);
  *(SC_HANDLE *)(unaff_EBP + -0x10) = hService;
  if (hService == (SC_HANDLE)0x0) {
    uVar1 = 0;
    goto LAB_0040796d;
  }
  FUN_004059a6((undefined4 *)(unaff_EBP + -0x48));
  *(undefined4 *)(unaff_EBP + -4) = 0;
  FUN_004061c9();
  FUN_004061c9();
  FUN_00405d3e((void *)(unaff_EBP + -0x48),s_"%s"_%s_0047579c);
  lpPassword = (LPCSTR)0x0;
  lpServiceStartName = (LPCSTR)0x0;
```

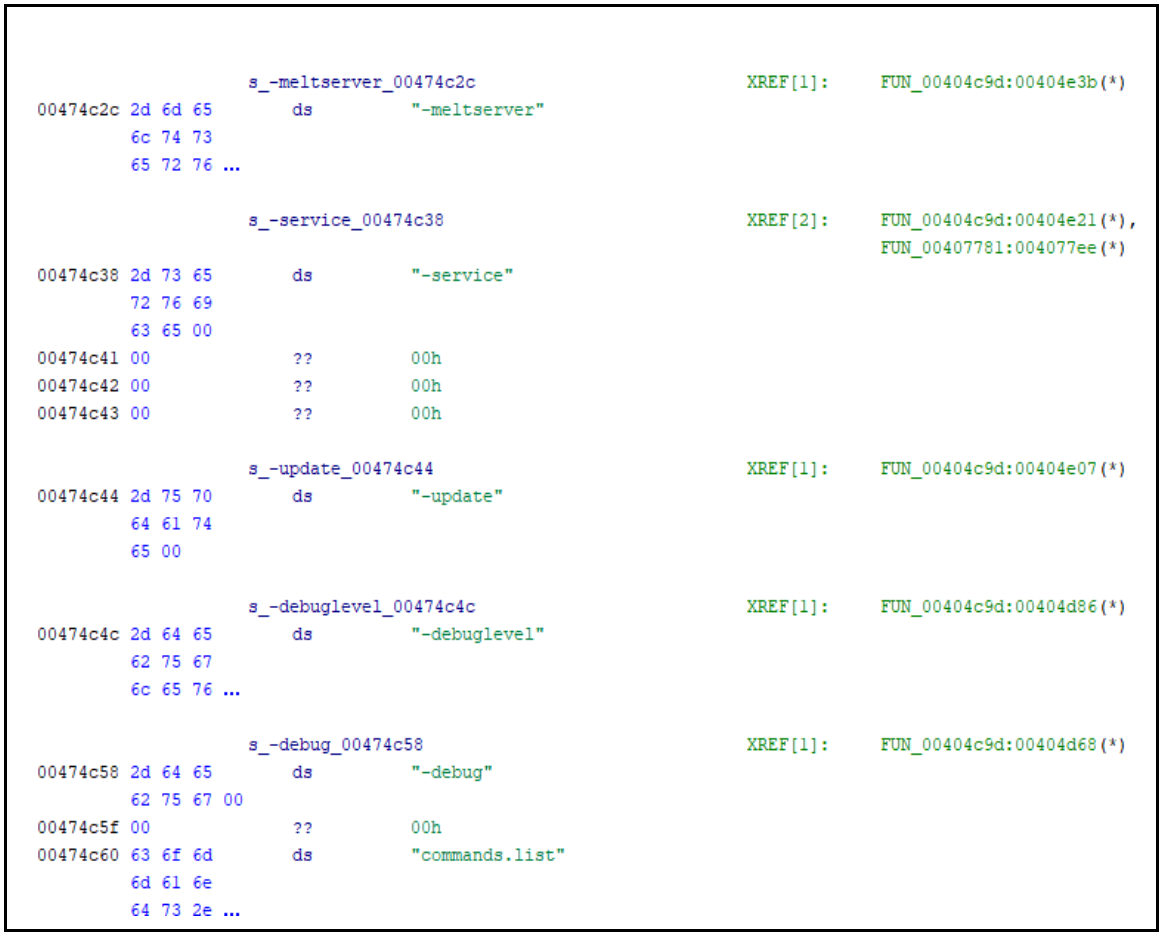
9) Network Connection & Attack Process

```
undefined4 FUN_00409acd(void)

{
  uint *puVar1;
  uint uVar2;
  int iVar3;
  undefined4 uVar4;
  int extraout_ECX;
  int unaff_EBP;
  undefined4 *in_FS_OFFSET;
  char **ppcVar5;

  FUN_0041b220();
  *(undefined *)(extraout_ECX + 0x24f5) = 1;
  FUN_00409f70((void *)(extraout_ECX + 0x2554));
  if (*(int *)(unaff_EBP + 0xc) == 0) {
    ppcVar5 = &_Str2_004860b4;
  }
  else {
    ppcVar5 = *(char ***)(unaff_EBP + 0xc);
  }
  FUN_00405aa2((void *)(extraout_ECX + 0x2520),(char *)ppcVar5);
  if (*(int *)(unaff_EBP + 8) == 0) {
    ppcVar5 = &_Str2_004860b4;
  }
  else {
    ppcVar5 = *(char ***)(unaff_EBP + 8);
  }
  FUN_00405aa2((void *)(extraout_ECX + 0x2535),(char *)ppcVar5);
  FUN_00404c9d();
  if (*(char *)(extraout_ECX + 0x42) != '\0') {
    FUN_00405951((void *)(extraout_ECX + 0x152),*(undefined4 *)(extraout_ECX + 0x43));
  }
  FUN_00405d3e((void *)(extraout_ECX + 0x250b),s_Agobot3_(%s)_"%s"_on_"%s"_00475904);
  WSAStartup(0x202,(LPWSADATA)(unaff_EBP + -0x1e0));
  FUN_0040662b();
  if ((*(char *)(extraout_ECX + 0x48) != '\0') || (*(char *)(extraout_ECX + 0x7c7) != '\0')
    FUN_00406661((uint *)(extraout_ECX + 0x244a));
  }
```

10) Executable Options List



```
                    s_-meltserver_00474c2c                          XREF[1]:     FUN_00404c9d:00404e3b(*)
00474c2c 2d 6d 65       ds          "-meltserver"
         6c 74 73
         65 72 76 ...

                    s_-service_00474c38                             XREF[2]:     FUN_00404c9d:00404e21(*),
                                                                                 FUN_00407781:004077ee(*)
00474c38 2d 73 65       ds          "-service"
         72 76 69
         63 65 00
00474c41 00             ??          00h
00474c42 00             ??          00h
00474c43 00             ??          00h

                    s_-update_00474c44                              XREF[1]:     FUN_00404c9d:00404e07(*)
00474c44 2d 75 70       ds          "-update"
         64 61 74
         65 00

                    s_-debuglevel_00474c4c                          XREF[1]:     FUN_00404c9d:00404d86(*)
00474c4c 2d 64 65       ds          "-debuglevel"
         62 75 67
         6c 65 76 ...

                    s_-debug_00474c58                               XREF[1]:     FUN_00404c9d:00404d68(*)
00474c58 2d 64 65       ds          "-debug"
         62 75 67 00
00474c5f 00             ??          00h
00474c60 63 6f 6d       ds          "commands.list"
         6d 61 6e
         64 73 2e ...
```

11) Anti-virus process killing

This malware can detect 450 various Anti-virus processes and terminate them.

```
uint __cdecl FUN_0040ccb5(char *param_1)

{
  HANDLE hProcess;
  BOOL BVar1;
  int iVar2;
  undefined4 unaff_EBX;
  uint uVar3;
  HMODULE local_10;
  uint unaff_EDI;
  byte local_5;

  FUN_0041b260();
  hProcess = (HANDLE)EnumProcesses((DWORD *)&stack0xffffeeec,0x1000,(LPDWORD)&stack0xfffffff4
  if (hProcess != (HANDLE)0x0) {
    local_5 = (byte)((uint)unaff_EBX >> 0x18);
    uVar3 = 0;
    if (unaff_EDI >> 2 != 0) {
      do {
        _mbscpy(&stack0xfffffeec,(uchar *)s_unknown_0047603c);
        hProcess = OpenProcess(0x411,0,*(DWORD *)(&stack0xffffeeec + uVar3 * 4));
        if (hProcess != (HANDLE)0x0) {
          BVar1 = EnumProcessModules(hProcess,(HMODULE *)&stack0xfffffff0,4,
                                     (LPDWORD)&stack0xfffffff4);
          if (BVar1 != 0) {
            GetModuleBaseNameA(hProcess,local_10,&stack0xfffffeec,0x104);
            iVar2 = _strcmpi(&stack0xfffffeec,param_1);
            if (iVar2 == 0) {
              TerminateProcess(hProcess,0);
              unaff_EBX = 0x1000000;
            }
          }
          hProcess = (HANDLE)CloseHandle(hProcess);
        }
        local_5 = (byte)((uint)unaff_EBX >> 0x18);
        uVar3 = uVar3 + 1;
      } while (uVar3 < unaff_EDI >> 2);
    }
    return (uint)hProcess & 0xffffff00 | (uint)local_5;
```

Decompile: FUN_0040cf01 - (LIMEWIRE_SONG2_REMIX_dump_.bin)

```
160   pcVar1 = s_ACKWIN32.EXE_00477970;
161   local_720 = s_ACKWIN32.EXE_00477970;
162   local_71c[0] = s_ADVXDWIN.EXE_00477960;
163   local_71c[1] = s_AGENTSVR.EXE_00477950;
164   local_71c[2] = s_ALERTSVC.EXE_00477940;
165   local_71c[3] = s_ALOGSERV.EXE_00477930;
166   local_70c = s_AMON9X.EXE_00477924;
167   local_708 = s_ANTI-TROJAN.EXE_00477914;
168   local_704 = s_ANTIVIRUS.EXE_00477904;
169   local_700 = s_ANTS.EXE_004778f8;
170   local_6fc = s_APIMONITOR.EXE_004778e8;
171   local_6f8 = s_APLICA32.EXE_004778d8;
172   local_6f4 = s_APVXDWIN.EXE_004778c8;
173   local_6f0 = s_ATCON.EXE_004778bc;
174   local_6ec = s_ATGUARD.EXE_004778b0;
175   local_6e8 = s_ATRO55EN.EXE_004778a0;
176   local_6e4 = s_ATUPDATER.EXE_00477890;
177   local_6e0 = s_ATWATCH.EXE_00477884;
178   local_6dc = s_AUPDATE.EXE_00477878;
179   local_6d8 = s_AUTODOWN.EXE_00477868;
180   local_6d4 = s_AUTOUPDATE.EXE_00477858;
181   local_6d0 = s_AVCONSOL.EXE_00477848;
182   local_6cc = s_AVE32.EXE_0047783c;
183   local_6c8 = s_AVGCC32.EXE_00477830;
184   local_6c4 = s_AVGCTRL.EXE_00477824;
185   local_6c0 = s_AVGNT.EXE_00477818;
186   local_6bc = s_AVGSERV.EXE_0047780c;
187   local_6b8 = s_AVGSERV9.EXE_004777fc;
188   local_6b4 = s_AVGUARD.EXE_004777f0;
189   local_6b0 = s_AVGW.EXE_004777e4;
190   local_6ac = s_AVNT.EXE_004777d8;
191   local_6a8 = s_AVP.EXE_004777d0;
192   local_6a4 = s_AVP32.EXE_004777c4;
193   local_6a0 = s_AVPCC.EXE_004777b8;
194   local_69c = s_AVPDOS32.EXE_004777a8;
195   local_698 = s_AVPM.EXE_0047779c;
196   local_694 = s_AVPTC32.EXE_00477790;
197   local_690 = s_AVPUPD.EXE_00477784;
198   local_68c = s_AVWIN95.EXE_00477778;
```

12) Word list

word lists for brute force local user account.

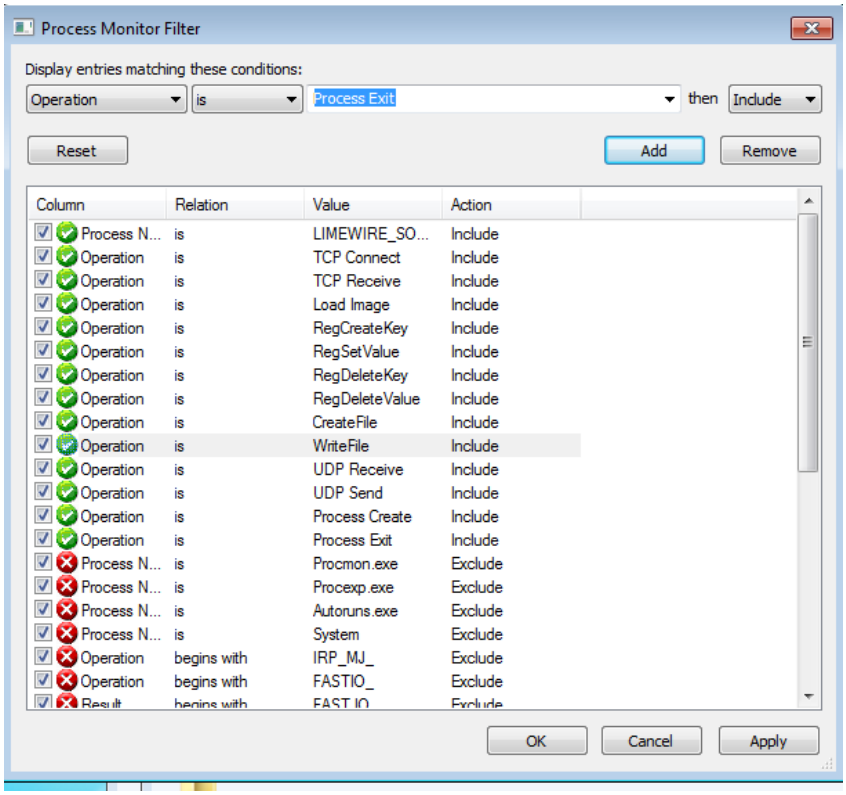
```
text:0047BE45 ; ------------------------------------------------------
text:0047BE46                 dw 47h
text:0047BE48                 dd offset aAbc123       ; "abc123"
text:0047BE4C                 dd offset aPassword123  ; "password123"
text:0047BE50                 dd offset aRed123       ; "red123"
text:0047BE54                 dd offset aQwerty       ; "qwerty"
text:0047BE58                 dd offset aAdmin123     ; "admin123"
text:0047BE5C                 dd offset aZxcvbnm      ; "zxcvbnm"
text:0047BE60                 dd offset aPoiuytrewq   ; "poiuytrewq"
text:0047BE64                 dd offset aPwd          ; "pwd"
text:0047BE68                 dd offset aPass         ; "pass"
text:0047BE6C                 dd offset aLove         ; "love"
text:0047BE70                 dd offset aMypc         ; "mypc"
text:0047BE74                 dd offset aMypass       ; "mypass"
text:0047BE78                 dd offset aPw           ; "pw"
text:0047BE7C                 dd offset Str2
text:0047BE80                 dd 0
text:0047BE84                 dd offset aAdmin_0      ; "admin$"
text:0047BE88                 dd offset aC_0          ; "c$"
text:0047BE8C                 dd offset aD_2          ; "d$"
text:0047BE90                 dd offset aE_0          ; "e$"
text:0047BC31 ; ------------------------------------------------------
text:0047BC34                 dd offset aOem          ; "OEM"
text:0047BC38                 dd offset aRoot         ; "root"
text:0047BC3C                 dd offset aWwwadmin     ; "wwwadmin"
text:0047BC40                 dd offset aLogin        ; "login"
text:0047BC44                 dd offset Str2
text:0047BC48                 dd offset aOwner        ; "owner"
text:0047BC4C                 dd offset aMary         ; "mary"
text:0047BC50                 dd offset aAdmins       ; "admins"
text:0047BC54                 dd offset aComputer     ; "computer"
text:0047BC58                 dd offset aXp_0         ; "xp"
text:0047BC5C                 dd offset aOwner_0      ; "OWNER"
text:0047BC60                 dd offset aMysql        ; "mysql"
text:0047BC64                 dd offset aDatabase     ; "database"
text:0047BC68                 dd offset aTeacher      ; "teacher"
text:0047BC6C                 dd offset aStudent      ; "student"
text:0047BC70                 dd 2 dup(0)
text:0047BC78                 dd offset aAdmin        ; "admin"
text:0047BC7C ; ------------------------------------------------------

text:0047BD85 ; ------------------------------------------------------
text:0047BD88                 dd offset aXxx          ; "xxx"
text:0047BD8C                 dd offset aOwner        ; "owner"
text:0047BD90                 dd offset aLogin        ; "login"
text:0047BD94                 dd offset aLogin_0      ; "Login"
text:0047BD98 ; ------------------------------------------------------
```

[Dynamic Analysis]

7. Analysis malware's behaviors (Procmon & Regshot)

Before capture the event using procmon, we set a filter with the specific operation we want to capture from the file.



run as normal user

1) Events Statistics

Event Identified	Number of Events
CreateFile	168
Load Image	39
RegCreatKey	10
TCP Connect	1
TCP Receive	12
Total Events	230

2) Suspicious Events Identified

Event Identified	Values Associated	Description of Behavior
CreateFile: NAME NOT FOUND	C:\Windows\Prefetch\scvhostn.EXE-6D712D90.pf	Read from this non-existent file, likely save the malicious binary in .pf format.
CreateFile: NAME NOT FOUND	C:\Windows\System32\scvhostn.exe -meltserver	Read from these non-existent files.
CreateFile - Access Denied	C:\Windows\System32\scvhostn.exe	Totally 84 attempts.
RegCreatKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	Read registry settings for network connection
TCP Connection	user-PC:49335 -> 131.9.16.172.in-addr.arpa:6667	Win7 sent Reverse DNS lookups DNS query to the Remnux vm (fakedns server)172.16.9.131 through port 6667.
TCP Receive	user-PC:49335 -> 131.9.16.172.in-addr.arpa:6667	Received DNS Resolution Response from Remnux through port:6667. DNS server resolved the evil url.

3) Suspicious Files Identified

File Path	Contents of the FIle	Description of Behavior
C:\Windows\System32\scvhostn.exe	Highly possible be the malicious binary.	The malware sample tried to create and write data into scvhostn.exe as a means of File-based persistence. The file was put under /System32 and has a similar name as Windows legitimate file Scvhost.exe.
C:\Windows\Prefetch\LIMEWIRE_SONG2_REMIX.EXE-6B84FDF3.pf		The malware sample tried to read from this non-exist file. This malware might spread as a .pf file.

4) Suspicious Registry Activity Identified

Registry Key	Key Value Added/Changed/ Deleted	Description of Behavior or Significance of the Finding
HKU\S-1-5-21-1975118509-2726110912-3963092078-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\0\1	Keys Added: 60	S-1-5-21-1975118509-2726110912-3963092078-1000-> Admin user id. The BagMRU is the database of folders which are currently stored. It has the location of the folder and which ID (NodeSlot) it has in the Bags tree.
HKU\S-1-5-21-1975118509-2726110912-3963092078-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\52\Shell	added	Added keys under the Bag tree. These keys infect registry for getting automatic restart. They can be used for data enumeration to identify the contents of long gone removable devices, and show the contents of previously mounted encrypted volumes or for deleting folders for malware persistence.
HKU\S-1-5-21-1975118509-2726110912-3963092078-1000\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\	Values added: 279	Adding shell extension handler uses for malware persistence.
\Software\Classes\Local Settings\MuiCache\5\52C64B7E\@%SystemRoot%\System32\FirewallControlPanel.dll,-1: "Windows Firewall"	Values added	Add values to Windows Firewall settings for allowing malware traffic passing through.
Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\AllFolders\Shell\MinPos1024x768x96(1)	Values modified: 44	change the parameter value for bagshell tree.
\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\HRZR_PGYFRFFVBA:	Values modified	UserAssist records the information related to programs run by administrator on system.In Windows 7 {CEBFF5CD-ACE2-4F4F-9178-9926F41749EA} is a list of applications, files, links, and other objects that have been accessed.

5) Suspicious Registry Activity Identified

Traffic Information	Source and Destination (in that order)	Description of Behavior or Significance of the Finding
Outbound connection made to DNS server	172.16.9.171 172.16.9.131(fakedns)	Connection initiated from the affected host to the DNS server using TCP protocols for DNS resolution

Run as administrator

1) Events Statistics

Events Statistics - Malware Sample Process

Event Identified	Number of Events
CreateFile	59
Process Create	1
Load Image	31
WriteFile	5
Total Events	96

Events Statistics - scvhostn.exe

Event Identified	Number of Events (PID 3096)	Number of Events (PID 2096)
CreateFile	30	18
Process Create	28	9
Load Image	0	10
Total Events	58	37

2) Suspicious Events Identified

Event Identified	Values Associated	Description of Behavior
CreateFile	C:\Windows\system32\scvhostn.exe	Create file under system32 directory, disguise as a legitimate operating system file(scvhostn.exe).
Name Not Found	C:\Windows\Prefetch\scvhostn.EXE-6D712D90.pf	Read from this non-existent file, likely saved the malicious binary in .pf format
WriteFile	C:\Windows\system32\scvhostn.exe	Write malicious code to scvhostn.exe.
RegCreateKey	HKLM\System\CurrentControlSet\Services\Tcpip\Parameters	Create registry keys for connecting back to the C2 server.
Process Create	C:\Windows\system32\scvhostn.exe	Execute scvhostn.exe as a way of persistence.

3) Suspicious Files Identified

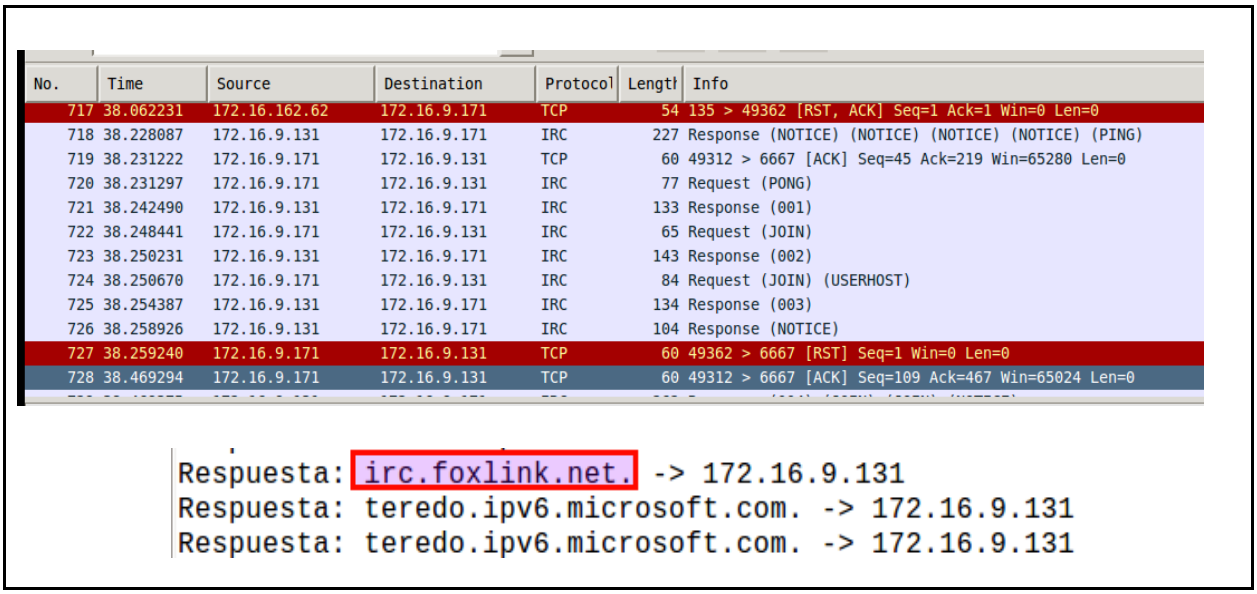
File Path	Contents of the File	Description of Behavior
C:\Windows\system32\scvhostn.exe	Malicious binary. Likely a copy of the original malware sample.	scvhostn.exe was created and written with malicious binary for persistence. It is put under /System32 disguising as a legitimate Windows file(scvhostn.exe)

4) Suspicious Registry Activity Identified

Registry Key	Key Value Added/Changed/Deleted	Description of Behavior or Significance of the Finding
HKLM\SYSTEM\ControlSet001\services\pB HKLM\SYSTEM\CurrentControlSet\services\pB	Key added: 2	Create new service driver by adding key pB. HKLM\SYSTEM\CurrentControlSet\services is registry tree stores information about each service on the system.
-	Values added: 15	Defined pB service driver parameters: Type, start, error control, imagepath, display name, object name, failureaction
HKLM\SYSTEM\ControlSet001\services\pB\ImagePath:""C:\Windows\system32\SCVHOSTN.exe" -service"	Values added	ImagePath specifies the fully qualified path of the driver's image file. Windows creates this value by using the required ServiceBinary entry in the driver's INF file.
HKLM\SYSTEM\ControlSet001\services\pB\DisplayName: "Microsoft Windows Connection Firewall"	Values added	Set the pB displayname as Microsoft Windows Connection Firewall as a mean of detection evasion.
HKU\S-1-5-21-1975118509-2726110912-3963092078-1000\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\MRUListEx:	Value modified	Changed value of the Bagshell setting for the administrator account.
HKU\S-1-5-21-1975118509-2726110912-3963092078-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\HRZR_PGYFRFFVBA	Value modified	UserAssist records the information related to programs run by administrator on system .In Windows 7 {CEBFF5CD-ACE2-4F4F-9178-9926F41749EA} is a list of applications, files, links, and other objects that have been accessed.

8. Network connection (Wireshark, REMNUX)

Compromised machine reached back to the C2 server through IRC protocol to join the irc channel as an irc client. The bot master then can issue command and control the irc bot.



No.	Time	Source	Destination	Protocol	Length	Info
717	38.062231	172.16.162.62	172.16.9.171	TCP	54	135 > 49362 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
718	38.228087	172.16.9.131	172.16.9.171	IRC	227	Response (NOTICE) (NOTICE) (NOTICE) (NOTICE) (PING)
719	38.231222	172.16.9.171	172.16.9.131	TCP	60	49312 > 6667 [ACK] Seq=45 Ack=219 Win=65280 Len=0
720	38.231297	172.16.9.171	172.16.9.131	IRC	77	Request (PONG)
721	38.242490	172.16.9.131	172.16.9.171	IRC	133	Response (001)
722	38.248441	172.16.9.171	172.16.9.131	IRC	65	Request (JOIN)
723	38.250231	172.16.9.131	172.16.9.171	IRC	143	Response (002)
724	38.250670	172.16.9.171	172.16.9.131	IRC	84	Request (JOIN) (USERHOST)
725	38.254387	172.16.9.131	172.16.9.171	IRC	134	Response (003)
726	38.258926	172.16.9.131	172.16.9.171	IRC	104	Response (NOTICE)
727	38.259240	172.16.9.171	172.16.9.131	TCP	60	49362 > 6667 [RST] Seq=1 Win=0 Len=0
728	38.469294	172.16.9.171	172.16.9.131	TCP	60	49312 > 6667 [ACK] Seq=109 Ack=467 Win=65024 Len=0

```
Respuesta: irc.foxlink.net. -> 172.16.9.131
Respuesta: teredo.ipv6.microsoft.com. -> 172.16.9.131
Respuesta: teredo.ipv6.microsoft.com. -> 172.16.9.131
```

There are SMTP and POP3 protocol ports are open for email traffic. This malware has the ability to spread through oal email spam. Also, it uses http protocol to use http commands such as http.visit, http.update, http.execute, http.download. Https ports (443) are open for ssl connection, which is used for sending back the file system encryption private keys back to the C2 server.

```
Configuration file parsed successfully.
=== INetSim main process started (PID 1528) ===
Session ID:     1528
Listening on:   172.16.9.131
Real Date/Time: 2019-11-29 14:38:32
Fake Date/Time: 2019-11-29 14:38:32 (Delta: 0 seconds)
Forking services...
 * irc_6667_tcp - started (PID 1538)
 * smtp_25_tcp - started (PID 1532)
 * smtps_465_tcp - started (PID 1533)
 * ftps_990_tcp - started (PID 1537)
 * https_443_tcp - started (PID 1531)
 * pop3_110_tcp - started (PID 1534)
 * http_80_tcp - started (PID 1530)
 * ftp_21_tcp - started (PID 1536)
 * pop3s_995_tcp - started (PID 1535)
done.
```

[Signature Creation]

Using clamscan with the signature, we can detect the malware easily. There are lots of types of signature and we make 4 signatures which are full hash, section, body-based detection and logical signature.

Signature Type	Signature	Description
Full File	e652d7c27f43ac16a2ec6ee5491166674882d458e49dd667f97b97992e912d1c:220672:Hash.Backdoor.UAE	Full hash signature using Sha-256
Section	218624:a5046320110153f3e4da8a3f7528b2bc3b06b4254425fc147d2baf1b7d11ec72:SecHash.Backdoor.UAE	using .text section
Body-based Detection	PE.Backdoor.UAE:1:SE2:3f3f316f75745f6f665f72616e676540737464404055414540585a	using one specific string for the signature
Logical	PE.Backdoor.UAE.A;Target:1;0&1;3f3f316f75745f6f665f72616e6765407374644040554145405 85a;28242e375c53506c2b73676d	using two specific strings for the signature with &

```
C:\Users\ajlee>"C:\Program Files\ClamAV\clamscan.exe" -d C:\Users\ajlee\Desktop\finalproject\signatures.hdb C:\Users\ajlee\Desktop\finalproject\malware
C:\Users\ajlee\Desktop\finalproject\malware\e652d7c27f43ac16a2ec6ee5491166674882d458e49dd667f97b97992e912d1c.bin: Hash.Backdoor.UAE.UNOFFICIAL FOUND
C:\Users\ajlee\Desktop\finalproject\malware\LIMEWIRE_SONG2_REMIX_dump_.bin: OK

----------- SCAN SUMMARY -----------
Known viruses: 1
Engine version: 0.101.3
Scanned directories: 1
Scanned files: 2
Infected files: 1
Data scanned: 0.78 MB
Data read: 0.78 MB (ratio 1.00:1)
Time: 0.046 sec (0 m 0 s)

C:\Users\ajlee>"C:\Program Files\ClamAV\clamscan.exe" -d C:\Users\ajlee\Desktop\finalproject\signatures.mdb C:\Users\ajlee\Desktop\finalproject\malware
C:\Users\ajlee\Desktop\finalproject\malware\e652d7c27f43ac16a2ec6ee5491166674882d458e49dd667f97b97992e912d1c.bin: SecHash.Backdoor.UAE.UNOFFICIAL FOUND
C:\Users\ajlee\Desktop\finalproject\malware\LIMEWIRE_SONG2_REMIX_dump_.bin: OK

----------- SCAN SUMMARY -----------
Known viruses: 1
Engine version: 0.101.3
Scanned directories: 1
Scanned files: 2
Infected files: 1
Data scanned: 0.78 MB
Data read: 0.78 MB (ratio 1.00:1)
Time: 0.021 sec (0 m 0 s)

C:\Users\ajlee>"C:\Program Files\ClamAV\clamscan.exe" -d C:\Users\ajlee\Desktop\finalproject\signatures.ndb C:\Users\ajlee\Desktop\finalproject\malware
C:\Users\ajlee\Desktop\finalproject\malware\e652d7c27f43ac16a2ec6ee5491166674882d458e49dd667f97b97992e912d1c.bin: PE.Backdoor.UAE.UNOFFICIAL FOUND
C:\Users\ajlee\Desktop\finalproject\malware\LIMEWIRE_SONG2_REMIX_dump_.bin: OK

----------- SCAN SUMMARY -----------
Known viruses: 1
Engine version: 0.101.3
Scanned directories: 1
Scanned files: 2
Infected files: 1
Data scanned: 0.78 MB
Data read: 0.78 MB (ratio 1.00:1)
Time: 0.021 sec (0 m 0 s)

C:\Users\ajlee>"C:\Program Files\ClamAV\clamscan.exe" -d C:\Users\ajlee\Desktop\finalproject\signatures.ldb C:\Users\ajlee\Desktop\finalproject\malware
C:\Users\ajlee\Desktop\finalproject\malware\e652d7c27f43ac16a2ec6ee5491166674882d458e49dd667f97b97992e912d1c.bin: PE.Backdoor.UAE.A.UNOFFICIAL FOUND
C:\Users\ajlee\Desktop\finalproject\malware\LIMEWIRE_SONG2_REMIX_dump_.bin: PE.Backdoor.UAE.A.UNOFFICIAL FOUND

----------- SCAN SUMMARY -----------
Known viruses: 1
Engine version: 0.101.3
Scanned directories: 1
Scanned files: 2
Infected files: 2
Data scanned: 0.78 MB
Data read: 0.78 MB (ratio 1.00:1)
Time: 0.023 sec (0 m 0 s)
```