

# NetTraveler Spear-Phishing Email Targets Diplomat of Uzbekistan

By Vicky Ray, Robert Falcone

Published: 2016-01-21 · Archived: 2026-04-05 17:00:52 UTC

Unit 42 recently identified a targeted attack against an individual working for the Foreign Ministry of Uzbekistan in China. A spear-phishing email was sent to a diplomat of the Embassy of Uzbekistan who is likely based in Beijing, China. In this report, we'll review how the actors attempted to exploit CVE-2012-0158 to install the NetTraveler Trojan.

On December 12, 2015, a spear-phishing email was sent to a diplomat of the Embassy of Uzbekistan. The body and subject of the email suggests that the email was spoofed to look like it was sent by the Russian Foreign Ministry and the attachment may contain an official annual report on CHS (Council of Heads of Member States), who form the [SCO](#) (Shanghai Cooperation Organization).

*Filename: "2015.12.11\_сроку СНГ 2015 в Уфе.doc.doc" (translated to: "2015.12.11\_sroki CHS in 2015 Ufe.doc.doc")*

*Body: "С уважением, ДАТС МИД России" (translated to: "Yours faithfully, ACSD Russian Foreign Ministry")*

It is interesting to note the reference of Ufa in the file name, as the city of Ufa in Russia hosted the [SCO BRICS Summit](#) on July 9 and 10, 2015. SCO and BRICS (Brazil, Russia, India, China and South Africa) are intergovernmental international organizations focused on issues of regional security and economic cooperation.



*Figure 1 Leaders of member nations at the 2015 Summit in Ufa*

## TARGETING AND MALWARE ANALYSIS

Our analysis shows that actors attempted to exploit CVE-2012-0158 to install NetTraveler Trojan.

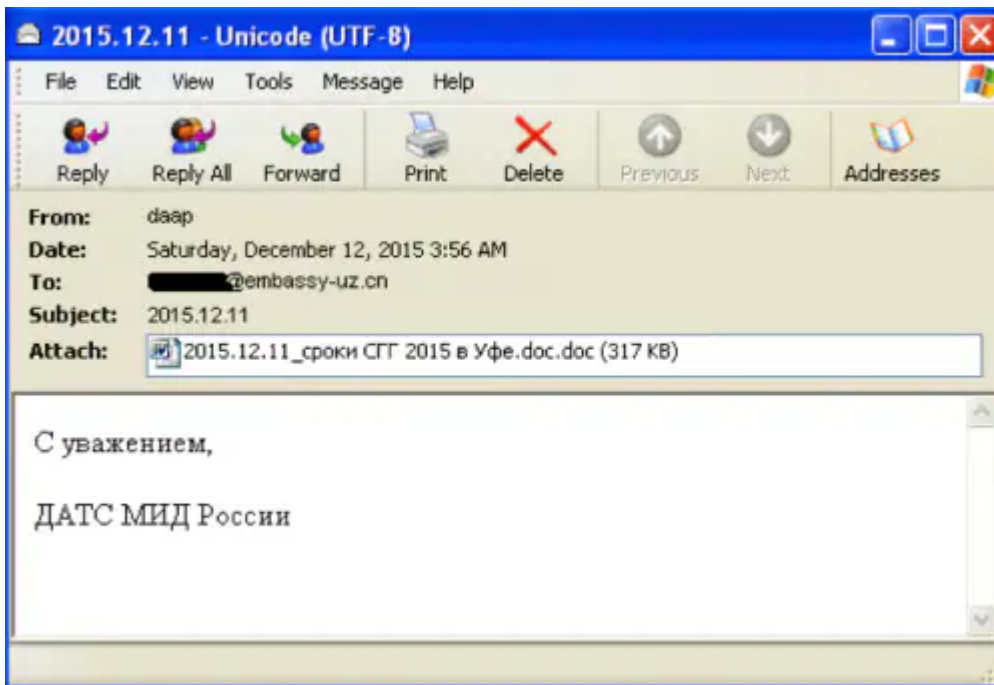


Figure 2 Email containing the malicious attachment

The malicious attachment “2015.12.11\_сроки СГГ 2015 в Уфе.doc.doc” is a malicious document created by the MNKit toolkit and exploits CVE-2012-0158.

Upon successful exploitation, the attachment will install the trojan known as NetTraveler using a [DLL side-loading](#) attack technique. The [NetTraveler](#) trojan has been known to be used in targeted cyber espionage attacks for [more than a decade](#) by nation state threat actors and continues to be used to target its victims and exfiltrate data.

The DLL side-loading attack technique has been gaining adoption within the cyber espionage realm by threat actors to bypass traditional security systems. Unit 42 also published a [blog](#) last year discussing an unrelated attack where the DLL side-loading technique was used.

Figure 3 illustrates the exploitation and the infection flow of the malware.

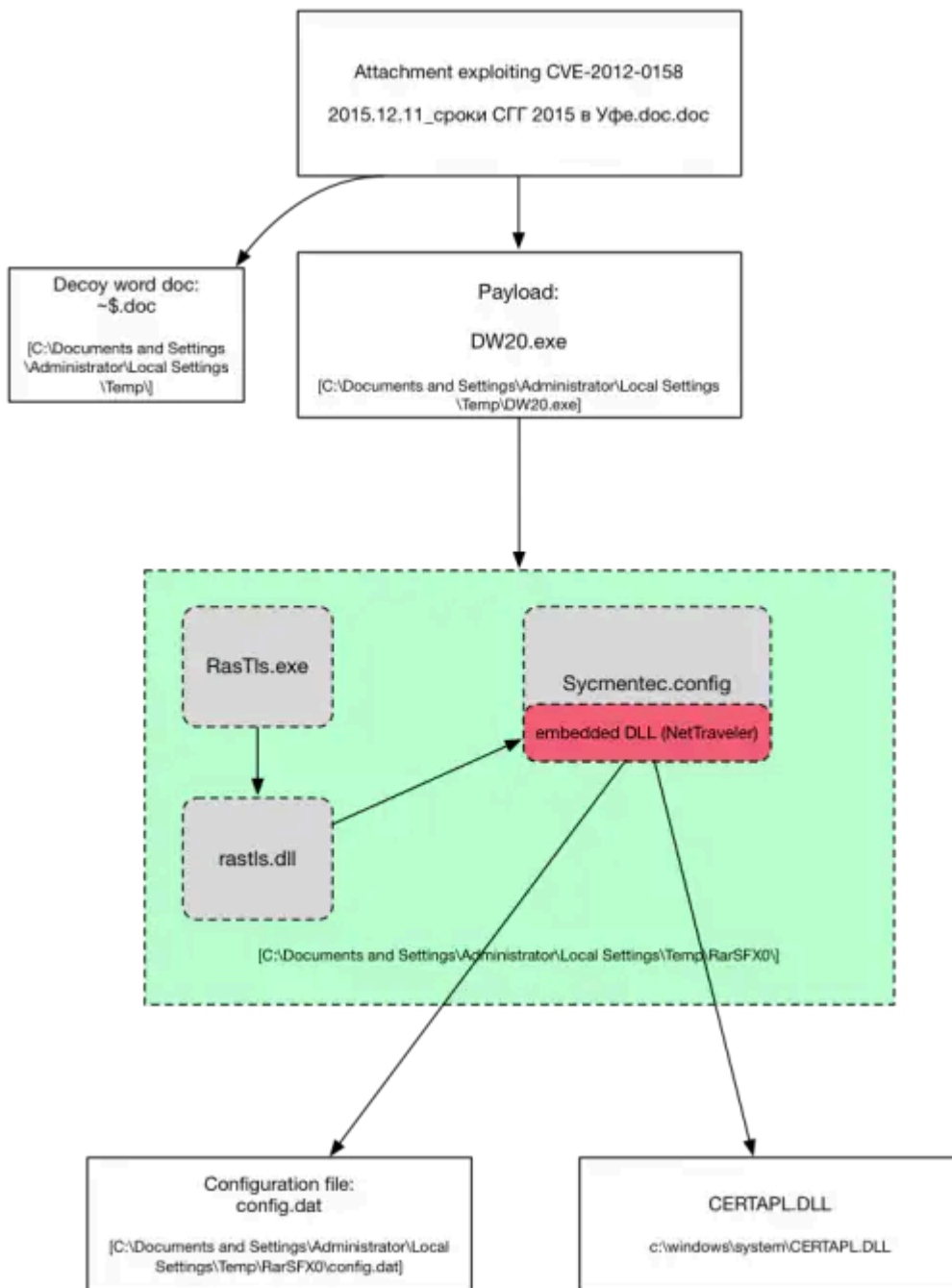


Figure 3 Overview of the infection flow

The document “2015.12.11\_сроки СГГ 2015 в Уфе.doc.doc” exploits CVE-2012-0158 to drop a decoy file “~\$.doc” and the actual payload “DW20.exe”. The decoy is a blank document with the meta data stripped.

The payload (DW20.exe) is a self-extracting (SFX) RAR archive that contains the following files:

*RasTls.exe*

*rasts.dll*

*Sycmentec.config*



Figure 4 The payload(DW20.exe) is a SFX RAR archive

The SFX RAR uses the following configuration to launch the embedded executable, which is a legitimate application created by Symantec that will side load the rastls.dll DLL:

*Setup=RasTls.exe*

*TempMode*

*Silent=1*

*Overwrite=1*

The figure below shows that the config file, 'Sycmentec.config' is encrypted.

The 'Sycmentec.config' file can be decrypted using a single byte XOR algorithm using '0x77' as a key.

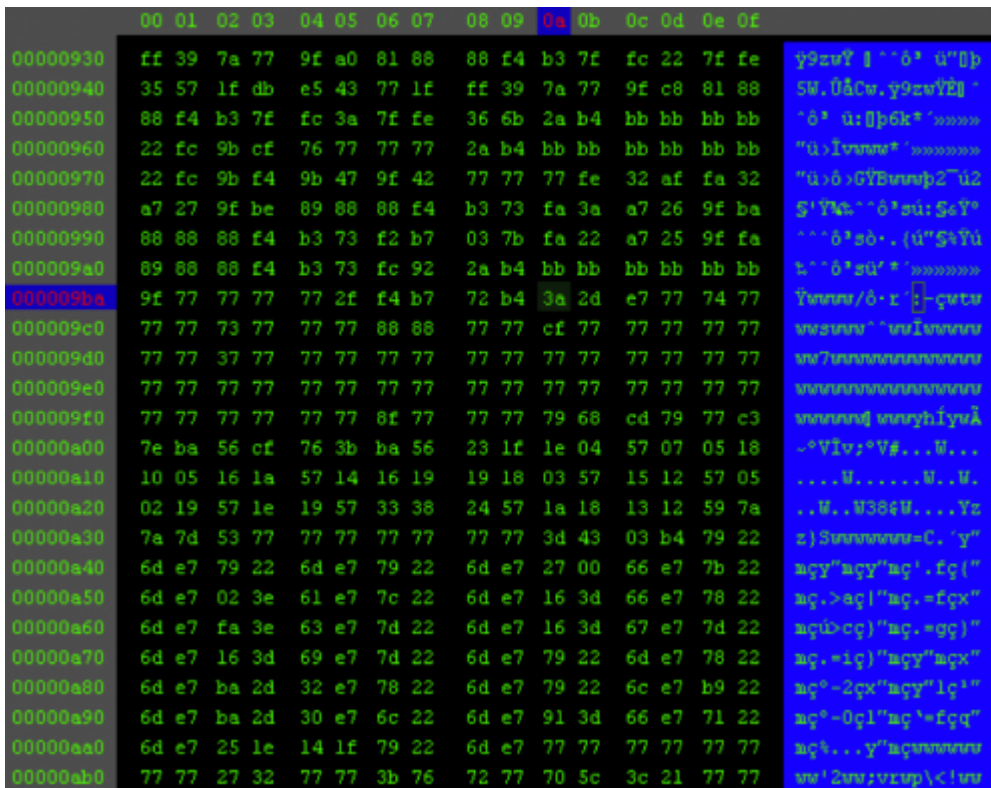


Figure 5 Encrypted 'Sycmentec.config' file

The 'rastls.dll' DLL will load and decrypt this file. The decrypted data starts with shellcode that is responsible for loading an embedded DLL and executing it.

Figure 6 shows the decrypted 'Sycmentec.config' file containing an embedded DLL.

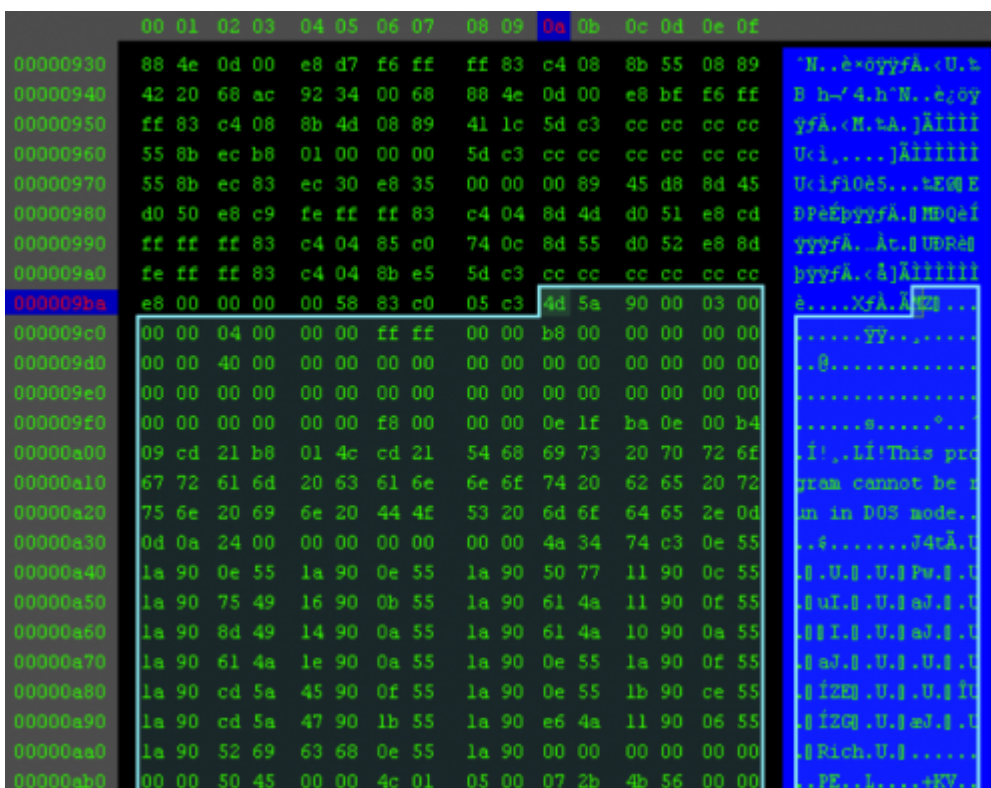


Figure 6 Decrypted 'Sycmentec.config' file contains an embedded DLL

The embedded DLL is the functional payload, which is a variant of the NetTraveler Trojan that has the following attributes:

<b>Size</b>	52736 bytes
<b>Type</b>	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
<b>Architecture</b>	32 Bits binary
<b>MD5</b>	3e3df4fe831d87d7f52f14933e464fc3
<b>SHA1</b>	cce65a0b67674a313091a947506ceb91d30605ad
<b>SHA256</b>	3b4e4d7a0b1185a45968d90ffe6346f4621116d14dbf88b5138040acc022c757
<b>ssdeep</b>	1536:jxKW1S8mWKFU7U9IYjhjXwVqTvS/G405:wCBmUw9lAhLWqW/G40
<b>imphash</b>	85ce31f87f06b02fec915d33d82958e8
<b>Date</b>	0x564B2B07 [Tue Nov 17 13:26:31 2015 UTC]
<b>CRC:(Claimed)</b>	0x0, (Actual): 0x19be0 [SUSPICIOUS]
<b>Packers</b>	Armadillo v1.xx - v2.xx
<b>Entry Point</b>	0x1000970b .text 1/5

Table 1 Attributes of the embedded DLL (NetTraveler)

The first execution of this NetTraveler Trojan starts off with an installation process. Like previous versions, this NetTraveler sample writes its configuration to a file, in this case the configuration is written to a file named "config.dat".

```

.text:1000430E      mov     dl, Default
.text:10004314      push   40h
.text:10004316      pop    ecx
.text:10004317      xor    eax, eax
.text:10004319      lea   edi, [ebp+var_110]
.text:1000431F      mov   [ebp+FileName], dl
.text:10004325      rep   stosd
.text:10004327      stosw
.text:10004329      stosb
.text:1000432A      push   40h
.text:1000432C      xor    eax, eax
.text:1000432E      pop    ecx
.text:1000432F      lea   edi, [ebp+var_4A7]
.text:10004335      mov   [ebp+var_4A8], dl
.text:1000433B      push  esi
.text:1000433C      rep   stosd
.text:1000433E      stosw
.text:10004340      stosb
.text:10004341      lea   eax, [ebp+FileName]
.text:10004347      push  offset aSConfig_dat ; "%s\\config.dat"
.text:1000434C      push  eax ; Dest
.text:1000434D      call  ebx ; sprintf
    
```

Figure 7 NetTraveler writes the configuration to 'config.dat' file

During execution, NetTraveler creates a mutex of 'YOYWOW!657', as shown in Figure 8 below to avoid running multiple instances of its code.

```
.text:1000401A      mov     edi, ds:Sleep
.text:10004020      push   4E20h          ; dwMilliseconds
.text:10004025      call   edi            ; Sleep
.text:10004027      push   offset Name    ; "YOYWOW!657"
.text:1000402C      xor     esi, esi
.text:1000402E      push   1              ; bInitialOwner
.text:10004030      push   esi            ; lpMutexAttributes
.text:10004031      call   ds:CreateMutexA
```

Figure 8 Mutex created for this NetTraveler payload

The code then enumerates the 'netsvcs' services, which are services that run within the process space of svchost.exe, specifically ignoring services named '6to4' and 'Ias' as these services have been used by other malware families.

When it finds another netsvcs service with a name not matching these two names, it will delete the file associated with the service and copy the 'rastls.dll' file to that folder using '<service name>ve.dll' as the filename as shown in Figure 9 below.

```

.text:10004696 loc_10004696: ; CODE XREF: sub_100044E3+297j
.text:10004696 mov     eax, [ebp+Str1]
.text:10004699 cmp     [eax], bl
.text:1000469B jz      loc_1000477F
.text:100046A1 lea    ecx, [ebp+Str2]
.text:100046A4 push   ecx ; Str2
.text:100046A5 push   eax ; Str1
.text:100046A6 call   strcmp
.text:100046A8 pop    ecx
.text:100046AC test   eax, eax
.text:100046AE pop    ecx
.text:100046AF jz      loc_1000476A
.text:100046B5 push   offset alias ; "ias"
.text:100046B8 push   [ebp+Str1] ; Str1
.text:100046BB call   strcmp
.text:100046C2 pop    ecx
.text:100046C3 test   eax, eax
.text:100046C5 pop    ecx
.text:100046C6 jz      loc_1000476A
.text:100046CC push   [ebp+Str1]
.text:100046CF lea    eax, [ebp+SubKey]
.text:100046D5 push   offset aSystemCurrentnc ; "SYSTEM\CurrentControlSet\Services\%s"
.text:100046D8 push   eax ; Dest
.text:100046DB call   ds:sprintf
.text:100046E1 add    esp, 0Ch
.text:100046E4 lea    eax, [ebp+hKey]
.text:100046E7 push   eax ; phkResult
.text:100046E8 push   1 ; sanDesired
.text:100046EA lea    eax, [ebp+SubKey]
.text:100046F0 push   ebx ; ulOptions
.text:100046F1 push   eax ; lpSubKey
.text:100046F2 push   8000002h ; hKey
.text:100046F7 call   ds:RegOpenKeyEx
.text:100046FD cmp    eax, ebx
.text:100046FF jnz    short loc_1000470C
.text:10004701 push   [ebp+hKey] ; hKey
.text:10004704 call   ds:RegCloseKey
.text:10004708 jnp    short loc_1000476A
;-----
.text:1000470C loc_1000470C: ; CODE XREF: sub_100044E3+210Tj
.text:1000470C push   100h ; Size
.text:10004711 push   ebx ; Ual
.text:10004712 push   esi ; Dst
.text:10004713 call   memset
.text:10004718 push   [ebp+Str1]
.text:1000471B push   edi
.text:1000471C push   offset aSvc_dll ; "%s\%s\svc.dll"
.text:10004721 push   esi ; LPSTR
.text:10004722 call   ds:wsprintfA
.text:10004728 add    esp, 1Ch
.text:1000472B push   esi ; lpFileName
.text:1000472C call   ds>DeleteFileA
.text:10004732 push   esi ; lpFileName
.text:10004733 call   ds:GetFileAttributesA
.text:10004739 cmp    eax, 0FFFFFFFh
.text:1000473C jnz    short loc_1000476A
.text:1000473E push   ebx ; lpPassword
.text:1000473F push   ebx ; lpServiceStartName
.text:10004740 push   ebx ; lpDependencies
.text:10004741 push   ebx ; lpdwTagId
.text:10004742 mov    eax, offset BinaryPathName ; "%SystemRoot%\System32\svchost.exe -k "
.text:10004747 push   ebx ; lpLoadOrderGroup
.text:10004748 push   eax ; lpBinaryPathName
.text:10004749 push   1 ; dwErrorControl
.text:1000474B push   2 ; dwStartType
.text:1000474D push   20h ; dwServiceType
.text:1000474F push   0x01FFh ; dwDesiredAccess
.text:10004754 push   [ebp+Str1] ; lpDisplayName
.text:10004757 push   [ebp+Str1] ; lpServiceName
.text:1000475A push   [ebp+hSChManager] ; hSChManager
.text:1000475D call   ds:CreateServiceA
.text:10004763 cmp    eax, ebx
.text:10004765 mov    [ebp+hSCObject], eax
.text:10004768 jnz    short loc_100047DA

```

Figure 9 Code enumerating 'netsvcs' services

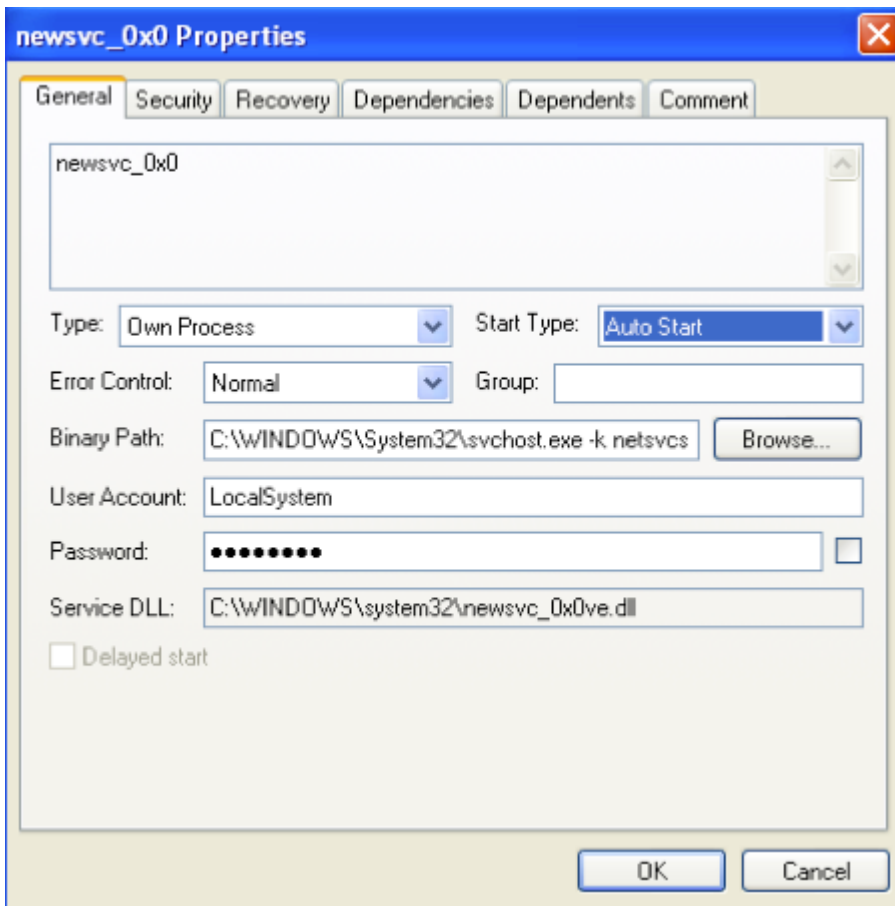


Figure 10 Renamed 'rastls.dll' DLL

The malware will then change the binary path of the service to point to this new filename and copies the "Sycmentec.config" file to the same folder and the 'config.dat' file to the following location:

`c:\windows\system\CERTAPL.DLL`

The NetTraveler payload relies on the 'rastls.dll' file to obtain its C2 server. At first glance, the NetTraveler payload appears as if it will use the following URL for its C2 server:

`http://192.168.3[.]201/downloader2013/asp/downloader.asp`

However, the NetTraveler payload reads the last '0xb0' bytes from the rastls.dll file and uses it to create the "config.dat" file that is later saved to "CERTAPL.DLL". This technique hides the true C2 server from researchers that do not have access to both the rastls.dll and Sycmentec.config files.

```

mov     ecx, var_40000000
push   offset aRastls_dll ; "\\rastls.dll"
push   esi
lea    eax, [ebp+Dest]
push   offset a88      ; "tala"
push   eax             ; Dest
call   ebx             ; sprintf
add    esp, 10h
lea    eax, [ebp+Dest]
push   edi             ; hTemplateFile
push   80h             ; dwFlagsAndAttributes
push   3               ; dwCreationDisposition
push   edi             ; lpSecurityAttributes
push   1               ; dwShareMode
push   80000000h      ; dwDesiredAccess
push   eax             ; lpFileName
call   ds:CreateFileA
cmp    eax, 0FFFFFFFh
mov    [ebp+hObject], eax
jns    short loc_100042D0

loc_100042D0:          ; dwMoveMethod
push   FILE_MOVE      ; lpDistanceToMoveHigh
push   edi             ; lpDistanceToMove
push   eax             ; hFile
call   ds:SetFilePointer
cmp    eax, 0FFFFFFFh
jns    short loc_100042F2

loc_100042F2:
lea    eax, [ebp+NumberOfBytesRead]
push   edi             ; lpOverlapped
push   eax             ; lpNumberOfBytesRead
push   000h           ; nNumberOfBytesToRead
push   offset aHttp192_168_3_201 ; "http://192.168.3.201/downloader2013/asp"
push   [ebp+hObject]   ; hFile
call   ds:ReadFile
test   eax, eax
js     short loc_100042E4

loc_100042E4:
mov    dl, Default
push   40h
pop    ecx
xor    eax, eax
lea    edi, [ebp+var_118]
mov    [ebp+FileName], dl
rep stq
stosq
stosb
push   40h
xor    eax, eax
pop    ecx
lea    edi, [ebp+var_4A7]
mov    [ebp+filename_neverused], dl
push   esi
rep stq
stosq
stosb
lea    eax, [ebp+FileName]
push   offset s_config_dat ; "%s\\config.dat"
push   eax             ; Dest
call   ebx             ; sprintf
    
```

Figure 11 Code snippet showing NetTraveler obtaining its configuration from rastls.dll.

The configuration file is structured as an ".ini" file as the Trojan uses GetPrivateProfileStringA to parse the contents. The configuration file has the following contents:

```

[OOOOOO]

U00P=r^?<80>}H?<88><89><8A>B<8B><85>|<86><87><89><91><8B><90><92><88>N<84><91>
<90>S<94><96><9B><8C><8E><9E>Z<95><9B><92><94><A8>_<93><A6><A4>

K00P=XMLNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop

P00D=5

F00G=True

MM1=0

MM6=1
    
```

Unit 42 analyzed the sample and found the following configuration fields that could appear in the CERTAPL.DLL configuration file and a brief description of each field:

U00P = C2 URL
K00P = Key for DES
P00D = Sleep interval in minutes
F00G = Boolean to determine if sample should use proxy to communicate with C2 server
MM1 = 0 or 1 if proxy is configured or not.
MM3 = Port for configured proxy
MM4 = Username for configured proxy
MM5 = Password for configured proxy
MM6 = 1 if Trojan is installed correctly

The "U00P" and "K00P" values are decrypted using a simple algorithm that subtracts the index and then subtracts ten from each character, which is depicted in the following:

```
def subtraction_algo(ct):  
    out = ""  
    i = 0  
    for e in ct:  
        out += chr(ord(e)-i-10)  
        i += 1  
    return out
```

These two fields decrypt to the following, the U00P value being the C2 URL and the K00P value being the basis for an encryption key for the DES algorithm:

*U00P: http://www.voennovosti.com/optdet/index.asp (decrypted)*

*K00P: NAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAM (decrypted)*

The C2 server will respond to requests issued by the Trojan with commands to carry out activities on the compromised system. We analyzed the code within NetTraveler that handles commands issued by the C2 server and found four available commands that are listed in Table 2.

Command	Description
<Unique System ID>:UNINSTALL	Deletes %APPDATA%\cert2013.dat and %STARTUP%\consent.lnk and exits the process. This attempts to uninstall the Trojan, but will not work as the filenames are not used by this version of NetTraveler
<Unique System ID>:RUN_REBOOT	Reboots the system
<Unique System ID>:RUN_STARTUP	Downloads a file to %TEMP%\Temp.bmp and copies it to the startup folder
<Unique System ID>:RUN_DIRECT	Download a file to %TEMP%\tmp.bmp and execute it

Table 2 Commands available within NetTraveler and a description of their functionality

## INFRASTRUCTURE

At the time of analysis, the domain voennovosti[.]com was resolving to IP ‘98.126.38[.]107’, which is hosted by Krypt Technologies. A [report](#) published by Kaspersky Labs in 2011 on NetTraveler also mentions the C2 servers were being hosted by Krypt Technologies. This web hosting service provider continues to be the hosting provider of choice for the threat actors behind NetTraveler.

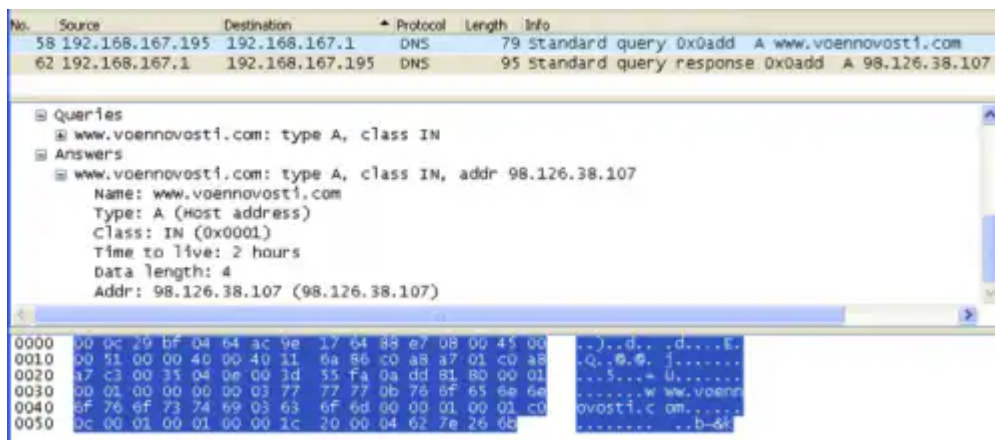


Figure 12 DNS query for voennovosti[.]com resolves to ‘98.126.38.107’

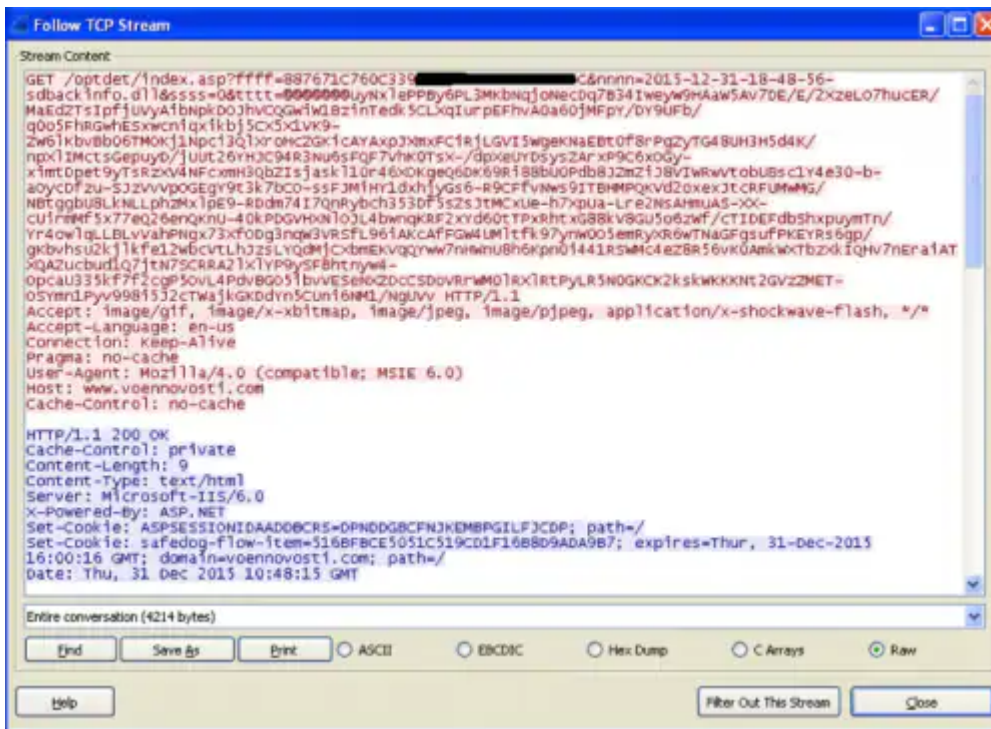


Figure 13 Encoded network communications

## CONCLUSION

NetTraveler has been used to target diplomats, embassies and government institutions for over a decade, and remains the tool of choice by the adversaries behind these cyber espionage campaigns. The use of NetTraveler for such a long period of time shows its effectiveness and success by the adversaries in targeting their victims with impunity.

As seen in this case, the threat actors continue to evolve and employ new techniques within their modus operandi, like ‘DLL side-loading’ to install malware. It is likely that the use of ‘DLL side loading’ attack technique will increase due to it’s effectiveness to bypass traditional security systems.

It is essential to raise awareness on such attacks to better protect organizations from adversaries who maybe backed by nation states.

[WildFire](#) correctly classifies NetTraveler as malicious. [AutoFocus](#) tags are created to identify NetTraveler samples and respective IOCs are added to Palo Alto Networks [Threat Prevention](#).

## INDICATORS

SHA256 Hash	File Name
3f4fcd99775b83bc88d30ca99f5c70c1dd8b96d970dbfd5a846b46c6ea3e534	2015.12.11_сроки СГГ 2015 в Уфе.doc.doc
001fff6c09497f56532e83e998aaa80690a668883b6655129d408dd098bd1b4b	DW20.exe

74db11900499aa74be9e62d51889e7611eb8161cd141b9379e05eeca9d7175c9	rastls.dll
8f6af103bf7e3201045ce6c2af41f7a17ef671f33f297d36d2aab8640d00b0f0	Sycmentec.config
495bb9c680f114b255f92448e784563e4fd34ad19cf616cc537bec6245931b7e	config.dat
41650cb6b4ae9f06c92628208d024845026c19af1ab3916c99c80c6457bd4fa9	CERTAPL.DLL
3b4e4d7a0b1185a45968d90ffe6346f4621116d14dbf88b5138040acc022c757	(NetTraveler DLL payload)

## Command and Control

voennovosti[.]com

98.126.38[.]107

## REFERENCES

- <https://securelist.com/blog/research/35936/nettraveler-is-running-red-star-apt-attacks-compromise-high-profile-victims/>
- <https://www.fireeye.com/blog/threat-research/2014/04/dll-side-loading-another-blind-spot-for-anti-virus.html>
- <http://blog.paloaltonetworks.com/2015/05/plugx-uses-legitimate-samsung-application-for-dll-side-loading/>
- <http://indianexpress.com/article/business/business-others/10-years-on-sco-decides-to-induct-india-as-full-member/>
- [https://en.wikipedia.org/wiki/Shanghai\\_Cooperation\\_Organisation](https://en.wikipedia.org/wiki/Shanghai_Cooperation_Organisation)
- <http://ufa2015.com/>

---

Source: <https://unit42.paloaltonetworks.com/nettraveler-spear-phishing-email-targets-diplomat-of-uzbekistan/>