

Luminosity RAT - Re-purposed

By malwarenailed

Published: 2016-07-30 · Archived: 2026-04-05 15:29:04 UTC

So I came across a sample which was sent inside a .7z file and strangely was detected by a file filter and not any spam or antivirus filter. The file was interestingly named as EmiratedNBD.exe, which indicates that this attack is targeted to the region as Emirates NBD is one of the biggest financial entities in the GCC region. The sample was not identified by any antivirus and also unknown to Virustotal till today.

MD5: 9b2da7bfb9dedaba7e4d14d623081d7f

SHA1: cfa3ce2a7743181775870d00f4f418efdd737a31

Moreover, the file seem encrypted as there were not many strings found and only a few libraries were visible in the import directory. The sample is coded in .NET C#.

I have performed basic static and dynamic analysis of the sample and initial findings strongly indicates that the sample is an encrypted payload of Luminosity RAT which is infecting the endpoint and performing a C2 communication with its server (C2 ip address: 204.45.103.37). Till date there is not much OSTI available on this ip address.

The sample when starts as a process, seems to start a child process in suspended mode and writes its memory space with the decrypted payload. The decrypted payload is responsible for all Botnet communication with the C2 server. The original parent process terminates after spawning the child process. I was able to extract many useful strings from the memory of the running child process. I also noticed another artifact left by the sample "explorer.exe" in the temp folder and placed in the startup as an IE shortcut.

The sample bot beaconing is periodic and seems to be utilizing a custom network protocol whereby destination port is 19881 and the information sent to the C2 server comprise of information on analysis tools running and the GUI processes which are running, the domain and user information, OS version, something "True", Antivirus running, a hash of some sort, current date, an "N" and lastly 8=_8 in the end. It is noticeable that the request sent across is always preceded by a ping to the ip address. The communication from the client starts with "CONNECT" and each message is preceded by "=P4CK3T". The response from the server is "ACT=P4CK3T=8=_8".

FireEye detects the communication protocol as Trojan Luminosity Link.

Attempting to decompile the sample with .NET Reflector and JetBrains dotPeek yields no results obviously as the sample is encrypted. Debugging with DILE also yields no results as the execution never stops at the entry point mentioned in the optional headers of the PE.

I had to resort on ILSPY to decompile to CIL opcode and see what is going on. Currently, I am also searching for some other ways to successfully debug the sample. The decompiled opcode shows the entry point of the code is the module lohnfraz("lohnfraz.NN v5.5.5").

The sample looks for certain config and ini files as well in the same directory as the sample, indicating that it was accompanied by some configuration and I am guessing that the threat actors have re-purposed the Luminosity RAT, encrypted the client payload and delivered through inside a weaponized .7z archive, aimed to target the GCC region.

Event Generated Time	Threat Type	Threat Name	Threat Target File Path
7/25/16 1:05:43 AM	File Filter	FileFilter	Emirates NBD.7z*Emir atesNBD.exe

The sample detection by file filter

```
Received from [redacted] by [redacted] with Microsoft SMTP Server id 14.3.248.2; Mon, 25 Jul 2016 01:05:40 +0400
[redacted]
[redacted]
Received from gnssayer.com ([68.169.48.240]) by [redacted] with ESMTP; 25 Jul 2016 01:05:34 +0400
Received from User (37-103-45-204 reverse-dns:server [204.45.108.37] [maybe forged]) ([authenticated:bits=0]) by cloudzones.org ([8.14.4/8.14.4]) with ESMTP id u60E6fCX001312; Sun, 24 Jul 2016 14:06:17 GMT
Message-ID: <201607241406.u60E6fCX001312@cloudzones.org>
From: EmiratesNBD Bank <accounts@emiratesnbd.com>
Subject: *****SPAM***** Your payment copy advice from Emirates NBD Bank/subsidiary
Date: Sun, 24 Jul 2016 07:06:33 -0700
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="====_NextPart_000_010A_01C2A9A6_1F053A20"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2600.0000
To: Undisclosed recipients;
Return-Path: accounts@emiratesnbd.com
X-MS-Exchange-Organization-OriginalArrivalTime: 24 Jul 2016 21:05:40.6522 (UTC)
[redacted]
[redacted]
[redacted]
```

The email header



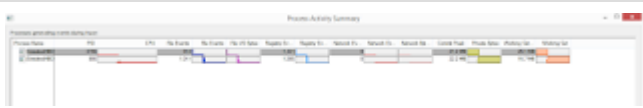
The delivery



Contained communication



Process spawning the child process



Proc Mon Activity output



Looking for config file



Looking for INI configuration

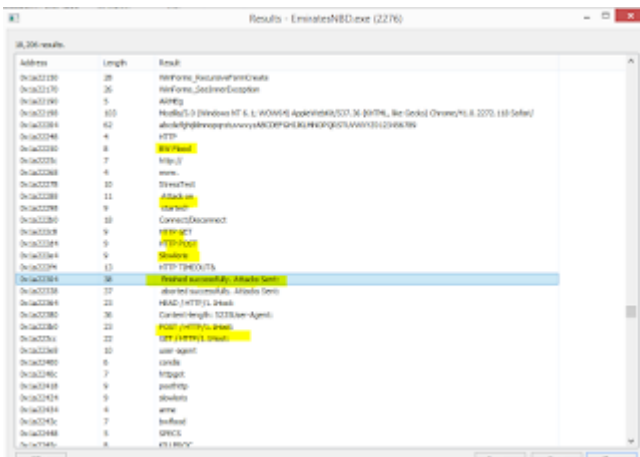


Malware can check if there are any unusual entries under the IFEO key(s) as a way of determining if it has landed on an analysts machine and change its behavior accordingly.



This file is created in the temp folder with the MD5 hash value of the original binary. I am guessing this is an integrity check being done by the sample of itself

Some of the interesting strings extracted from the decrypted child process memory space:

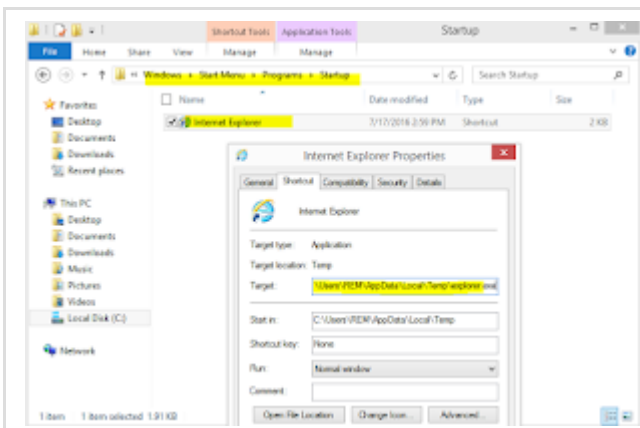


0x67f570 (60): C:\Windows\system32\ws2_32.dll

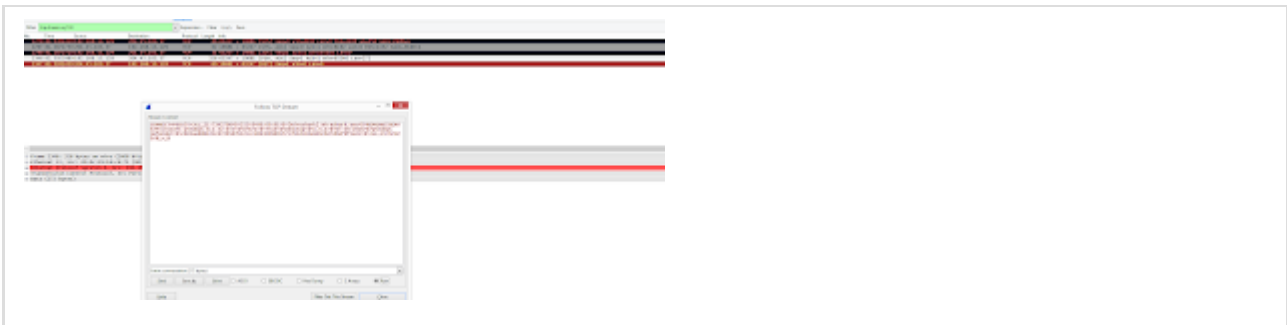
0x67f570 (60): C:\Windows\SYSTEM32\bcrypt.dll

```

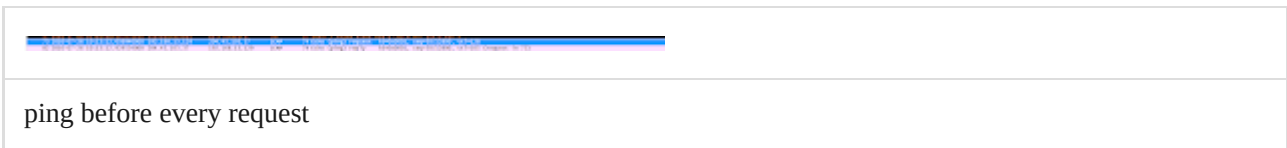
13760 0x1a2230c (22): GET / HTTP/1.1
13761 Host:
13762 0x1a2230c (10): user-agent
13763 0x1a22400 (4): cookie
13764 0x1a22400 (7): DSG0005
13765 0x1a22410 (9): postscript
13766 0x1a22424 (9): showdoc
13767 0x1a22434 (4): GMS
13768 0x1a2243c (7): bwload
13769 0x1a22448 (5): SECS
13770 0x1a2245c (8): WSIPROG
13771 0x1a22460 (7): NOTKILL
13772 0x1a2247c (7): ADDHOST
13773 0x1a22490 (4): NOTIFY
13774 0x1a2249c (4): FWFILE
13775 0x1a224a8 (4): SCLEAN
13776 0x1a224b4 (4): DELREG
13777 0x1a224c0 (4): Once
13778 0x1a224d8 (5): HOSTS
13779 0x1a224e4 (7): SETHOST
13780 0x1a224f0 (4): CLIP
13781 0x1a22508 (7): SETCLIP
13782 0x1a22514 (5): SHELL
13783 0x1a22520 (10): STARTSHELL
13784 0x1a22530 (7): STOPFILE
13785 0x1a2253c (9): STOPSHELL
13786 0x1a2254c (12): RESTARTSHELL
13787 0x1a22564 (5): SWLIB
13788 0x1a22570 (4): RWND
13789 0x1a22576 (4): CWND
13790 0x1a22588 (4): LDIR
13791 0x1a22600 (15): ClientHandler
13792 0x1a22634 (11): UserProfile
13793 0x1a22644 (4): FLOO
13794 0x1a22660 (17): Shell.Application
13795 0x1a22664 (9): NameSpace
13796 0x1a22674 (8): CopyHere
13797 0x1a22680 (5): Item
13798 0x1a22684 (5): FMSHEL
    
```



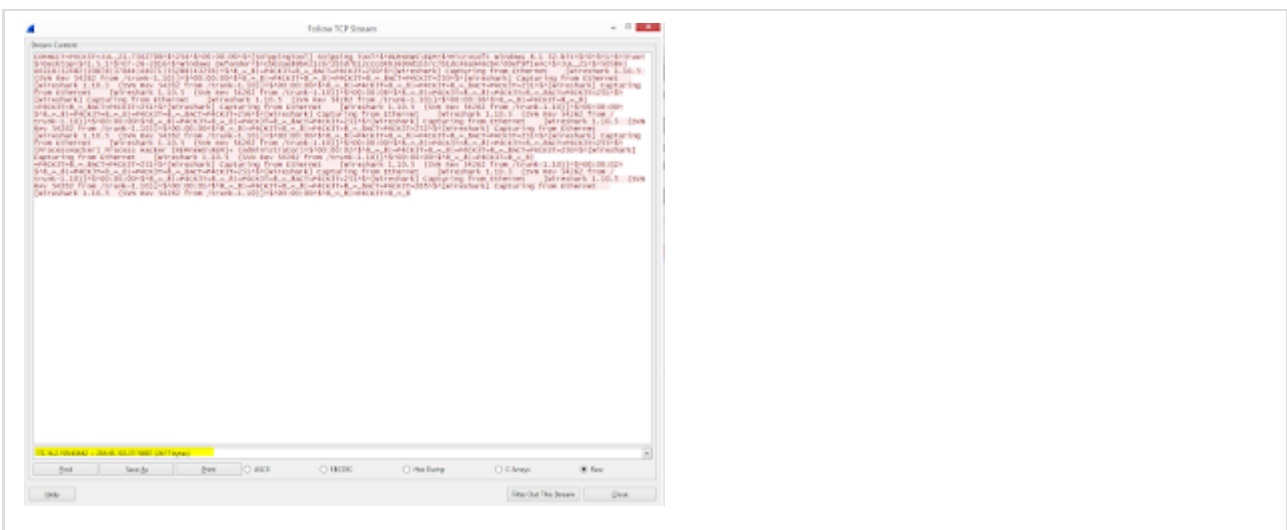
The explorer.exe artifact left by the sample



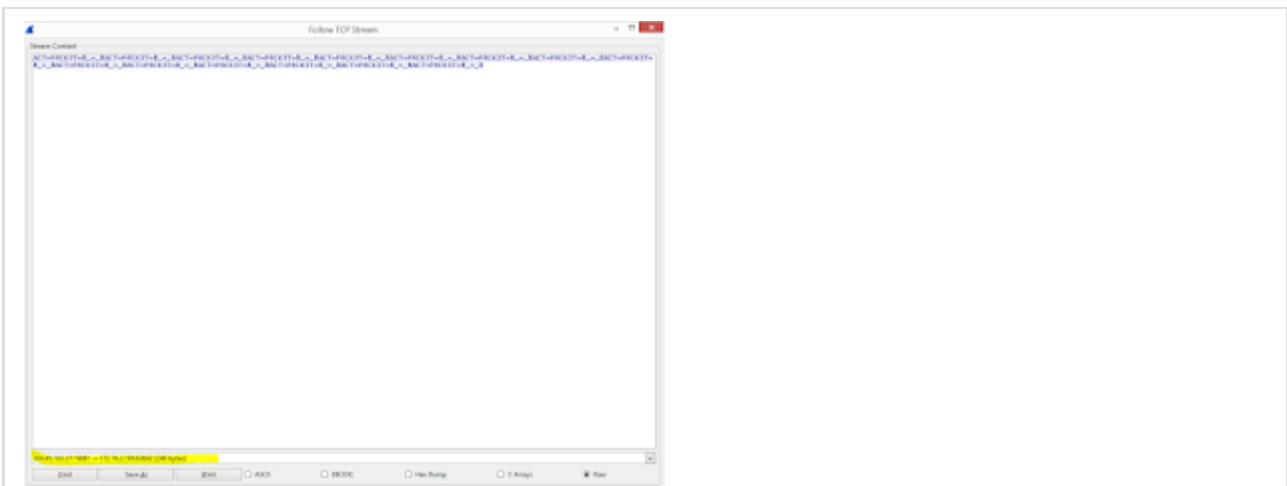
One of client message (hello to C2 server)



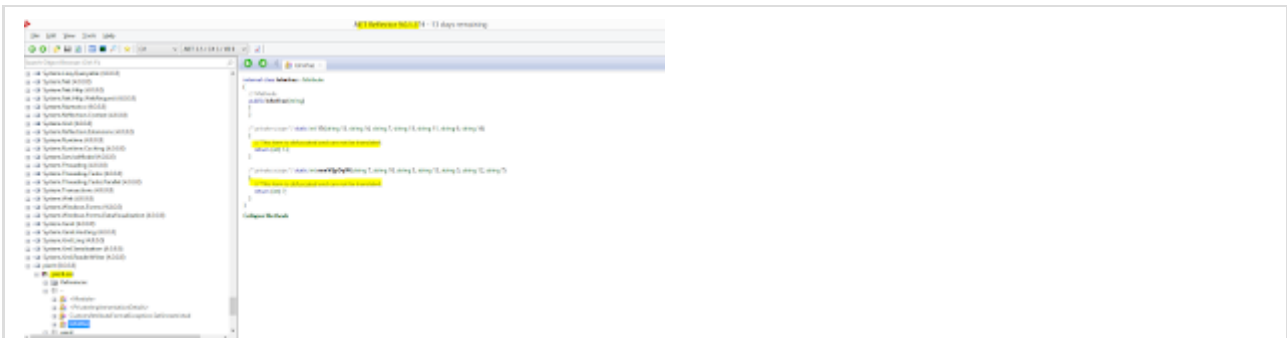
ping before every request



Client communications



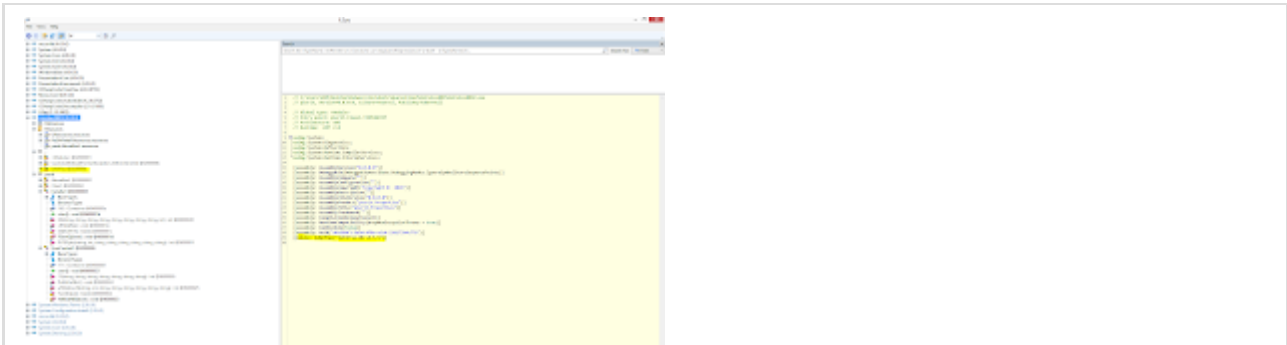
Server replies



De-compile failure



De-compile failure



Reversing the CIL opcode