

A short journey into DarkVNC attack chain

reaqta.com/2017/11/short-journey-darkvnc/



During an analysis of different remote desktop trojans we came across an interesting attack-chain which leverages an RTF that exploits **CVE-2017-8759** to deliver **DarkVNC**, a malicious version of the well-known **VNC**, designed to silently remote-control a victim.

DarkVNC Attack Chain

The DarkVNC chain as reconstructed by ReaQta-Hive can be seen below:



After opening the RTF document, one of the first processes to start is `csc.exe` which is a Command-Line build tool used to invoke the C# compiler, even though `csc.exe` is a perfectly legit, tool it can be abused for malicious purposes. The first step is to inspect the command-line of `csc.exe` to discover what is going to be compiled on-the-fly:

The Injector

As previously stated, *result.exe* acts as a loader, its goal is to decrypt and inject the malicious DLL that contains DarkVNC. From a static analysis point of view we have the following characteristics:

SHA256: 1D6F4CAC33FFF1B744DCE13BDF003B15D8EABCE53B0578E3B4BDBC5CBF001D78

SHA1: 2BB1BE823ED569EF3DAC008B2FEC4A8D04E46922

MD5: 22E2B492108F9D5517EE52C37912F24D

File size: 551.50 KB (564736 bytes)

File name: result.exe

File type: Win32 EXE

The executable does not have a *Version Information* and from an initial inspection it's encrypted with some private PE cryptor. We will skip the detailed analysis of the packer and subsequent unpacking steps because we are more interested in the overall behavior. *result.exe* uses several layers of encryption but does not implement complex anti-reverse engineering countermeasures, so the fastest way track the core behavior by setting a breakpoint on *VirtualAlloc()* and following the various layers.

dword ptr [ebx+47C03C] = [0047C03C - result..&virtualAlloc] = &kernel32.virtualAlloc

.text:00403C4C result..ecx_1:324C #204C

Address	hex	ASCII
002D0000	4D 1A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00yy..
002D0010	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002D0020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002D0030	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002D0040	04 1F 8A 04 00 84 09 C0 21 88 01 4C CB 31 54 68	...?..I..Lith
002D0050	69 73 20 76 72 6F 87 72 61 6B 20 62 61 66 66 6F	is program canno
002D0060	74 20 62 65 20 72 75 68 20 69 68 20 44 4F 53 20	t: be ran in oos
002D0070	6D 6F 64 65 20 00 0D 0A 24 00 00 00 00 00 00	mode....\$......
002D0080	58 45 00 00 4C 01 01 00 15 82 0D 54 00 00 00	PE..t.....ly....
002D0090	00 00 00 00 00 00 01 01 08 01 02 05 00 92 07 00\$......
002D00A0	00 02 01 00 00 00 00 00 F9 24 00 00 00 20 00 00k.....

Execution will jump from layer to layer until we reach the last one where it's possible to get the most important aspects of the injector.

Obtains the *ComputerName* and an additional identifier in order to assemble the string that will identify the victim's endpoint, the final string will be: **(COMPUTER_NAME)_ADDITIONAL_ID-DARKVNC**. Immediately after, the VNC Server is started. We will not go through the analysis of the whole module for the sake of brevity, but from the inspection of strings we can speed-up the initial assessment.

```

75158DAE 61 79 4E 6F 74 69 66 79 57 6E 64 00 00 00 53 79 ayHosIfyMhd Sy
75158DBE 79 50 61 67 65 72 00 00 00 00 56 69 73 75 61 6C sPager Visual
75158DCE 45 66 66 65 63 74 73 00 00 00 2A 00 00 00 29 68 Effects # ah
75158DDE 76 6E 63 00 00 00 5C 00 2A 00 00 00 00 00 2E 00 unc \ #
75158DEE 00 00 2E 00 2E 00 00 00 00 00 70 00 75 00 6C 00
75158DFE 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

The string **#hvnc** is pretty indicative, this core shares many similarities with HVNC (HiddenVNC) a well-known Remote-Control Module whose source-code can be found in the [carberp leak](#). This module shares with it a large amount of similarities like:

Hidden VNC capabilities: The module will create a *new Window Desktop to keep hidden the malicious VNC instance*. This technique is usually adopted to bypass anti-fraud engines on personal banking websites by impersonating the victim's computer and logging in with stolen credentials without raising alerts on the bank's side. Here's a quick representation of the above behavior taken from ReaQta-Hive's process-tree point of view:



We have a new *explorer.exe* instance and one of the child processes is Chrome!
 While there are also some basic differences between DarkVNC and HVNC, one of the most interesting is represented by the following:

```
SetEnvironmentVariable("MOZ_DISABLE_CONTENT_SANDBOX", "1")
```

According to the [documentation](#) **MOZ_DISABLE_CONTENT_SANDBOX** disables content process sandboxing.

The threat from a higher perspective

So far we have identified the following DarkVNC samples:

Collected samples:

1. deb02b28605a2b9c80b25c5fa1fa43ac8c71b10961f7517c1a0394531d3b0b40
2. 9a57cefbfcdf1b18cc31a2784a2ed3e0e11dd4a3c4608b1243b4141a475b182f
3. a67e96b01520183babfae285b5d692b5b3dda7edff7378b281ace7fd381d3c93
4. e0a73dd11f0f2c41859bf01cf8a5b7a2a9946303d6e7898f696037323d038f56
5. Delivered via Terror EK: <http://www.malware-traffic-analysis.net/2017/10/17/index.html>

Hashes of the sample analyzed in this post:

1. RTF: 7a641c8fa1b7a428bfb66d235064407ab56d119411fbaca6268c8e69696e6729
2. result.exe: 1d6f4cac33fff1b744dce13bdf003b15d8eabce53b0578e3b4bdb5cbf001d78

Detection & Protection

Visibility over the endpoints is essential to quickly detect new threats as they're deployed by the attackers. Real-time behavioral analysis creates a window of opportunity to detect behaviors that are unusual, running VNC or Teamviewer is not a malicious activity by itself but those same tools can be abused to get control over an endpoint. Being capable of detecting such anomalies allows for a timely analysis and response before the severity of the incident escalates.

Check out [ReaQta-Hive](#) to understand how an *Endpoint Threat Response* platform can help your organization to secure the infrastructure from threats like the one just analyzed, track incidents and respond in real-time. Anomalous behaviors can be hard to understand manually and the help offered by the algorithms greatly increase the chances of detection and consequently the reaction time.