A Deep Dive Into RUNDLL32.EXE

nasbench.medium.com/a-deep-dive-into-rundll32-exe-642344b41e90

October 10, 2020

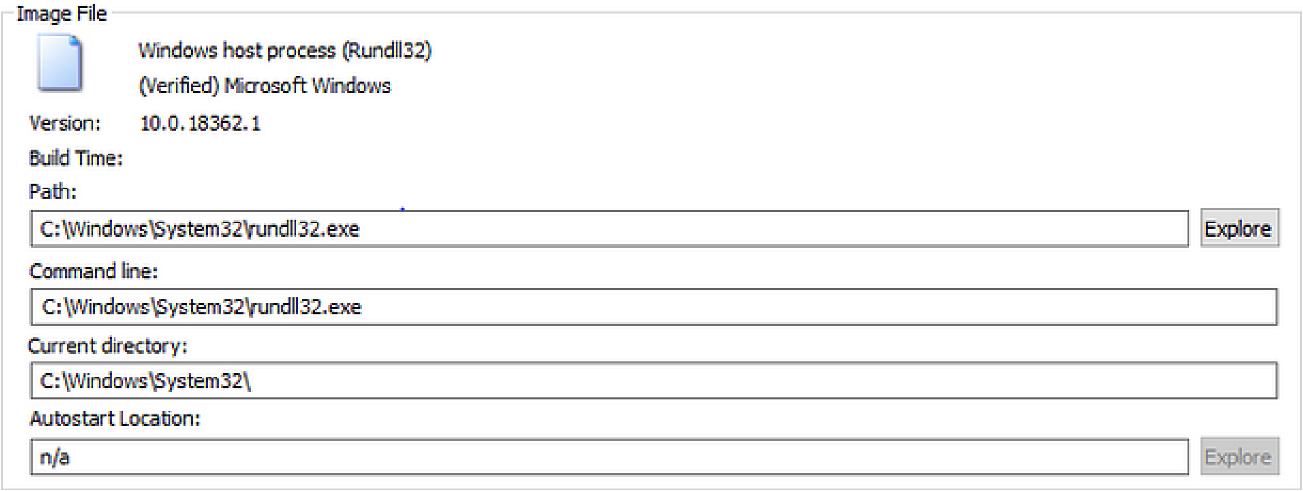


Image File

Windows host process (Rundll32)
(Verified) Microsoft Windows

Version: 10.0.18362.1
Build Time:
Path:
C:\Windows\System32\rundll32.exe
Command line:
C:\Windows\System32\rundll32.exe
Current directory:
C:\Windows\System32\
Autostart Location:
n/a

Rundll32

When threat hunting malware one of the key skills to have is an understanding of the platform and the OS. To make the distinction between the good and the bad one has to know what’s good first.

On windows this can be a little tricky to achieve because of the complexity of the OS (after all it’s a 30+ years’ operating system).

Knowing this fact, malware authors write their malware to mimic normal windows processes. So you’ll see malware disguising itself as an “svchost.exe”, “rundll32.exe” or “lsass.exe” process, exploiting the fact that the majority of people using windows don’t know how these system processes behave in normal conditions.

Last time we’ve talked about the “svchost.exe” process and its command line options.

Demystifying the “SVCHOST.EXE” Process and Its Command Line Options

Understanding the “svchost.exe” process and its command line options

medium.com

Today however we’ll be taking a look at “rundll32.exe” and understanding a little bit more about it.

RUNDLL32.EXE

As the name suggest, the “rundll32.exe” executable is used to “RUN DLL’s” or Dynamic Link Libraries (Below is the definition of a DLL from MSDN).

> A *dynamic-link library* (DLL) is a module that contains functions and data that can be used by another module (application or DLL) — MSDN

The most basic syntax for using “rundll32.exe” is the following.

The “rundll32.exe” executable can be a child or a parent process, it all depend on the context of the execution. And to determine if an instance of “rundll32.exe” is malicious or not we need to take a look at a couple of things. First is the path from which its being launched and second is its command line.

The valid “RUNDLL32.EXE” process is always located at:

As for the command line of a “rundll32.exe” instance it all depends on what’s being launched whether be it a CPL file, a DLL install…etc.

For this let’s take a look at a couple of examples.

Running a DLL

In its basic form, “rundll32.exe” will just execute a DLL, so the first thing to check when seeing an instance of **“rundll32.exe”** is the legitimacy of the DLL being called.

Always check the location from where the DLL is called, for example kernel32.dll being called from %temp% is obviously malicious. And as a side note always check the hash on sites like VT.

SHELL32.DLL — “OpenAs_RunDLL”

“rundll32.exe” can also execute specific functions in DLL’s. For example, when selecting a file and performing a right click on it, a context menu will be shown that offers multiple options. One of the options is the **“OpenWith”** option. Once selected a pop-up will appear that’ll let’s select from a set of applications on the system.

Behind the scene this is actually launching the “rundll32.exe” utility with the “shell32.dll” and the “OpenAs_RunDLL” function.

This behavior of calling specific functions in a DLL is very common and it can be tricky to know all of them in advance. Below is a list containing a batch of “rundll32.exe” calls and their meaning.

SHELL32.DLL — “Control_RunDLL”, “Control_RunDLLAsUser” and Control Panel Applets

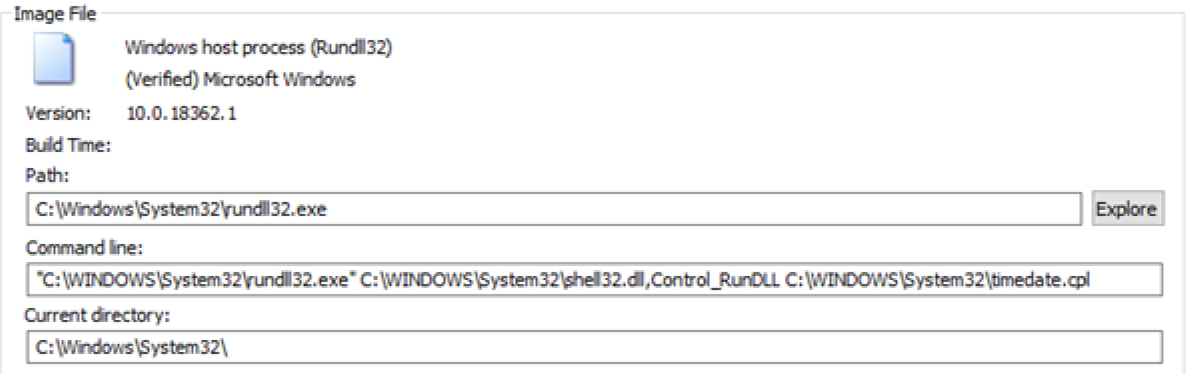


Image File

Windows host process (Rundll32)
(Verified) Microsoft Windows

Version: 10.0.18362.1
Build Time:
Path:
C:\Windows\System32\rundll32.exe

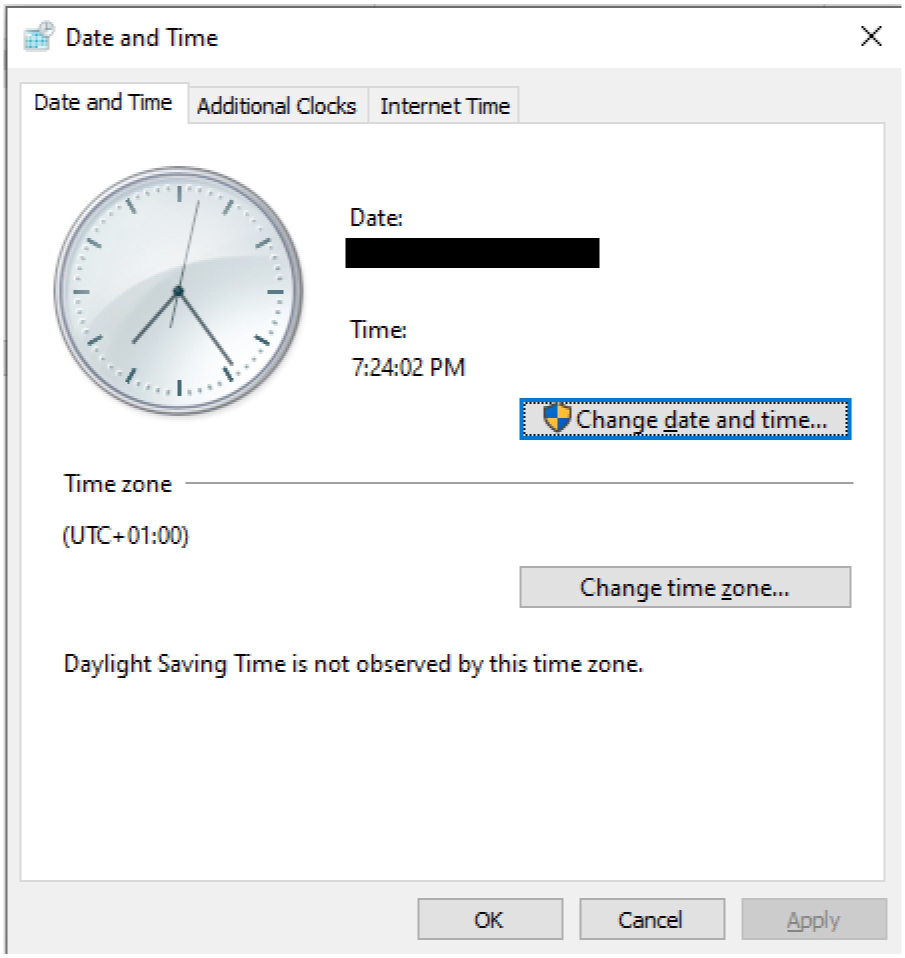
Command line:

```
"C:\WINDOWS\System32\rundll32.exe" C:\WINDOWS\System32\shell32.dll,Control_RunDLL C:\WINDOWS\System32\timedate.cpl
```

Current directory:
C:\Windows\System32\

Another common function we’ll see used with the **“shell32.dll”** is **“Control_RunDLL” / “Control_RunDLLAsUser”**. These two are used to run “.CPL” files or control panel items.

For example, when we want to change the Date and Time of the computer we launch the applet from the control panel.



Date:
[redacted]

Time:
7:24:02 PM

Time zone

(UTC+01:00)

Daylight Saving Time is not observed by this time zone.

Behind the scene, windows launched a "rundll32.exe" instance with the following command line.

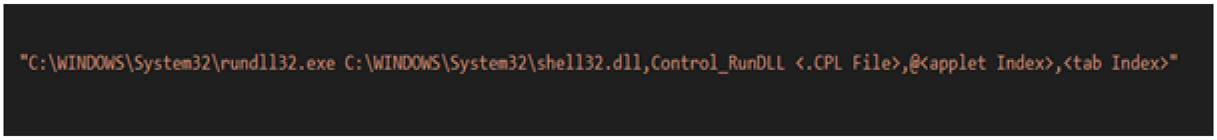
In addition to verifying the legitimacy of a DLL. When using the **"Control_RunDLL"** / **"Control_RunDLLAsUser"** functions, you should always check the legitimacy of a ".CPL" file.

Control Panel Items (.CPL)

CPL or Control Panel Items are programs that represent a functionality provided by the control panel or in other terms, they are DLL's that exports the **CPlApplet** Function.

A ".CPL" file can contain multiple applets that can be referred to by an applet index and each applet can contain multiple tabs that can be referred to by a tab index.

We can access and request this information via the "rundll32.exe" utility as follow.



```
"C:\WINDOWS\System32\rundll32.exe C:\WINDOWS\System32\shell32.dll,Control_RunDLL <.CPL File>,@<applet Index>,<tab Index>"
```

For example, the **“main.cpl”** file in the System32 folder contains two applets. The **“Mouse”** and **“Keyboard”** properties. If we want to access the mouse properties and change the pointer, we’ll do it like this.

As you can see, one can easily replace the “main.cpl” file with a malicious version and come by unnoticed to the untrained eye. In fact, that’s what malware authors have been doing to infect users.

In a normal case scenario, the parent process of a **“rundll32.exe”** instance with the **“Control_RunDLL”** function should be **“explorer.exe”** or **“control.exe”**

Other processes can also launch “rundll32.exe” with that function. For example, it can be a child of **“Google Chrom”**, **“MSGEDGE” or “IE”** when launching the **“inetcpl.cpl”** for proxy / network configuration.

If you want more details about CPL and how malware is using it, you can read this trend micro research paper called **.**

DEVCLNT.DLL — “DavSetCookie” (Web Dav Client)

One of the mysterious command lines in a “rundll32.exe” instance that’ll show up a lot in the logs, takes the following format.

When using the **“file://”** protocol, whether be it in a word file, or via share windows will sometimes use (*if SMB is disabled in some cases*) the WebDav Client to request these files. When that happens a request will be made via the “rundll32.exe” utility.

The parent process of such requests will be “svchost.exe” like so. (The “-s WebClient” is not obligatory)

Malware like Emotet has already used this technique in the past. So always analyze the host that is present in this type of command line and make sure that everything is legitimate.

RUNDLL32.EXE — “-sta” / “-localserver” Flags

A lesser known command line arguments are the **“-sta”** and **“-localserver”**. Which both can be used to load malicious registered COM objects.

If you see in your logs or a process running with one of the following command line arguments.

You need to verify the corresponding registry key **[\HKEY_CLASSES_ROOT\CLSID\<GUID>]** and its sub-keys and values for any malicious **DLL** or **SCT** script.

I highly suggest you read blog post for a detailed explanation on this technique and check hexacorn for the "-localserver" variant.

Abusing the COM Registry Structure: CLSID, LocalServer32, &amp; InprocServer32

TL;DR Vendors are notorious for including and/or leaving behind Registry artifacts that could potentially be abused by…

bohops.com

RUNDLL32.EXE — Executing HTML / JAVASCRIPT

One other command line argument that attackers may use with **"rundll32.exe"** is the **"javascript"** flag.

In fact a **"rundll32.exe"** instance can run HTML / JavaScript code using the **"mshtml.dll"** and the **"javascript"** keyword **(See Below).**

I've never seen this used in a legitimate way. So if you spot this in your logs, it is worth investigating.