

Monitoring malware abusing CVE-2020-1599

 webcache.googleusercontent.com/search

CVE-2020-1599 is a vulnerability that can be abused by adding data (that will be later executed) to the signature section of a file, for instance appending a VB script. Unfortunately, Microsoft signature chain certification will not detect that the signature was modified and accept the file as legitimately signed, which can be used to avoid security checks. This is all described in this blog post by our colleagues at Checkpoint, also explaining how ZLoader is using this technique for persistence in recent campaigns.

A non-malicious file abusing this technique can be found here. The file is not malicious per se, as it simply opens the calc.exe utility.

```
000B0440 08 bd 00 ea ec bd c9 c7 76 80 b0 2d 98 3b 51 50 .....v...-.;QP
000B0450 9a db 8a b5 79 d3 0c d2 24 6c 3c 1a 73 eb 02 ee .....y...$l<.s...
000B0460 57 3f 2f 26 51 c1 1b d2 13 c5 a0 6d e6 a5 ad 1c W?/&Q.....m.....
000B0470 c7 5a f8 73 db bd 67 a9 0d bb c7 7f db 1d 14 8c .Z.s.g.....
000B0480 dd c7 36 c5 78 e5 29 52 c6 20 be fd 1f 9b 6a 85 ..6.x.)R.....j.
000B0490 8c ec 00 00 00 00 00 00 3c 73 63 72 69 70 74 20 .....<script
000B04A0 6c 61 6e 67 75 61 67 65 3d 22 56 42 53 63 72 69 language="VBscri
000B04B0 70 74 22 3e 0a 53 65 74 20 6f 62 6a 20 3d 20 47 pt">.Set obj = G
000B04C0 65 74 4f 62 6a 65 63 74 28 22 6e 65 77 3a 43 30 etObject("new:CO
000B04D0 38 41 46 44 39 30 2d 46 32 41 31 2d 31 31 44 31 8AFD90-F2A1-11D1
000B04E0 2d 38 34 35 35 2d 30 30 41 30 43 39 31 46 33 38 -8455-00A0C91F38
000B04F0 38 30 22 29 0a 6f 62 6a 2e 44 6f 63 75 6d 65 6e 80").obj.Document
000B0500 74 2e 41 70 70 6c 69 63 61 74 69 6f 6e 2e 53 68 t.Application.Sh
000B0510 65 6c 6c 45 78 65 63 75 74 65 20 22 63 61 6c 63 ellExecute "calc
000B0520 2e 65 78 65 22 2c 4e 75 6c 6c 2c 22 43 3a 5c 57 .exe", Null, "C:\W
000B0530 69 6e 64 6f 77 73 5c 53 79 73 74 65 6d 33 32 22 indows\System32"
000B0540 2c 4e 75 6c 6c 2c 30 0a 73 65 6c 66 2e 63 6c 6f ,Null,0.self.clo
000B0550 73 65 0a 3c 2f 73 63 72 69 70 74 3e 20 20 20 20 se.</script>
```

This malicious technique can be mitigated as described here.

In order to monitor any additional malware abusing this vulnerability, we decided to create a YARA and run a VirusTotal Livehunt, so we will get notified any time a new suspicious file shows up in VirusTotal:

```
import "pe"
import "vt"
```

```
rule CVE-2020-1599_suspicious_signed {
```

```
meta:
```

```
author = "@fcojsantos"
created = "2022.01.07"
```

```
reference = "https://research.checkpoint.com/2022/can-you-trust-a-files-digital-signature-new-zloader-campaign-exploits-microsofts-signature-verification-putting-users-at-risk/"
```

strings:

```
$script = "<script" nocase  
$script2 = "language" nocase
```

```
$script3 = "vbscript" nocase
```

condition:

```
pe.is_pe  
and pe.number_of_signatures > 0  
and not for all i in (0..pe.number_of_signatures - 1): (  
    pe.signatures[i].valid_on(pe.timestamp)
```

```
)
```

```
// Searches for script literal from the signature offset on
```

```
and $script in
```

```
(pe.data_directories[pe.IMAGE_DIRECTORY_ENTRY_SECURITY].virtual_address..filesize)
```

```
and $script2 in
```

```
(pe.data_directories[pe.IMAGE_DIRECTORY_ENTRY_SECURITY].virtual_address..filesize)
```

```
and $script3 in
```

```
(pe.data_directories[pe.IMAGE_DIRECTORY_ENTRY_SECURITY].virtual_address..filesize)
```

```
and for any tag in vt.metadata.tags : ( tag == "signed" )
```

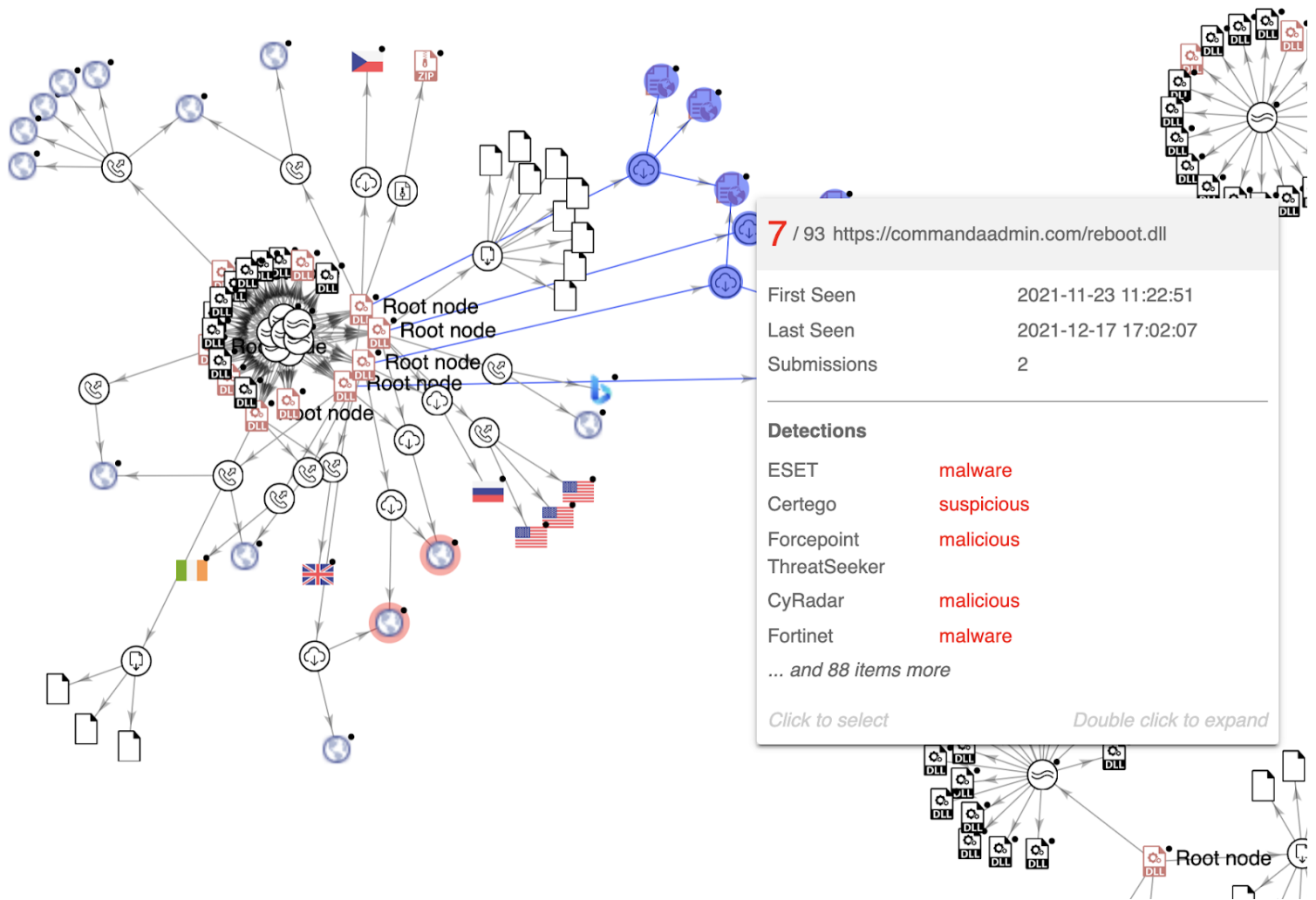
```
}
```

This YARA searches for suspicious script-related strings appended to the signature. However, YARA cannot check the certificate chain that confirms if the signature itself is valid or not, it only checks that the certificate exists. And here is where the YARA's vt module comes to the rescue.

In this case, the last condition 'for any tag in vt.metadata.tags : (tag == "signed")' will check that there exists at least one "signed" tag for the file, meaning that Microsoft Windows WinVerifyTrust function confirms this is a fully valid signature (it is not, as it abuses CVE-2020-1599).

Now, armed with this, we can find several interesting samples abusing this vulnerability that we added to a VT collection.

Additionally, we were interested in understanding how these files were distributed. We created a small graph to visualize any distribution vectors:



In addition to teamworks455[.]com (already listed as malicious in Checkpoint’s blog post), we found commandadmin[.]com distributing similar malware. You can monitor any malware distributed in the wild by these domains with the following VT intelligence query:

entity:file (itw:commandadmin.com or itw:teamworks455.com)

This query returns some of the indicators already published by Checkpoint plus a few new ones that might be interesting to take a look at.

We hope this post will be useful to understand how we can quickly monitor and do some hunting every time attackers use new techniques. Happy hunting!