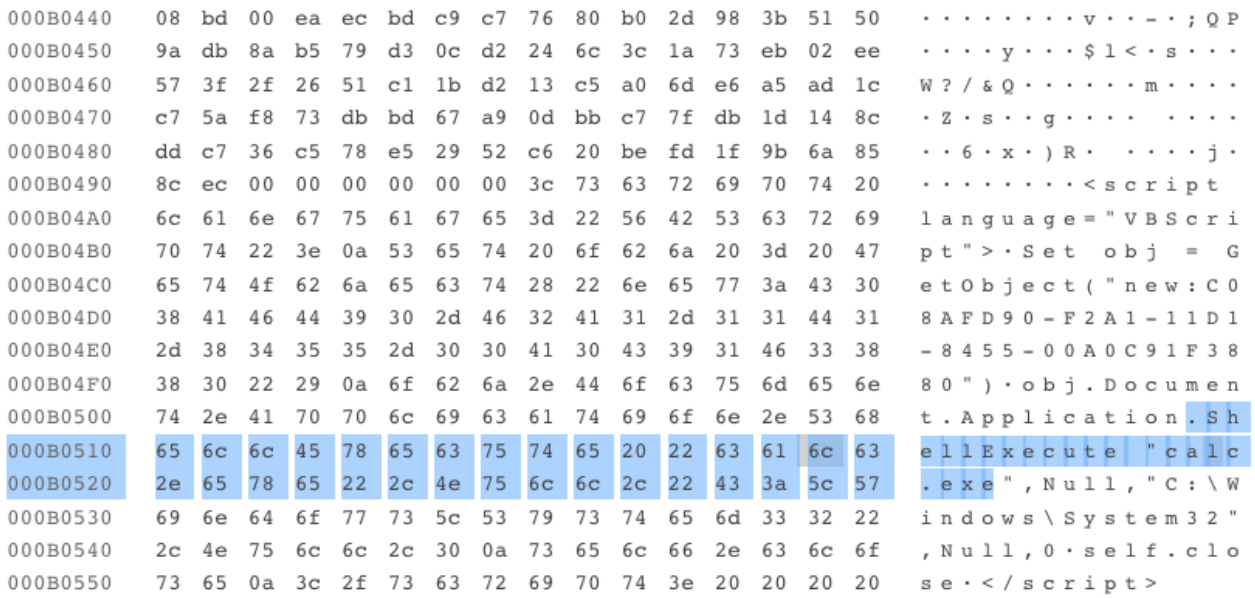
Monitoring malware abusing CVE-2020-1599

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.



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"
```

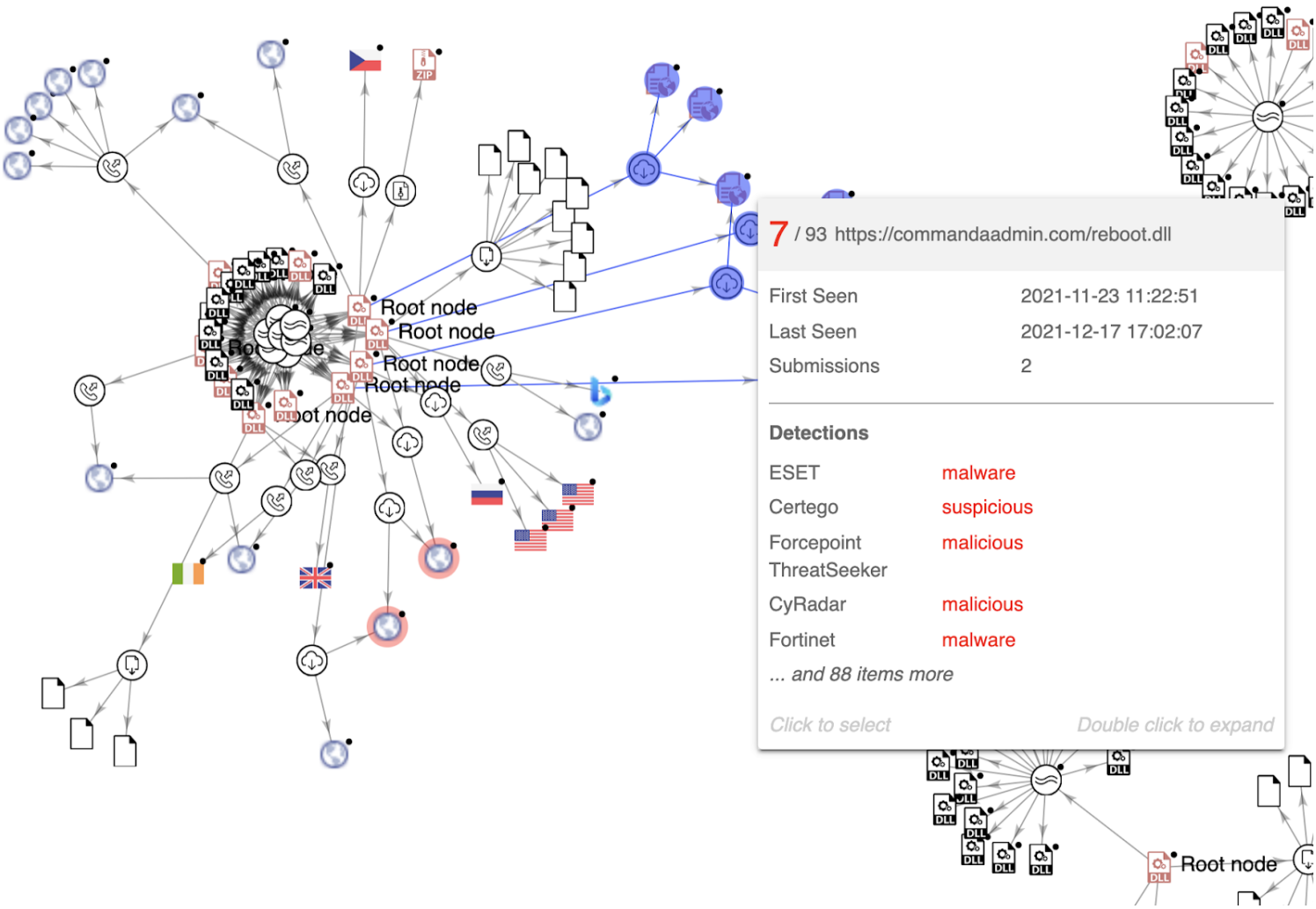
```
reference = "https://research.checkpoint.com/2022/can-you-trust-a-files-digital-signature-new-zloader-campaign-exploits-microsofts-signature-verification-putting-users-at-risk/"


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 commandaadmin[.]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:commandaadmin.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!