Meet NEMTY Successor, Nefilim/Nephilim Ransomware

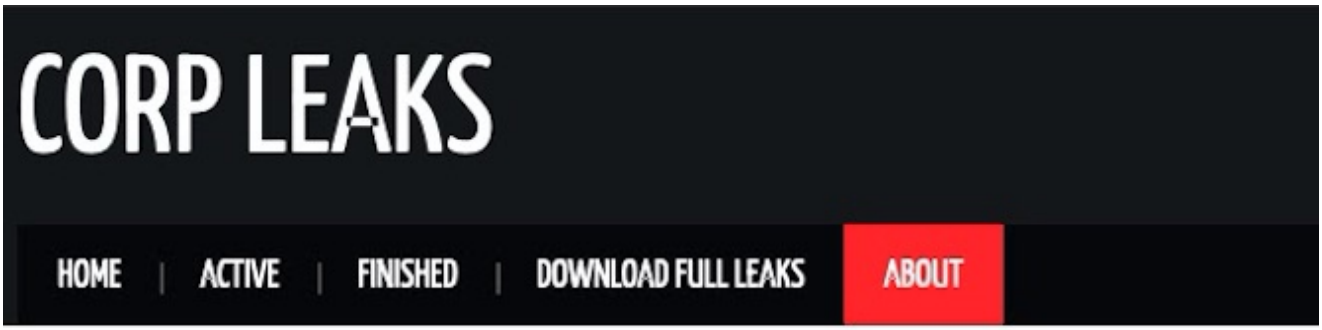
labs.sentinelone.com/meet-nemty-successor-nefilim-nephilim-ransomware/

Jim Walter

Ransomware families NEMTY, Nefilim and Nephilim continue to evolve and merge, taking on aspects of other successful variants that aim to encrypt and extort.

This is an interesting time to study and follow ransomware trends. In particular, over the last year or two, we have seen an expansion of 'mainstream' ransomware even further into the data extortion and theft realm. It is one thing to have files encrypted, but having to treat every ransomware infection as a breach adds multiple new layers of complexity for victims of these campaigns. This is especially complex with GDPR and similar legal and compliance hurdles to now figure in.

Ransomware families like Maze, CLOP, DoppelPaymer, Sekhmet, and Nefilim/Nephilim are examples of threats which, upon infection, result in this complex issue for their victims. While Maze, DopplePayer & REvil tend to get the bulk of media coverage, Nephilim is another family which has very quickly risen to prominence with multiple damaging campaigns that threaten to publish victims' sensitive information in the event they fail to 'cooperate' with the attacker's demands.



Overview

Nefilim emerged in March 2020 and shares a substantial portion of code with another ransomware family, NEMTY. The exact relationship between the actors behind NEMTY and Nefilim/Nephilim is less than clear.

NEMTY launched in August of 2019 as a public affiliate program, and has since gone private. Current data indicates that rather than the same actors being behind both families, it is more likely that those behind Nephilim 'acquired' necessary code out of NEMTY in one way or another.

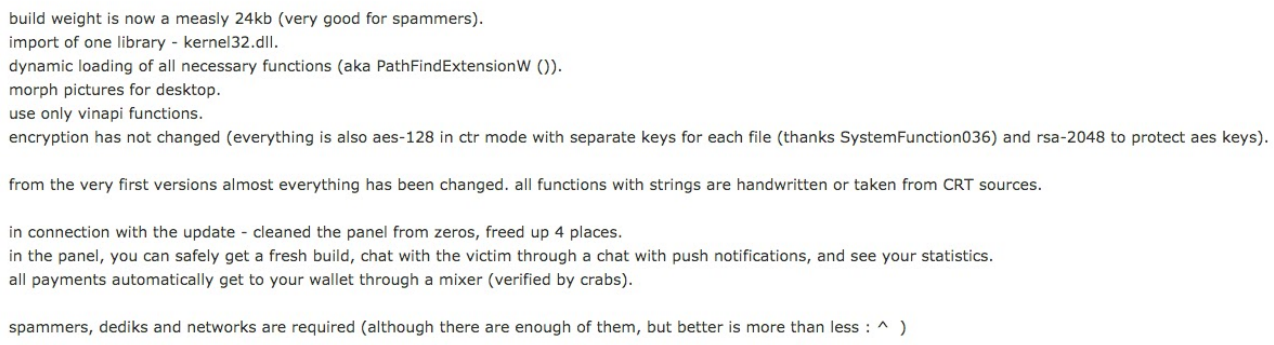
The two primary differences between Nefilim and NEMTY are the payment model, and the lack of a RaaS operation. Nefilim instructs victims to contact the attackers via email, as opposed to directing them to a TOR-based payment portal. To add even more confusion to the family tree, Nefilim appears to have evolved into 'Nephilim', and the two are technically similar, differentiated primarily by extension and artifacts in encrypted files.

However, there is also intelligence indicating that NEMTY has continued and forked into a new 'NEMTY Revenue' version. This comes after the actors behind NEMTY announced that they would be taking the threat private (no more publicly accessible RaaS operation).

Technically, Nephilim is not dissimilar from other well-known ransomware families. The primary method of delivery is currently vulnerable RDP services. Once the attackers have compromised the environment via RDP, they then proceed to establish persistence, to locate and exfiltrate additional credentials where possible, and then to deliver the ransomware payloads to their intended targets.

Nephilim Encryption Protocols

In the Nephilim samples we have analyzed the actual file encryption is handled via a tag-team of AES-128 and RSA-2048. Note, the original vendor behind Nefilim/Nephilim advertises it as such as well.


build weight is now a measly 24kb (very good for spammers).
import of one library - kernel32.dll.
dynamic loading of all necessary functions (aka PathFindExtensionW ()).
morph pictures for desktop.
use only vinapi functions.
encryption has not changed (everything is also aes-128 in ctr mode with separate keys for each file (thanks SystemFunction036) and rsa-2048 to protect aes keys).

from the very first versions almost everything has been changed. all functions with strings are handwritten or taken from CRT sources.

in connection with the update - cleaned the panel from zeros, freed up 4 places.
in the panel, you can safely get a fresh build, chat with the victim through a chat with push notifications, and see your statistics.
all payments automatically get to your wallet through a mixer (verified by crabs).

spammers, dediks and networks are required (although there are enough of them, but better is more than less : ^)

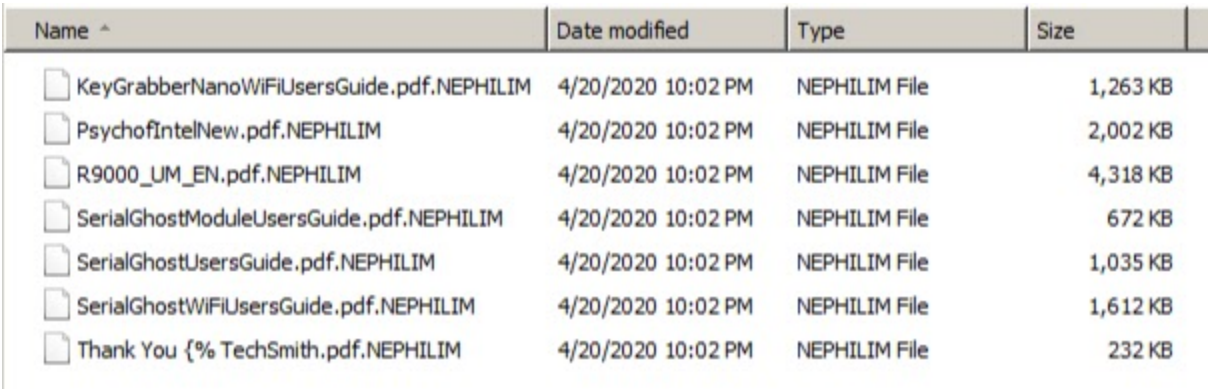
Specific files are encrypted using AES-128. At that point, an RSA-2048 public key is used to encrypt the AES encryption key. The public key is subsequently embedded in the ransomware executable payloads. This is one area that differs from pure NEMTY, which is known to have used different key lengths. For example, prior versions of NEMTY have used RSA-8192 as a “master key” for encryption of target configuration data along with the rest of the keys (src: Acronis).

We are also aware of variants of NEMTY that utilize an RSA-1024 public key for processing the AES encryption key. Also, with earlier versions of NEMTY, there was variance across how files of specific size ranges were handled. Later versions of NEMTY (aka NEMTY REVENUE 3.1) utilize AES-128 in counter mode, along with RSA-2048 for encrypting the AES keys.

At this time only the actors behind Nephilim are able to decrypt affected files. That is to say, there are no known flaws or methods to bypass the attackers’ safeguards on the encrypted files.

Nephilim Post-infection Behavior

After infection, encrypted files are given the `.Nefilim` or `.NEPHILIM` extension. A similarly named ransom note is deposited in directories containing encrypted files.



Name	Date modified	Type	Size
KeyGrabberNanoWiFiUsersGuide.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	1,263 KB
PsychofIntelNew.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	2,002 KB
R9000_UM_EN.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	4,318 KB
SerialGhostModuleUsersGuide.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	672 KB
SerialGhostUsersGuide.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	1,035 KB
SerialGhostWiFiUsersGuide.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	1,612 KB
Thank You {% TechSmith.pdf.NEPHILIM	4/20/2020 10:02 PM	NEPHILIM File	232 KB

In some cases, with Nephilim, the 'NEPHILIM-DECRYPT.txt' will only be written to ~AppDataLocalVirtualStore. Location and name of the locally-stored desktop wallpaper varies. In recent Nephilim infections, the alternate desktop image is written to %temp% with the filename 'god.jpg'.

Strings, Distinguishing Traits

Another hallmark of Nephilim is the use of embedded strings and compiler paths to send "subtle messages", primarily to researchers and analysts it would seem. For example, the following compiler path can be found in these samples (both compiled on April 7, 2020):

```
b8066b7ec376bc5928d78693d236dbf47414571df05f818a43fb5f52136e8f2e
fcc2921020690a58c60eba35df885e575669e9803212f7791d7e1956f9bf8020
```

While the sample

```
d4492a9eb36f87a9b3156b59052ebaf10e264d5d1ce4c015a6b0d205614e58e3
```

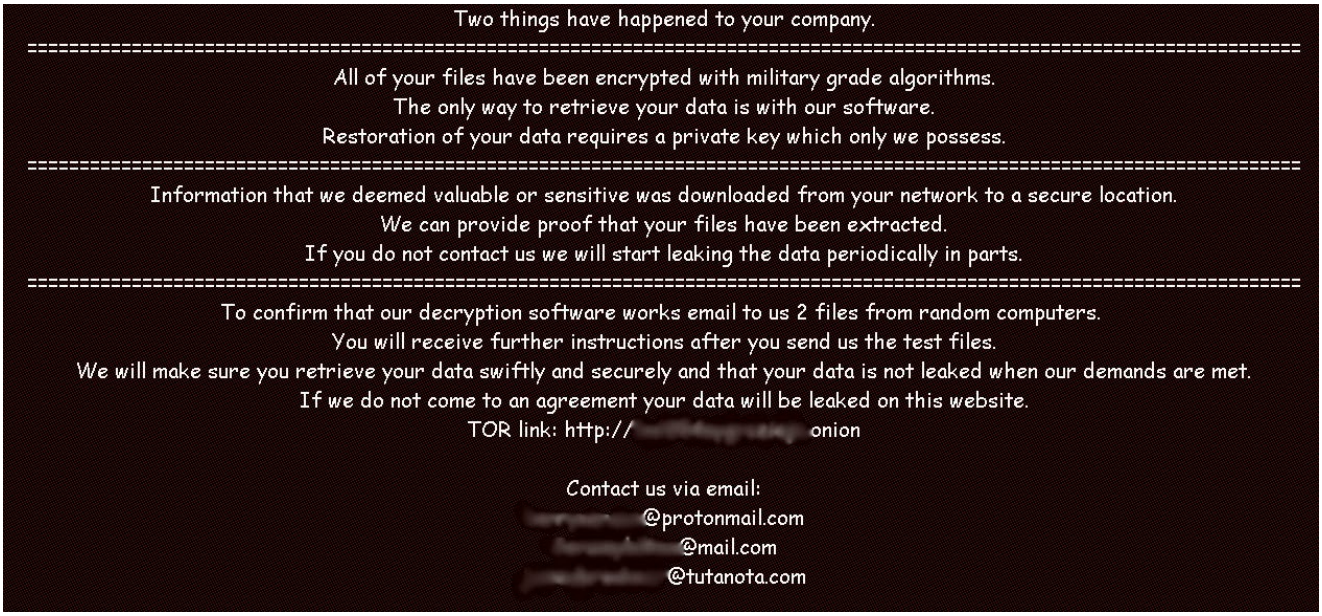
from March 2020 contains additional jabs at specific AV vendors.

```
.rdata:0040CB08 ProcName        db 'SystemFunction036',0
.rdata:0040CB08                                         ; DATA XREF: sub_401B4F+25↑o
.rdata:0040CB1A                 align 4
.rdata:0040CB1C ; const char Str[]
.rdata:0040CB1C Str             db 'oh how i did it??? bypass sofos hah',0
.rdata:0040CB1C                                         ; DATA XREF: sub_401B93+D0↑o
.rdata:0040CB40 ; const wchar_t aHowToFuckAllTh
.rdata:0040CB40 aHowToFuckAllTh:                        ; DATA XREF: sub_401B93+18B↑o
.rdata:0040CB40                 text "UTF-16LE", 'how to fuck all the world?',0
.rdata:0040CB76                 align 4
.rdata:0040CB78 ; const char aFukSosorin[]
.rdata:0040CB78 aFukSosorin     db 'fuk sosorin',0      ; DATA XREF: sub_401B93+loc_401E8C↑o
.rdata:0040CB84 ; const char aFukAnlab[]
.rdata:0040CB84 aFukAnlab       db 'fuk anlab',0        ; DATA XREF: sub_401B93+3C3↑o
.rdata:0040CB8E                 align 10h
.rdata:0040CB90 ; const char aInvalidStringP[]
.rdata:0040CB90 aInvalidStringP db 'invalid string position',0
.rdata:0040CB90                                         ; DATA XREF: sub_402241+14↑o
.rdata:0040CB90                                         ; sub_40241C+A↑o ...
.rdata:0040CBA8 ; const char aStringTooLong[]
.rdata:0040CBA8 aStringTooLong  db 'string too long',0  ; DATA XREF: sub_40248D+E↑o
.rdata:0040CBA8                                         ; c_sub_4019C7_vvvtest+5↑o ...
.rdata:0040CBB8 ; const CHAR szContainer[]
.rdata:0040CBB8 szContainer     db 'rsa public',0       ; DATA XREF: sub_402B29+53↑o
.rdata:0040CBC3                 align 4
.rdata:0040CBC4 ; const wchar_t aSFQ
.rdata:0040CBC4 aSFQ:                                   ; DATA XREF: sub_402C32+9E↑o
.rdata:0040CBC4                 text "UTF-16LE", '" /s /f /q',0
.rdata:0040CBDA                 align 10h
```

This sample was also referenced by @malwrhunterteam in a March 13th tweet.

Name & Shame Strategy

Nefilim/Nephilim also threatens to publish sensitive information from the infected environments in the event that the victim refuses to cooperate with the attackers' demands, as evidenced in this typical Nephilim ransom note.



Attempting to negotiate, or refusal to pay, fall under the category of non-compliance. To date, two companies have been published on Nephilim's "shaming" websites (clearnet and TOR-based). It is worth noting that initially, all the companies listed on their site were oil and energy companies. However, between April 23 and April 27, 2020, the group has added three additional victims to the site. One of these is another large oil and gas company, and the other two are classified as "Apparel and Fashion" and "Engineering and Construction Services".

Multiple other familles follow this same practice, which turns "basic" ransomware infections into full (and sometimes catastrophic) data breaches. Other well known families embracing this model are Maze, REvil, DoppelPaymer, CLOP, Sekhmet, and more recently, Ragnar. We note that Nefilim/Nephilim is also one of the families that has "vowed" not to attack medical entities, nonprofits and other "critical" entities during the current pandemic.

Conclusion

Protecting your environment against threats like Nephilim is more critical than ever. In order to prevent loss of data and the consequences of a large-scale data breach, organizations must rely on a modern, well maintained, and properly-tuned and trusted security solution. Prevention is key with these attacks. Even in the event that the encryption/data-loss can be mitigated through decryptors, backups or rollbacks, victims still face the problem of their data being posted publicly. We encourage our customers to analyze and understand the threats and to take swift and appropriate action to prevent incidents occurring in the first place.

SentinelOne Endpoint Protection detects and prevents malicious actions associated with NEMTY, Nefilim, and Nephilim.

Indicators of Compromise

For convenience, we provide both SHA256 and SHA1 hashes below.

SHA256

8be1c54a1a4d07c84b7454e789a26f04a30ca09933b41475423167e232abea2b
b8066b7ec376bc5928d78693d236dbf47414571df05f818a43fb5f52136e8f2e
3080b45bab3f804a297ec6d8f407ae762782fa092164f8ed4e106b1ee7e24953
7de8ca88e240fb905fc2e8fd5db6c5af82d8e21556f0ae36d055f623128c3377
b227fa0485e34511627a8a4a7d3f1abb6231517be62d022916273b7a51b80a17
3bac058dbea51f52ce154fed0325fd835f35c1cd521462ce048b41c9b099e1e5
353ee5805bc5c7a98fb5d522b15743055484dc47144535628d102a4098532cd5
5ab834f599c6ad35fcd0a168d93c52c399c6de7d1c20f33e25cb1fdb25aec9c6
52e25bdd600695cfed0d4ee3aca4f121bfebf0de889593e6ba06282845cf39ea
35a0bced28fd345f3ebfb37b6f9a20cc3ab36ab168e079498f3adb25b41e156f
7a73032ece59af3316c4a64490344ee111e4cb06aaf00b4a96c10adfdd655599
08c7dfde13ade4b13350ae290616d7c2f4a87cbeac9a3886e90a175ee40fb641

D4492a9eb36f87a9b3156b59052ebaf10e264d5d1ce4c015a6b0d205614e58e3
B8066b7ec376bc5928d78693d236dbf47414571df05f818a43fb5f52136e8f2e
fcc2921020690a58c60eba35df885e575669e9803212f7791d7e1956f9bf8020

SHA1

4595cdd47b63a4ae256ed22590311f388bc7a2d8
1f594456d88591d3a88e1cdd4e93c6c4e59b746c
6c9ae388fa5d723a458de0d2bea3eb63bc921af7
9770fb41be1af0e8c9e1a69b8f92f2a3a5ca9b1a
e99460b4e8759909d3bd4e385d7e3f9b67aa1242
e53d4b589f5c5ef6afd23299550f70c69bc2fe1c
c61f2cdb0faf31120e33e023b7b923b01bc97fbf
0d339d08a546591aab246f3cf799f3e2aaee3889
bbcb2354ef001f476025635741a6caa00818cbe7
2483dc7273b8004ecc0403fbb25d8972470c4ee4
d87847810db8af546698e47653452dcd089c113e
E94089137a41fd95c790f88cc9b57c2b4d5625ba
Bd59d7c734ca2f9cbaf7f12bc851f7dce94955d4
f246984193c927414e543d936d1fb643a2dff77b
d87847810db8af546698e47653452dcd089c113e