Snake Ransomware Analysis Updates

insights.sei.cmu.edu/cert/2020/03/snake-ransomware-analysis-updates.html

Kyle O'Meara

March 23, 2020

In January 2020, Sentinel Labs published two reports on Snake (also known as Ekans) ransomware.[1][2] The Snake ransomware gained attention due to its ability to terminate specific industrial control system (ICS) processes. After reading the reports, I wanted to expand the corpus of knowledge and provide OT and IT network defenders with increased defense capabilities against Snake. The key takeaways from the Sentinel Labs' reports for additional analysis were the hash of the ransomware and the string decoder script from sysopfb.[3] Two questions I pursued were:

- Can I find more samples of the Snake ransomware?
- If yes, do these samples use the same string decoding process?

Discovering More Samples

By analyzing the code and applying a combination of using IDA, Pharos tools fn2hash and fn2yara, BigGrep, and the CERT/CC Malware Analysis and Storage System (MASS) repository, I was able to find one sample with a 100% function overlap with that of the known Snake ransomware sample.[4] The hashes of these samples are shown in Table 1 in the Appendix. In reviewing my BigGrep search parameter, I realized I had potentially limited my search results. I expanded my search parameter and found two more candidate samples which are shown in Table 2 in the Appendix.

Decoding the Strings

The string decoder, further referred to as the config dumper, decoded the same strings from the new Snake ransomware sample that were also found in the original Snake ransomware sample.[1] See the sysopfb link in the references section for complete decoded string list.[3] Unfortunately, the config dumper did not return any results for the two new candidate samples.

Again, using IDA and Pharos tools fn2hash and fn2yara, I wanted to see how much code overlap there was with these four files from Table 1 and Table 2 in the Appendix. The two new candidate samples shared a 100% function overlap. When comparing these two files to the Snake ransomware samples, there was only a 50% function overlap. With a significant function overlap between all of the four files, why didn't the config dumper work on the two new candidate samples?

Looking at the code further, I identified a 1-byte difference in the string decoding function in the new candidate samples versus the known Snake ransomware samples. I edited the config dumper and ran it on the new candidate samples. The modified config dumper was successful in decoding strings from the new candidate samples as shown in Table 3 in the Appendix.

The newly returned strings from the new candidate samples were different than those that were found in the known Snake ransomware samples. However, using the new config dumper, I successfully decoded new strings from the known Snake ransomware samples. These strings all appear to be host intrusion prevention system (HIPS) process and service names, as shown in Table 4 in the Appendix.

Summary of Findings

Through my additional analysis process, I discovered another Snake ransomware sample as well as new candidate samples. However, dynamic analysis demonstrated that these new candidate samples did not act like ransomware. Upon execution, the new candidates tried to establish a connection to IP address 18.222.249[.]59 on port 7777. Without allowing the candidate sample to establish a connection, I saw no further action from the candidate samples. The assumption is that the Snake ransomware and the new candidate samples are potentially created by a similar actor, given the large code overlap as well as the nearly identical string decoding routine.

I created a YARA rule to identify samples that contain a similar string decoding function, as shown in Table 5 in the Appendix.

I also developed an updated config dumper which decodes the new set of strings. This config dumper is available upon request.

In another report, Dragos highlights that the Snake ransomware terminate process list is similar to the list found in the MegaCoretx ransomware.[5] My analysis uncovered an additional 252 decoded strings related to HIPS processes that the Snake ransomware attempts to terminate. These 252 processes are found in the 1104 processes list in the Accenture Security MegaCortex ransomware report.[6] However,

after completing similar analyses, as mentioned above, as well as testing known YARA rules, I found that the Snake and MegaCortex ransomwares shared no code overlap. I believe it is a matter of coincidence that there is an overlap in this process list. The possible reasons for this coincidence could include that the Snake ransomware took information from the Accenture report on MegaCortex or used the published curated open source HIPS process list.[7]

Conclusion

I have provided more samples, a YARA rule, new config dumper, and new decoded data (see the tables in the Appendix). This information, in addition to previous industry analyses, will allow for network defenders in the OT and IT space to increase their defense capabilities against the Snake ransomware.

Appendix

Table 1: Snake Ransomware Hashes

e5262db186c97bbe533f0a674b08ecdafa3798ea7bc17c705df526419c168b60
a5a7e6ddf99634a253a060adb1f0871a5a861624382e8ca6d086e54f03bed493

Table 2: New Candidate Hashes

b17863d41c0b915052fea85a354ec985280f4d38b46d64158a75b17ef89d76da
a8f0ff40d1e624dd2aad4d689ed47a900e4f719923647cacb58d1a4809c7bd31

Table 3: Decoded Strings from New Candidate Samples

u
u
https://18.222.249.59/uploaad
./123
ok
18.222.249.59:7777
tcp
POST
https://18.222.249.59:443/uploaad
./123
OK

title
endgame
Content-Type
multipart/form-data
file
endgame
Y23QyJCj%kAK
POST
Content-Type

Table 4: New Decoded Strings from Snake Ransomware Samples

acctmgr.exe	nprotect.exe	savservice.exe
alertsvc.exe	npscheck.exe	savui.exe
almon.exe	npssvc.exe	sbserv.exe
alsvc.exe	nscsrvce.exe	scanfrm.exe
alunotify.exe	nsctop.exe	scfmanager.exe
alupdate.exe	nsmdemf.exe	scfservice.exe
aluschedulersvc.exe	nsmdmon.exe	scftray.exe
aphost.exe	nsmdreal.exe	schdsrvc.exe
appsvc32.exe	nsmdsch.exe	schupd.exe
apvxdwin.exe	nsmdtr.exe	sdtrayapp.exe
asupport.exe	ofcdog.exe	seestat.exe
avltmain.exe	ofcpfwsvc.exe	semsvc.exe
ccap.exe	olfsnt40.exe	sesclu.exe
ccapp.exe	omslogmanager.exe	sevinst.exe
ccenter.exe	opscan.exe	sgbhp.exe
ccevtmgr.exe	op_viewer.exe	slee81.exe
ccproxy.exe	pagent.exe	smsectrl.exe
ccpxysvc.exe	pagentwd.exe	smselog.exe
ccsetmgr.exe	patch.exe	smsesjm.exe

certificationmanagerservicent.exe
checkup.exe
cka.exe
comhost.exe
cpdclnt.exe
csinject.exe
csinsm32.exe
csinsmnt.exe
dbserv.exe
defwatch
defwatch.exe
diskmon.exe
djsnetcn.exe
dlservice.exe
dltray.exe
doscan.exe
dwhwizrd.exe
dwwin.exe
emlibupdateagentnt.exe
entitymain.exe
execstat.exe
scanexplicit.exe
firewallgui.exe
fwcfg.exe
fws.exe
ghost_2.exe
ghosttray.exe
icepack.exe
idsinst.exe
inicio.exe
isntsmtp.exe
isntsysmonitor
ispwdsvc.exe
issvc.exe
isuac.exe
knownsvr.exe
kpf4gui.exe
kpf4ss.exe
lmon.exe
luall.exe
lucallbackproxy.exe
lucoms~1.exe
lucomserver.exe
lucoms.exe
lwdmserver.exe
managementagentnt.exe
mcui32.exe
mgntsvc.exe
mrf.exe
navapsvc.exe
navapw32.exe
navectrl.exe
navelog.exe
navesp.exe
navshcom.exe
navw32.exe
navwnt.exe
ndetect.exe
ngctw32.exe
ngserver.exe
nisoptui.exe
nisserv.exe
nisum.exe
nmain.exe
npfmntor.exe
pavbckpt.exe
pavjobs.exe
pavsrv52.exe
pccnt.exe
pccntupd.exe
pcctlcom.exe
pcscnsrv.exe
pctsauxs.exe
pctsgui.exe
pctssvc.exe
pctstray.exe
pmon.exe
poproxy.exe
pqibrowser.exe
pqv2isvc.exe
prevsrv.exe
procexp.exe
psctris.exe
psctrls.exe
pshost.exe
psimreal.exe
pskmssvc.exe
pviewer.exe
pview.exe
pxeservice.exe
qdcsfs.exe
qoeloader.exe
qserver.exe
ras.exe
rasupd.exe
ravalert.exe
rav.exe
ravmond.exe
ravmon.exe
ravservice.exe
ravstub.exe
ravtask.exe
ravtray.exe
ravupdate.exe
ravxp.exe
regmech.exe
reportersvc.exe
reportsvc.exe
rfwmain.exe
rfwproxy.exe
rfwsrv.exe
rfwstub.exe
rnav.exe
rnreport.exe
routernt.exe
rsnetsvr.exe
rstray.exe
sav32cli.exe
savadminservice.exe
savfmsectrl.exe
savfmselog.exe
savfmsesjm.exe
savfmsespamstatsmanager.exe
savfmsesp.exe
savfmsesrv.exe
savfmsetask.exe
savfmseui.exe
savmain.exe
savroam.exe
savscan.exe
smsesp.exe
smsesrv.exe
smsetask.exe
smseui.exe
sms.exe
smsx.exe
snac.exe
sndmon.exe
sndsrvc.exe
snhwsrv.exe
snicheckadm.exe
snichecksrv.exe
snicon.exe
snsrv.exe
spbbcsvc.exe
srvload.exe
sschk.exe
ssecuritymanager.exe
ssm.exe
svcharge.exe
svcntaux.exe
svdealer.exe
svframe.exe
svtray.exe
swdsvc.exe
sweepsrv.sys
swnetsup.exe
swnxt.exe
swserver.exe
symlcsvc.exe
symproxysvc.exe
symsport.exe
symtray.exe
symwsc.exe
sysdoc32.exe
tdimon.exe
tfgui.exe
tfservice.exe
tftray.exe
tfun.exe
tiaspn~1.exe
tmas.exe
tmntsrv.exe
tmpfw.exe
tmproxy.exe
tpsrv.exe
traflnsp.exe
trjscan.exe
trupd.exe
ucservice.exe
updtnv28.exe
upfile.exe
urllstck.exe
usrprmpt.exe
v2iconsole.exe
vpc32.exe
vpdn_lu.exe
vprosvc.exe
vptray.exe
webproxy.exe
wfxctl32.exe
wfxmod32.exe
wfxsnt40.exe
winlog.exe
wrspysetup.exe

Table 5: Snake Ransomware YARA Rule

```
rule Snake_Ransomware
{
   meta:
   author = “CERT/CC RE Team”
   description = “Snake Ransomware String Decoder Function”
   date = “21 Feb 2020”

strings:
   $bytes = { 8D 05 ?? ?? ?? ?? 89 44 24 04 C7 44 24 08 05 00 00 00 E8 ?? ?? ?? ?? 8B 44 24 0C 89 44 24 64 8B 4C 24 10 89 4C 24 18
8D 54 24
24 89 14 24 8D 15 ?? ?? ?? ?? 89 54 24 04 C7 44 24 08 05 00 00 00 E8 ?? ?? ?? ?? }
condition:
   $bytes
}
```


rule Snake_Ransomware
{
meta:
author = “CERT/CC RE Team”
description = “Snake Ransomware String Decoder Function”
date = “21 Feb 2020”

strings:
$bytes = { 8D 05 ?? ?? ?? ?? 89 44 24 04 C7 44 24 08 05 00 00 00 E8 ?? ?? ?? ?? 8B 44 24 0C 89 44 24 64 8B 4C 24 10 89 4C 24 18 8D
54 24 24 89 14 24 8D 15 ?? ?? ?? ?? 89 54 24 04 C7 44 24 08 05 00 00 00 E8 ?? ?? ?? ?? }
condition:
$bytes
}

References

[1] https://twitter.com/VK_Intel/status/1214333066245812224

[2] https://labs.sentinelone.com/new-snake-ransomware-adds-itself-to-the-increasing-collection-of-golang-crimeware/

[3] https://github.com/sysopfb/open_mal_analysis_notes/blob/master/e5262db186c97bbe533f0a674b08ecdafa3798ea7bc17c705df526419c168b60.m

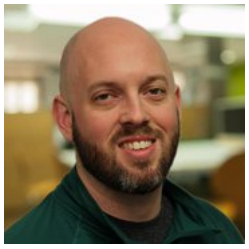
[4] https://github.com/cmu-sei/pharos

[5] https://dragos.com/blog/industry-news/ekans-ransomware-and-ics-operations/

[6] https://www.accenture.com/_acnmedia/pdf-106/accenture-technical-analysis-megacortex.pdf

[7] https://github.com/v1ado/HIPS_LIPS

WRITTEN BY



MORE BY THE AUTHOR

API Hashing Tool, Imagine That

March 25, 2019 • By Kyle O'Meara , CERT Insider Threat Center

MORE IN CERT/CC VULNERABILITIES

The Latest Work from the SEI: Coordinated Vulnerability Disclosure, Cybersecurity Research, Cyber Risk and Resilience, and the Importance of Fostering Diversity in Software Engineering

September 6, 2021 • By Douglas C. Schmidt

Vulnerabilities: Everybody’s Got One!

June 16, 2021 • By Leigh Metcalf

CERT/CC Comments on Standards and Guidelines to Enhance Software Supply Chain Security

June 1, 2021 • By Jonathan Spring

Cat and Mouse in the Age of .NET

November 19, 2020 • By Brandon Marzik

Adversarial ML Threat Matrix: Adversarial Tactics, Techniques, and Common Knowledge of Machine Learning

October 22, 2020 • By Jonathan Spring