

Snake Ransomware Analysis Updates

By Kyle O'Meara March 23, 2020

Published: 2020-03-23 · Archived: 2026-04-05 16:17:29 UTC

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 ransoms 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	nscsrvc.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	schdsrv.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	pavbckpt.exe	smsesp.exe
checkup.exe	pavjobs.exe	smsesrv.exe
cka.exe	pavsrv52.exe	smsetask.exe
comhost.exe	pccnt.exe	smseui.exe
cpdclnt.exe	pccntupd.exe	sms.exe
csinject.exe	pcctlcom.exe	smsx.exe
csinsm32.exe	pcscsrv.exe	snac.exe
csinsmnt.exe	pctsauxs.exe	sndmon.exe
dserv.exe	pctsgui.exe	sndsrv.exe
defwatch	pctssvc.exe	snhwsrv.exe
defwatch.exe	pctstray.exe	snicheckadm.exe
diskmon.exe	pmon.exe	snichecksrv.exe
djsnetcn.exe	poproxy.exe	snicon.exe
dlservice.exe	pqibrowser.exe	snsrv.exe
dltray.exe	pqv2isvc.exe	spbbcsvc.exe
doscan.exe	prevsrv.exe	srvload.exe
dwhwizrd.exe	procexp.exe	sschk.exe
dwwin.exe	psctris.exe	ssecuritymanager.exe
emlibupdateagentnt.exe	psctrls.exe	ssm.exe
entitymain.exe	pshost.exe	svcharge.exe
execstat.exe	psimreal.exe	svcntaux.exe
scanexplicit.exe	pskmssvc.exe	svdealer.exe
firewallgui.exe	pviewer.exe	svframe.exe
fwcfg.exe	pview.exe	svtray.exe
fws.exe	pxeservice.exe	swdsvc.exe
ghost_2.exe	qdcfs.exe	sweepsrv.sys

ghosttray.exe	goeloder.exe	swnetsup.exe
icepack.exe	qserver.exe	swnxt.exe
idsinst.exe	ras.exe	swserver.exe
inicio.exe	rasupd.exe	symlcsvc.exe
isntsmtp.exe	ravalert.exe	symproxysvc.exe
isntsysmonitor	rav.exe	symSPORT.exe
ispwdsvc.exe	ravmond.exe	symtray.exe
issvc.exe	ravmon.exe	symwsc.exe
isuac.exe	ravservice.exe	sysdoc32.exe
knownsvr.exe	ravstub.exe	tdimon.exe
kpf4gui.exe	ravtask.exe	tfgui.exe
kpf4ss.exe	ravtray.exe	tfSERVICE.exe
lmon.exe	ravupdate.exe	tftray.exe
luall.exe	ravxp.exe	tfun.exe
lucallbackproxy.exe	regmech.exe	tiaspn~1.exe
lucoms~1.exe	reportersvc.exe	tmas.exe
lucomserver.exe	reportsvc.exe	tmntsrv.exe
lucoms.exe	rfwmain.exe	tmpfw.exe
lwdmserver.exe	rfwproxy.exe	tmproxy.exe
managementagentnt.exe	rfwsrv.exe	tpsrv.exe
mcui32.exe	rfwstub.exe	traflnsp.exe
mgntsvc.exe	rnav.exe	trjscan.exe
mrf.exe	rnreport.exe	trupd.exe
navapsvc.exe	routernt.exe	ucSERVICE.exe
navapw32.exe	rsnetsvr.exe	updtv28.exe
navctrl.exe	rstray.exe	upfile.exe
navelog.exe	sav32cli.exe	urlstck.exe
navesp.exe	savadminservice.exe	usrprmt.exe
navshcom.exe	savfmsectrl.exe	v2iconsole.exe
navw32.exe	savfmselog.exe	vpc32.exe
navwnt.exe	savfmsesjm.exe	vpdn_lu.exe
ndetect.exe	savfmsespamstatsmanager.exe	vprosvc.exe
ngctw32.exe	savfmsesp.exe	vpTRAY.exe
ngserver.exe	savfmsesrv.exe	webproxy.exe
nisoptui.exe	savfmsetask.exe	wfxctl32.exe
nisserv.exe	savfmseui.exe	wfxmod32.exe
nisum.exe	savmain.exe	wfxsnt40.exe
nmain.exe	savroam.exe	winlog.exe
npfmntor.exe	savscan.exe	wrspyssetup.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  
}
```

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/e5262db186c97bbe533f0a674b08ecdafa3798ea7bc17c705df526419c16

[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

Source: <https://insights.sei.cmu.edu/cert/2020/03/snake-ransomware-analysis-updates.html>