the attacking abilities and strategies

Archived: 2026-04-06 00:08:01 UTC

Summary

This is the head part of the Akira ransom note, and it claims:
Whatever who you are and what your title is if you're reading this it means the internal infrastructure of your company is fully or partially dead, all your backups - virtual, physical - everything that we managed to reach - are completely removed. Moreover, we have taken a great amount of your corporate data prior to encryption.

Well, for now let's keep all the tears and resentment to ourselves and try to build a constructive dialogue. We're fully aware of what damage we caused by locking your internal sources.

As you know, recently ransomware has become so popular, and threat actors further expanded the attack surface to Linux. In 2023, I had collected many ransomwares that run on Linux and posted them to X (formerly Twitter), and last week I noted Akira ransom gang. I am very curious about what happened one year later.

Technical analysis

Basic info

The sample hashs:

md5 6B03B31C8CBD4A0A5829B63D16936ED3

Sha1 a90790c35bea365befd3af55cbedfffd2cc4481b

Operation system: Linux(ABI: 3.2.0)[AMD64, 64-bit, EXEC]

Packer: no

Messages on the screen and imply

The Akira uses /proc/stat to get system-wide statistics about CPU usage, system activity, and process counts. It also checks the number of CPUs with /proc/cpuinfo, and it will print out the tip messages on the screen which

including detected number of CPU, “no path to encrypt” if without any path parameter and the time It took, such as:

```
root@kali:~# ./bcae.elf
Detected number of cpus = 2
No path to encrypt
3ms
root@kali:~#
```

Fig.1-message without running

From the message, it seems that it is helpful for the ransomware group to debug and expand new abilities. Of course, it also implies they are developing

Static analysis

Supporting parameters and abilities

Let’s try a static analysis on IDA and look for some strings. The Akira ransomware supports many parameters to run, but it does not support command-line parameter help like “-h or /? or –help” to display them. Here they are:

1. –p(--encryption_path) to set the path of directory or file, e.g, -p=/root/abc .
2. –s(--share_file) to encrypt share file through network drive path.
3. –n(--encryption_percent) to encrypt with percent, such as to set –n=5, -n=10 with the character “%”.
4. –e(--exclude) to use “regular” expression to skip all specific files and not to encrypt, e.g. –e=”pwn*.*”
5. –fork to create a child process for encryption in the background without any message output

```
v44 = __readfsqword(0x28u);
sub_409C2E(v42);
sub_408E24(v42, a1, a2, 1LL);
v40 = "-p";
v41 = "--encryption_path";
sub_409A2C(v43, v42, &v40, 2LL);
sub_4FC330(v37, v43);
sub_4F8AD0(v43);
v40 = "-s";
v41 = "--share_file";
sub_409A2C(v43, v42, &v40, 2LL);
sub_4FC330(v38, v43);
sub_4F8AD0(v43);
v40 = "-n";
v41 = "--encryption_percent";
sub_409A2C(v43, v42, &v40, 2LL);
sub_4FC330(v39, v43);
sub_4F8AD0(v43);
v36[0] = "-e";
v36[1] = "--exclude";
sub_409A2C(v43, v42, v36, 2LL);
sub_4FC330(&v40, v43);
sub_4F8AD0(v43);
sub_4877D0(v35);
sub_509FE0(v43, "-fork", v35);
v16 = sub_409A02(v42, v43);
sub_507A30(v43);
sub_4877F0(v35);
```

Fig.2-Supporting parameters

From the design, the –p parameter is very convenient to encrypt the specified directory and files; the –s parameter is to further expand the attack surface with the network drive path; and the –n parameter is to make faster encryption, especially if the size of encrypted files is too large. And combining the following will mention the lock strategy and its multiple **LWP technique**s; all in all, it is a very convenient, faster, and more powerful design.

Ransom note and contact strategy

As you know, the ransomware named Akira is the cause of the file extension, and it will create a text file "akira_readme.txt," which we call a ransom note, including the common intel of threat from the attacker or the victim's information, such as an anonymous email address, onion address, Bitcoin address, and so on. At this ransomware as follows.

1. Publish victims address : hxxp[:]//akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3u2kollpj5z3z636bad[.]onion

2. Onion address for contact: hxxps[:]//akiralkzxzq2dsrzsrvbr2xgbbu2wgsmxryd4csgfameg52n7efvr2id[.]onion

3. Unique code for logging to chat: xxxx-xx-xxx-xxxx

4. Bitcoin address and Wallet: In the ransom note, it does not claim how many bitcoins to pay, and without exposing any wallet address provided by the Akira gang, the threat actors

From the two onion addresses we have found, which also include the ransom group name strings "Akira.".

And let's have a look at the ransom note as follows.

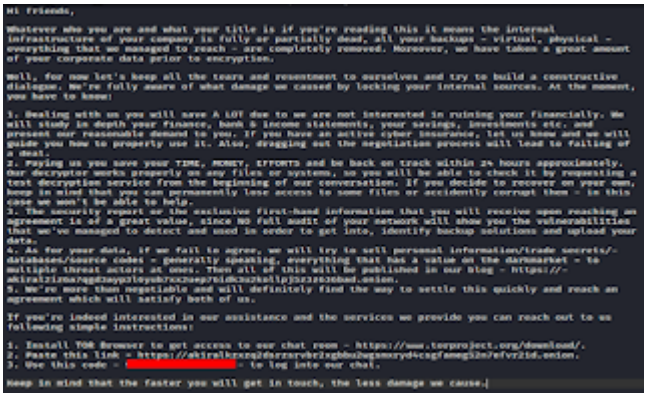

Fig.3-ransom note

Lock strategy for new extensions

Including the below important different types, such as database files, virtual machine files, disk images, and binary data formats, here they are as follows:

Database Files

Microsoft Access: .accdb, .accdc, .accde, .mdb

SQL-based Databases: .db, .db3, .sqlite, .sqlite3, .sdf, .mdf, .ndf

dBase & FoxPro: .dbf, .dbx, .fpt

Oracle Databases: .ora, .dbs, .dbc

Firebird & InterBase: .fdb, .gdb

IBM DB2: .db2

MySQL/MariaDB: .myd, .frm

Lotus Notes Database: .nsf, .ns2, .ns3, .ns4

Virtual Machine & Disk Image Files
Virtual Machine Files:

VMware: .vmdk, .vmem, .vmsn, .vmsd, .nvram, .vmx

VirtualBox: .vdi

Microsoft Hyper-V: .vhd, .vhdx, .avhd, .vmrs, .avdx, .vmcx

Parallels: .pvm

Disk Image Files:

ISO Image: .iso

QEMU: .qcow2, .raw

Virtual Server Files: .vsv

Backup & Log Files
Backup Files: .bak, .ndf, .sdf, .trc, .log

Checkpoints & Snapshots: .ckp, .snap

Error & Transaction Logs: .trm, .rpd, .sbf

Miscellaneous Data Files
Metadata & Configurations: .dad, .daschema, .dadiagrams, .pdm

Encryption & Key Storage: .kdb, .lgc

User & Profile Data: .usr, .hdb, .epim

Binary & Raw Data Files
.bin, .raw, .subvo, .gcow2

Dynamic analysis

LWPs technique and debug skill

Akira is creating multiple **Lightweight Processes (LWPs)**, which are likely **threads**. However, they seem to exit quickly when the numbers of the files are small. This makes debugging difficult.

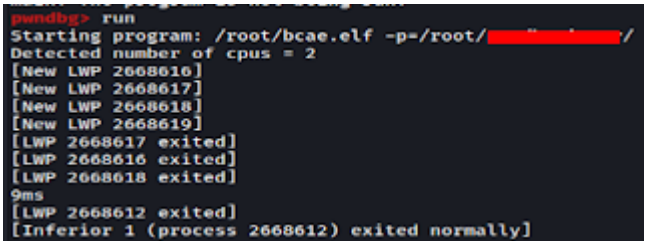

Fig.4-LWPs

To overcome the above problem, just set encryption like this: –p=/root, which will encrypt the whole root directory, it is so big and time-consuming. First press Ctrl+C to make an interrupt, and then using ***info threads*** to get how many threads it created and choose one with ***thread number*** and trying ***backtrace*** to debug.

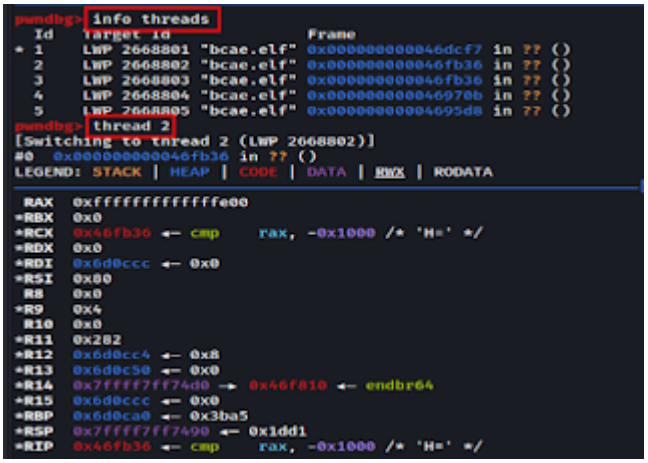

Fig.5-get threads and choose one thread to debug

Encryption algorithm strategy

on this variant, the Akira combing standard AES with RSA public-key cryptosystem as encryption strategy, each file encrypted was appending 512 bytes random data to the end, as you know, they are used to decrypt by RSA private key. It does encryption with the Nettle library. Let’s take one of them showing.

Fig.6- AES+RSA (Nettle cryptographic library)

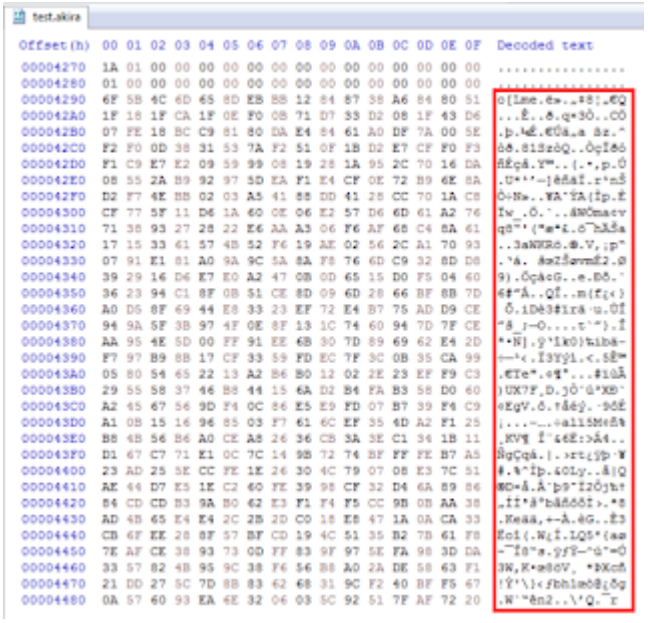

Fig.7-512 bytes of random data to the end of the encrypted file

Conclusion

From the above analysis, it appears that Akira tried to use a simple, convenient, faster, and more powerful strategy to expand their attacking campaign as threat actors, and they consciously avoided exposing personal information like wallet addresses, which means that they are an experienced ransom gang, a more hidden threat around the digital world; let’s pay close attention.

IoCs

Files:

md5 6B03B31C8CBD4A0A5829B63D16936ED3

Sha1 a90790c35bea365befd3af55cbedfffd2cc4481b

urls:

hxxps[:]//akiralkzxzq2dsrzsrvbr2xgbbu2wgsmxryd4csgfameg52n7efvr2id[.]onion

hxxps[:]//akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3u2kollpj5z3z636bad[.]onion

Akira Analysis Briefing

End.

Seeker(李标明) · @clibm079

China · Independent Malware Analyst & Researcher

Labels: #LinuxSecurity, #MalwareAnalysis, #ransomware, #ThreatIntel

---