Akira Ransomware Expands to Linux: the attacking abilities and strategies

malwareanalysisspace.blogspot.com/2025/03/akira-ransomware-expands-to-linux.html

Seeker(李标明)

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:

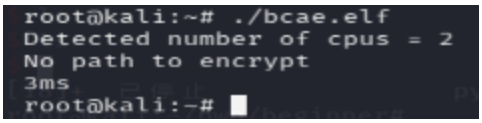


Fig.1-message without running

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 techniques**; 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

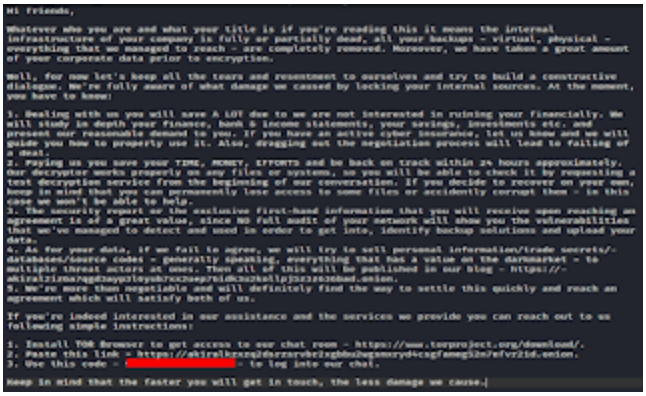
2. Onion address for contact: hxxps[:]//akiralkzxzq2dsrzsrvbr2xgbbu2wgsmxryd4csgfameg52n7efvr2id[.]onion

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.


Hi friends,

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. At the moment, you have to know:

1. Dealing with us you will save A LOT due to we are not interested in ruining your financially. We will study in depth your finance, bank & income statements, your savings, investments etc. and present our reasonable demand to you. If you have an active cyber insurance, let us know and we will guide you how to properly use it. Also, dragging out the negotiation process will lead to failing of a deal.
2. Paying us you save your TIME, MONEY, EFFORTS and be back on track within 24 hours approximately. Our decryptor works properly on any files or systems, so you will be able to check it by requesting a test decryption service from the beginning of our conversation. If you decide to recover on your own, keep in mind that you can permanently lose access to some files or accidently corrupt them - in this case we won't be able to help.
3. The security report or the exclusive first-hand information that you will receive upon reaching an agreement is of a great value, since NO full audit of your network will show you the vulnerabilities that we've managed to detect and used in order to get into, identify backup solutions and upload your data.
4. As for your data, if we fail to agree, we will try to sell personal information/trade secrets/-databases/source codes - generally speaking, everything that has a value on the darkmarket - to multiple threat actors at once. Then all of this will be published in our blog - https://-akiral2iz6a7qgd3ayp3l6yub7xx2uep76idk3u2kollpj5z3z636bad.onion.
5. We're more than negotiable and will definitely find the way to settle this quickly and reach an agreement which will satisfy both of us.

If you're indeed interested in our assistance and the services we provide you can reach out to us following simple instructions:

1. Install TOR Browser to get access to our chat room - https://www.torproject.org/download/.
2. Paste this link - https://akiralkzxzq2dsrzsrvbr2xgbbu2wgsmxryd4csgfameg52n7efvr2id.onion.
3. Use this code - [redacted] - to log into our chat.

Keep in mind that the faster you will get in touch, the less damage we cause.

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.

```
pwndbg> run
Starting program: /root/bcae.elf -p=/root/[redacted]/
Detected number of cpus = 2
[New LWP 2668616]
[New LWP 2668617]
[New LWP 2668618]
[New LWP 2668619]
[LWP 2668617 exited]
[LWP 2668616 exited]
[LWP 2668618 exited]
9ms
[LWP 2668612 exited]
[Inferior 1 (process 2668612) exited normally]
```

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.

```
pwndbg> info threads
  Id   Target Id         Frame
* 1    LWP 2668801 "bcae.elf" 0x000000000046dcf7 in ?? ()
  2    LWP 2668802 "bcae.elf" 0x0000000000046fb36 in ?? () 
  3    LWP 2668803 "bcae.elf" 0x0000000000046fb36 in ?? ()
  4    LWP 2668804 "bcae.elf" 0x000000000046970b in ?? ()
  5    LWP 2668805 "bcae.elf" 0x00000000004695d8 in ?? ()
pwndbg> thread 2
[Switching to thread 2 (LWP 2668802)]
#0  0x0000000000046fb36 in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA

 RAX  0xffffffffffffffe00
*RBX  0x0
*RCX  0x46fb36 <- cmp    rax, -0x1000 /* 'H=' */
*RDX  0x0
*RDI  0x6d0ccc <- 0x0
*RSI  0x80
 R8   0x0
*R9   0x4
 R10  0x0
*R11  0x282
*R12  0x6d0cc4 <- 0x8
*R13  0x6d0c50 <- 0x0
*R14  0x7ffff7ff74d0 -> 0x46f810 <- endbr64
*R15  0x6d0ccc <- 0x0
*RBP  0x6d0ca0 <- 0x3ba5
*RSP  0x7fffff7ff7490 <- 0x1dd1
*RIP  0x46fb36 <- cmp    rax, -0x1000 /* 'H=' */
```

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.

```
aMntZMancoWorkG_8 db '/mnt/z/manco_work/gittedprojects/esxi6_5/cryptolib/nettle_rsa/aes'
                                        ; DATA XREF: sub_467D4F+34↑o
                  db '256/aes256-encrypt.c',0
aLengthAesBlock_0 db '!(length % AES_BLOCK_SIZE)',0
                                        ; DATA XREF: sub_467D4F+3B↑o
                  align 20h
aNettleAes256En   db 'nettle_aes256_encrypt',0
                                        ; DATA XREF: sub_467D4F+28↑o
                  align 20h
aMntZMancoWorkG_9 db '/mnt/z/manco_work/gittedprojects/esxi6_5/cryptolib/nettle_rsa/der'
                                        ; DATA XREF: sub_468110+30↑o
                                        ; sub_4681B4+2C↑o
                  db '/der-iterator.c',0
                  align 8
aITypeAsn1TypeC   db 'i->type & ASN1_TYPE_CONSTRUCTED',0
                                        ; DATA XREF: sub_468110+37↑o
aITypeAsn1Bitst   db 'i->type == ASN1_BITSTRING',0
                                        ; DATA XREF: sub_4681B4+33↑o
                  align 20h
aNettleAsn1DerD   db 'nettle_asn1_der_decode_constructed',0
                                        ; DATA XREF: sub_468110+24↑o
                  align 20h
aNettleAsn1DerD_0 db 'nettle_asn1_der_decode_bitstring',0
                                        ; DATA XREF: sub_4681B4+20↑o
                  align 10h
aMntZMancoWorkG_10 db '/mnt/z/manco_work/gittedprojects/esxi6_5/cryptolib/nettle_rsa/pkc'
                                        ; DATA XREF: sub_468667+5F↑o
                  db 's1/pkcs1-encrypt.c',0
aPadding8         db 'padding >= 8',0   ; DATA XREF: sub_468667+66↑o
                  align 20h
aNettlePkcs1Enc   db 'nettle_pkcs1_encrypt',0
                                        ; DATA XREF: sub_468667+53↑o
                  align 20h
aMntZMancoWorkG_11 db '/mnt/z/manco_work/gittedprojects/esxi6_5/cryptolib/nettle_rsa/aes'
                                        ; DATA XREF: sub_469CCF+3F↑o
                  db '256/aes-encrypt-internal.c',0
aLength16         db '!((length) % (16))',0
                                        ; DATA XREF: sub_469CCF+46↑o
                  align 10h
aNettleAesEncry   db '_nettle_aes_encrypt',0
```

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

```
test.akira

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00004270  1A 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00004280  01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00004290  6F 5B 4C 6D 65 8D EB BB 12 84 87 38 A6 86 80 51  o[Lme.ë»..‡‡8¦.€Q
000042A0  1F 18 1F CA 1F 0E F0 0B 71 D7 33 D2 08 1F 43 D6  ...Ê..ð.q×3Ò..CÖ
000042B0  07 FE 18 BC C9 81 80 DA E4 84 61 A0 DF 7A 00 5E  .þ.¼É.€Úä„a ßz.^
000042C0  F2 F0 0D 38 31 53 7A F2 51 0F 1B D2 E7 CF F0 F3  òð.81Szò Q..ÒçÏðó
000042D0  F1 C9 E7 E2 09 59 99 08 19 28 1A 95 2C 70 16 DA  ñÉçâ.Y™..(.•,p.Ú
000042E0  08 55 2A B9 92 97 5D EA F1 E4 CF 0E 72 B9 6E 8A  .U*'’—]êñäÏ.r¹nŠ
000042F0  D2 F7 4E BB 02 03 A5 41 88 DD 41 28 CC 70 1A C8  Ò÷N».¥A^ÝA(Ìp.È
00004300  CF 77 5F 11 D6 1A 60 0E 06 E2 57 D6 6D 61 A2 76  Ïw_.Ö.`..âWÖmaev
00004310  71 38 93 27 28 22 E6 AA A3 06 F6 AF 6B C4 8A 61  q8“'(“æª£.ö¯kÄŠa
00004320  17 15 33 61 57 4B 52 F6 19 AE 02 56 2C A1 70 93  ..3aWKRö.®.V,¡p“
00004330  07 91 E1 81 A0 9A 9C 5A 8A F8 76 6D C9 32 8D D8  .‘á. šœZŠøvmÉ2.Ø
00004340  39 29 16 D6 E7 E0 A2 47 0B 0D 65 15 D0 F5 04 60  9).ÖçàcG..e.Ðõ.`
00004350  36 23 94 C1 8F 0B 51 CE 8D 09 6D 28 66 BF 8B 7D  6#“Á..QÎ..m(f¿‹}
00004360  A0 D5 8F 69 44 E8 33 23 EF 72 E4 B7 75 AD D9 CE   Õ.iDè3#irä·u.ÙÎ
00004370  94 9A 5F 3B 97 4F 0E 8F 13 1C 74 60 94 7D 7F CE  “š_;—O....t`”}.Î
00004380  AA 95 4E 5D 00 FF 91 EE 6B 30 7D 89 69 62 E4 2D  ª•N].ÿ‘îk0}‰ibä-
00004390  F7 97 B9 8B 17 CF 33 59 FD EC 7F 3C 0B 35 CA 99  ÷—¹‹.Ï3Yýì.<.5Ê™
000043A0  05 80 54 65 22 13 A2 B6 B0 12 02 2E 23 EF F9 C3  .€Te“.¢¶°...#ïùÃ
000043B0  29 55 58 37 46 B8 44 15 6A D2 B4 FA B3 58 D0 60  )UX7F¸D.jÒ´ú³XÐ`
000043C0  A2 45 67 56 9D F4 0C 86 E5 E9 FD 07 B7 39 F4 C9  ¢EgV.ô.†åéý.·9ôÉ
000043D0  A1 0B 15 16 96 85 03 F7 61 6C EF 35 4D A2 F1 25  ¡...–….÷alï5M¢ñ%
000043E0  B8 4B 56 B6 A0 CE A8 26 36 CB 3A 3E C1 34 1B 11  ¸KV¶ Î¨&6Ë:>Á4..
000043F0  D1 67 C7 71 E1 0C 7C 14 9B 72 74 BF FF FE B7 A5  ÑgÇqá.|.›rt¿ÿþ·¥
00004400  23 AD 25 5E CC FE 1E 26 30 4C 79 07 08 E3 7C 51  #.%^Ìþ.&0Ly..ã|Q
00004410  AE 44 D7 E5 1E C2 60 FE 39 98 CF 32 D4 6A 89 86  ®D×å.Â`þ9˜Ï2Ôj‰†
00004420  84 CD CD B3 9A B0 62 E3 F1 F4 F5 CC 9B 0B AA 38  „ÍÍ³šºbãñôõÌ›.ª8
00004430  AD 4B 65 E4 E4 2C 2B 2D C0 18 E8 47 1A 0A CA 33  .Keää,+-À.èG..Ê3
00004440  CB 6F EE 28 8F 57 BF CD 19 4C 51 35 B2 7B 61 F8  Ëoî(.W¿Í.LQ5²{aø
00004450  7E AF CE 38 93 73 0D FF 83 9F 97 5E FA 98 3D DA  ~¯Î8“s.ÿƒŸ—^ú˜=Ú
00004460  33 57 82 4B 95 9C 38 F6 56 B8 A0 2A DE 58 63 F1  3W‚K•œ8öV¸ *ÞXcñ
00004470  21 DD 27 5C 7D 8B 83 62 68 31 9C F2 40 BF F5 67  !Ý'\}‹ƒbh1œò@¿õg
00004480  0A 57 60 93 EA 4E 32 06 03 5C 92 51 7F AF 72 20  .W`“êN2..\’Q.¯r
```

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

Akira Analysis Briefing	
Analyst	Sooker(@4fdd)
Background	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 face 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.
Sample	md5 6B0583103CBED4A0A5829B63D16936ED3 Sha256 BCAF97BC178CD0C08F64194AF97BF3471197801C36F75CFF2FC8BCFCBF3DBBD1A
OS	Linux
Name	Akira (AMD64, 64-bit)
Type	Ransomware, File-Encrypting virus
Algorithm	AES + RSA (Nettle cryptographic library)
Extension	.akira
Threat Level	High
Ransomw Note	akira_readme.txt
Victim	Publish victims information address: hxxps[:]//akira42le6a7qgd3ayp3l6ruls7xx2wep76idk5o2kolfpj5z5z6l36bxd[.]onion
Contact	1. Onion address: hxxps[:]//akirafkzxzg2dvzcnvbr2xgbbu2wgsmxryd4csgfameg52n7efvr2id[.]onion 2. Unique code for logging to chat: xxxx-xx-xxx-xxxx to chat
Suspicious strings	akira_readme.txt, _nettle_aes256_encrypt , .akira /mnt/d/manco_work/gitedprojects/esxi6_5/cryptolib/nettle_nar/pkcs1/pkcs1-encrypt.c Detected number of cpus =
Key Informations	./bcae.elf -p= ./ 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. -m(--encryption_percent) to encrypt with percent, such as to set -m=5, -m=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
Infected and Damaged	stack005.c.akira 0x08048809c B3ec0c sub esp, 0xc,akira 0x08048897 e8B4fcffff call sym.login.akira akira_readme.txt

End.

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