Harder, Better, Faster, Locker: Ransomware Groups Flex on Defenders

Recorded Future®

Introduction

Lindsay Kaye
Senior Director, ARMOR, Insikt Group
Recorded Future

Being Evil is Hard Work

Every ransomware group is subject to the realities of the economy - and many innovate their tools or behavior in order to remain successful in a competitive market. You can’t simply “build it and they will come”.

But, like “New Coke”, sometimes these changes don’t quite work out the way the threat actors intended and can occasionally backfire. Today, we’ll tell you about some of the “innovations” we’ve observed in ransomware, and talk about what made them a feather in the group’s cap, or a flop

The Dark Web is Not a Vacuum

World events directly impact the dark web ecosystem - changes in TTPs, threat actor behavior and even new “professions” have emerged over the past several years

The COVID-19 pandemic

- Initial Access Brokers took advantage of home/work laptop use
- Pulse, Fortinet VPN, Citrix ADC vulnerability exploitation increased

Russia/Ukraine War

- Conti Leaks
- “Brain drain” of technical talent fleeing the country
- Losing “top cover” from Russian state

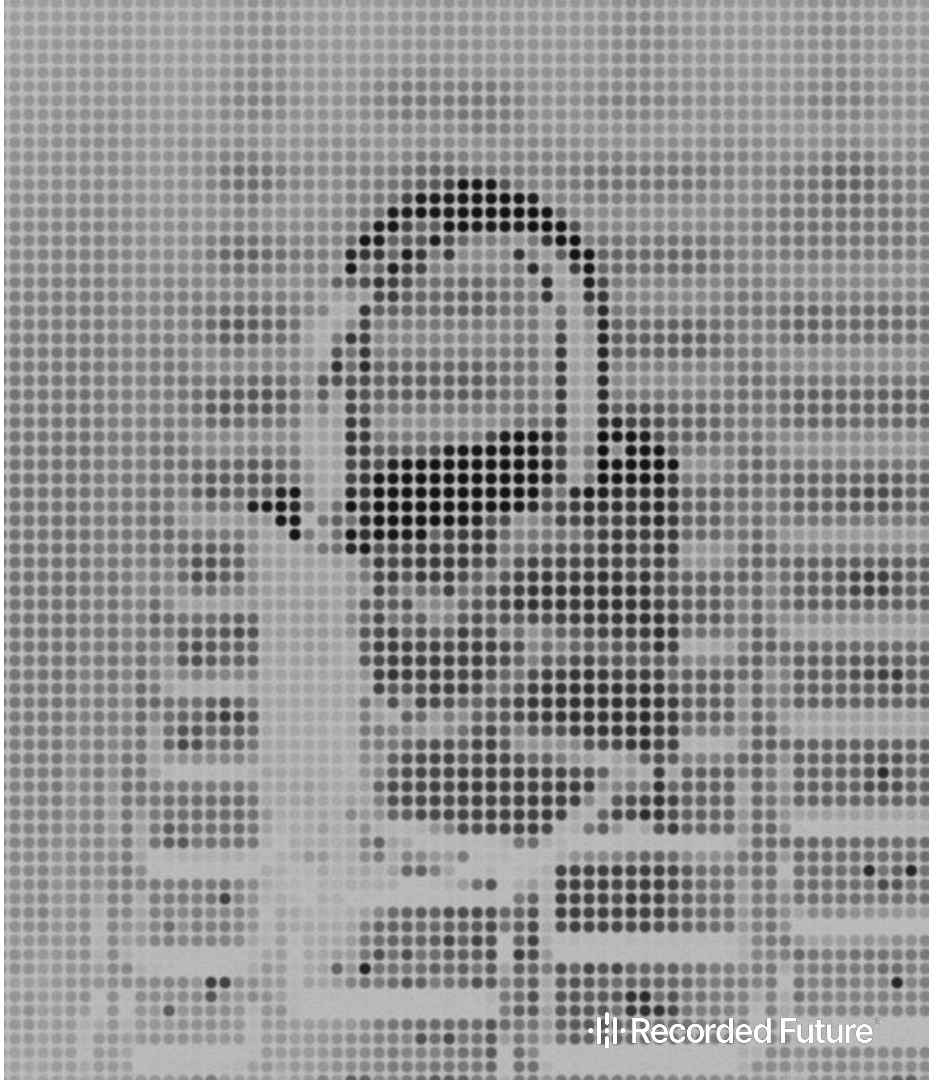
Law enforcement takes notice of high-profile ransomware attacks

- Colonial pipeline
- JBS foods

Evolution of Tools

Over 2022 (and into 2023!) we observed several types of changes in lockers and ransomware threat actor TTPs for a variety of threat actors

- New lockers!
- Feature additions to existing lockers
- New ways of deploying, spreading lockers
- Not just C/C++ anymore: Golang, Rust, Python
- Additional extortion methods
- Targeting additional hardware



What Worked Well

Deployment improvements

- Using MSPs and “benign” tools to spread (Kaseya Incident)
- Move laterally using SMB shares, CIFS, NFS
- Impersonation Tokens built into the locker

Optimizing current offerings

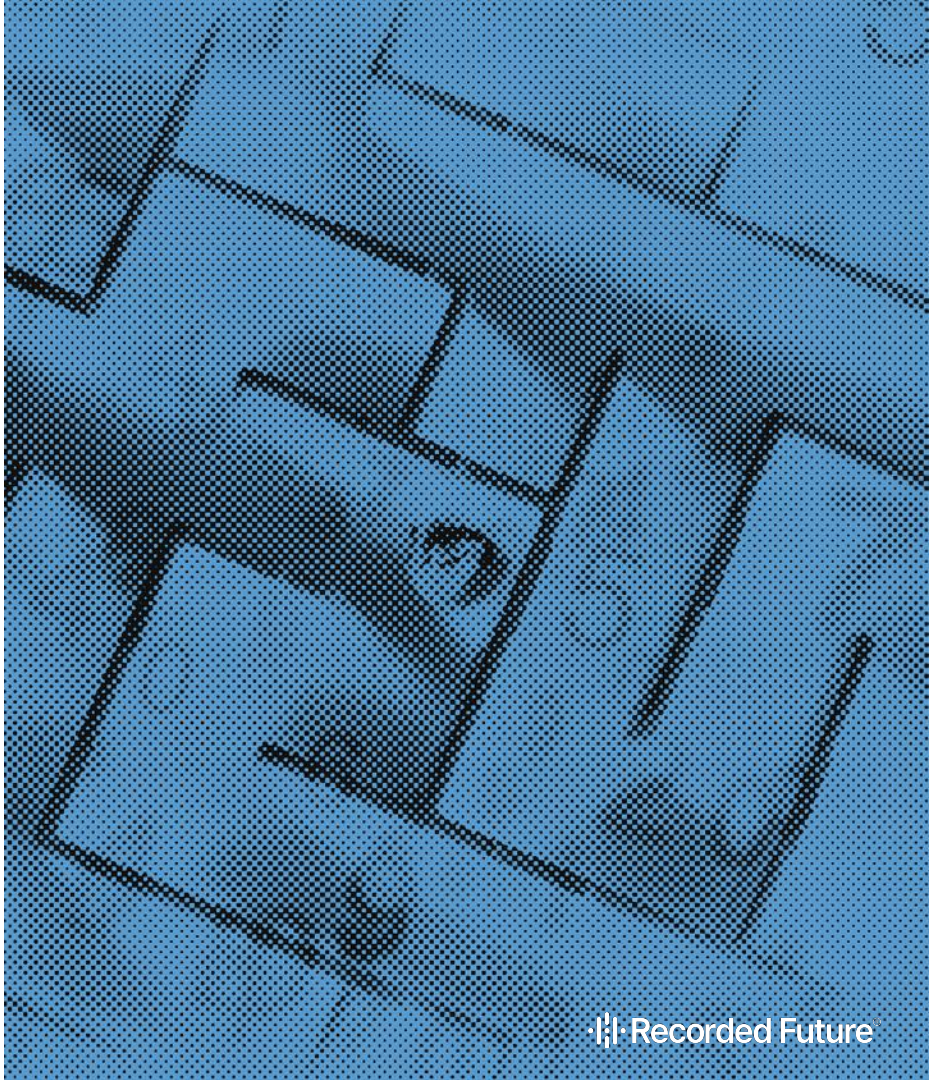
- Intermittent encryption makes lockers faster
- Adding functionality to lockers

Filling “gaps” in the ecosystem

- BlackMatter and Conti’s Linux/ESXi lockers after REvil, Darkside disbanded
- ALPHV’s addition of chat access codes
- ALPHV’s victim files index site
- Make panels more user-friendly (adding BTC mixers, support tickets, moderating victims)

Really try to make ‘em pay

- Additional Extortion Techniques (DDoS, Calling Board Members, Contacting Media)
- Printing Ransomware notes to physical printers



And what left something to be desired…

Roll your own crypto

- DarkSide, BlackMatter both had encryption flaws

False flag attribution

- Lockbit Recorded Future interview
- Xing, Shao ransomware
- Russian strings in “Chinese” ransomware
- Machine Translation Forum Posts

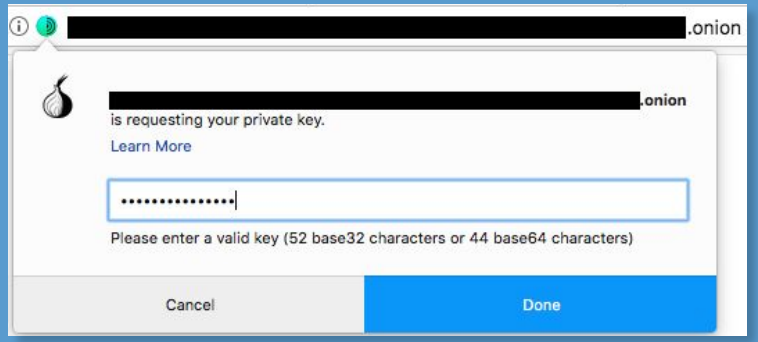
Making tools more signaturable

- ALPHV Morph Linux edition
- LockBit Black (and everyone else) using BlackMatter’s code
- Automated obfuscations like PLAY, ALPHV

Making it so secure no one can use it

Letting politics get involved

- Conti sides with Russia in RU/UA Invasion



Recorded Future®

Image Attribution: Warner Music Group

PLAY Ransomware

PLAY ransomware is a relatively new, but fairly active ransomware variant first observed in June 2022

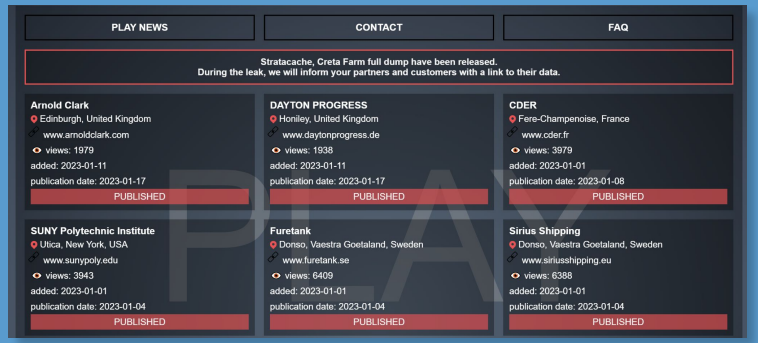
Written in C++

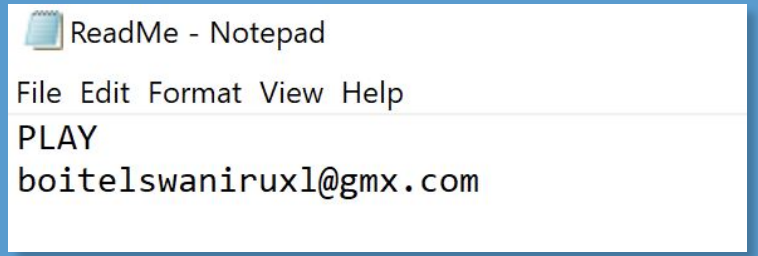
First used against Argentina Court of Cordoba in August 2022

- RackSpace
- City of Antwerp
- H-Hotels

Notable Features

- Minimal ransomware note
- No ROP to ROP, other added obfuscations
- Intermittent Encryption





ROP in Ransomware??

Increment ESP by 0×32 and RET causes a “jump” to real code

```
local_8 = DAT_0041e004 ^ (uint)&stack0xfffffffc;
uVar5 = 0x104;
local_624 = 0;
vol = do_FindFirstVolumeW_z(drive,0x104);
uVar4 = (undefined)uVar5;
uVar2 = extraout_DL;
if (vol != (void *)0xffffffff) {
  do {
    drivetype = do_GetDriveTypeW_z(drive);
    if (((drivetype != 5) && (drivetype != 6)) &&
       (do_GetVolumePathNamesForVolumeNameW_z(drive,local_620,0x208,&local_624), local_624 < 2)) {
      uVar3 = check_if_disk_has_free_space_z(drive,0);
      if (((int)((ulonglong)uVar3 >> 0x20) != 0) || (0x40000000 < (uint)uVar3)) {
        creates_directory_in_temp_sets_volmountpoint_z(drive);
      }
    }
    iVar1 = do_FindNextVolumeW_z(vol,drive,0x104);
    uVar4 = (undefined)uVar5;
  } while (iVar1 != 0);
  do_FindVolumeClose_z(vol);
  uVar2 = extraout_DL_00;
}
```

Earlier PLAY code without ROP

(Source: Recorded Future)

```
                    SUB_004177f3                                    XREF[1]:    004177e9(c)
004177f3 83 04 24 32        ADD     dword ptr [ESP],0x32
004177f7 c3                 RET
004177f8 7f 10 c9 7c        ddw     7CC9107Fh
004177fc 09 b5 bf b0        ddw     B0BFB509h
00417800 bc 8a d4 fc        ddw     FCD48ABCh
00417804 bc 2d 90 c8        ddw     C8902DBCh
00417808 b6 35 87 6e        ddw     6E8735B6h
0041780c 7c 3a 02 30        ddw     30023A7Ch
00417810 be 7f 00 6b        ddw     6B007FBEh
00417814 8e 34 8b 7b        ddw     7B8B348Eh
00417818 26 70 d5 b3        ddw     B3D57026h
0041781c a3 c8 7a 56        ddw     567AC8A3h
00417820 8b 0d b0 06        ddw     6B00D8Bh
00417824 43                 INC     EBX
00417825 00 e8              ADD     AL,CH
00417827 c5 5c ff ff        LDS     EBX,[EDI + EDI*0x8 + -0x1]
0041782b 68 58 06 43 00     PUSH    DAT_00430658
00417830 ff d0              CALL    EAX
00417832 8b 73 08           MOV     ESI,dword ptr [EBX + 0x8]
00417835 ba 4c cb 42 00     MOV     EDX=>s_THREAD:_0042cb4c,s_THREAD:_0042cb4c
0041783a ff 76 10           PUSH    dword ptr [ESI + 0x10]
0041783d 8b 0e              MOV     ECX,dword ptr [ESI]
0041783f e8 8c 7a ff ff     CALL    do_log_z
00417844 83 c4 04           ADD     ESP,0x4
00417847 8b ce              MOV     ECX,ESI
00417849 e8 82 03 00        ddw     382E8h
0041784d 00 b9 58 06 43 00  ADD     byte ptr [ECX + DAT_00430658],BH
00417853 e8 28 71 ff ff     CALL    FUN_0040e980
00417858 8b 45 f4           MOV     EAX,dword ptr [EBP + -0xc]
0041785b 5f                 POP     EDI
0041785c 5e                 POP     ESI
0041785d 8b e5              MOV     ESP,EBP
0041785f 5d                 POP     EBP
00417860 8b e3              MOV     ESP,EBX
00417862 5b                 POP     EBX
00417863 c2 04 00           RET     0x4
00417866 cc                 ??      CCh
00417867 cc                 ??      CCh
```

“Garbage” code bytes

Newer PLAY code using ROP

(Source: Recorded Future)

Recorded Future

Rule 1: Adding Obfuscation is Good, But Consider it From the Start

First PLAY sample observed in mid-June 2022

- String obfuscation
- API hashing technique
- Fairly easy to reverse engineer

Additional obfuscations first observed in early August 2022

- Return-oriented programming (ROP)
- Garbage code insertion

ROP is a positive addition to make the code harder to RE, however:

- Underlying functionality did not change
- Automated garbage code addition is somewhat signaturable

```
void __cdecl do_string_decrypt_z(char *in,uint size,char xorkey [8],char *out)

{
  uint uVar1;
  byte ctr2;
  uint uVar2;
  int inner_ctr;
  uint ctr;

  for (ctr = 0; ctr < size; ctr = ctr + 1) {
    out[ctr] = in[ctr];
    for (inner_ctr = 0; inner_ctr < 8; inner_ctr = inner_ctr + 2) {
      ctr2 = (byte)inner_ctr;
      uVar1 = (int)out[ctr] >> (ctr2 & 0x1f) & 1;
      uVar2 = (int)out[ctr] >> (ctr2 + 1 & 0x1f) & 1;
      if (uVar1 != uVar2) {
        if (uVar1 == 0) {
          out[ctr] = out[ctr] & ~(byte)(1 << (ctr2 + 1 & 0x1f));
        }
        else {
          out[ctr] = out[ctr] | (byte)(1 << (ctr2 + 1 & 0x1f));
        }
        if (uVar2 == 0) {
          out[ctr] = out[ctr] & ~(byte)(1 << (ctr2 & 0x1f));
        }
        else {
          out[ctr] = out[ctr] | (byte)(1 << (ctr2 & 0x1f));
        }
      }
    }
    out[ctr] = ~out[ctr];
    out[ctr] = out[ctr] ^ xorkey[ctr % 8];
  }
  return;
}
```

```
004161c2 a1 44 c9 42 00        MOV     EAX,[DAT_0042c944]
004161c7 89 85 14 ff ff ff     MOV     dword ptr [EBP + 0xffffff14]=>local_f0,EAX
004161cd 0f b7 05 48 c9 42 00  MOVZX   EAX,word ptr [DAT_0042c948]
004161d4 0f 11 85 58 fd ff ff  MOVUPS  xmmword ptr [EBP + 0xfffffd58]=>local_2ac[0],XMM0
004161db 66 89 85 18 ff ff ff  MOV     word ptr [EBP + 0xffffff18]=>local_ec,AX
004161e2 0f 10 05 fc c8 42 00  MOVUPS  XMM0,xmmword ptr [u_lodging_0042c8fc]
004161e9 a1 4c c9 42 00        MOV     EAX,[s_sensor_0042c94c]
004161ee 89 85 2c ff ff ff     MOV     dword ptr [EBP + 0xffffff2c]=>local_d8,EAX
004161f4 0f b7 05 50 c9 42 00  MOVZX   EAX,word ptr [s_or_0042c94c+4]
004161fb 0f 11 85 80 fd ff ff  MOVUPS  xmmword ptr [EBP + 0xfffffd80]=>local_284[0],XMM0
00416202 66 89 85 30 ff ff ff  MOV     word ptr [EBP + 0xffffff30]=>local_d4,AX
00416209 f3 0f 7e 05 0c c9 42 00  MOVQ  XMM0,qword ptr [u_movers_0042c90c]
00416211 0f b6 05 52 c9 42 00  MOVZX   EAX,byte ptr [s__0042c94c+6]
00416218 66 0f d6 85 1c ff ff ff  MOVQ  qword ptr [EBP + 0xffffff1c]=>local_e8,XMM0
00416220 f3 0f 7e 05 1c c9 42 00  MOVQ  XMM0,qword ptr [s_GETMQCHMk_0042c91c]
00416228 88 85 32 ff ff ff     MOV     byte ptr [EBP + 0xffffff32]=>local_d2,AL
0041622e 0f b7 05 5c c9 42 00  MOVZX   EAX,word ptr [DAT_0042c95c]
00416235 66 0f d6 85 8c fe ff ff  MOVQ  qword ptr [EBP + 0xfffffe8c]=>local_178,XMM0
0041623d f3 0f 7e 05 30 c9 42 00  MOVQ  XMM0,qword ptr [DAT_0042c930]
00416245 66 89 85 54 ff ff ff  MOV     word ptr [EBP + 0xffffff54]=>local_b8,AX
0041624c 0f b6 05 5e c9 42 00  MOVZX   EAX,byte ptr [DAT_0042c95e]
00416253 66 0f d6 85 64 ff ff ff  MOVQ  qword ptr [EBP + 0xffffff64]=>local_a8,XMM0
0041625b f3 0f 7e 05 3c c9 42 00  MOVQ  XMM0,qword ptr [DAT_0042c93c]
00416263 88 85 56 ff ff ff     MOV     byte ptr [EBP + 0xffffff56]=>local_ae,AL
00416269 b0 d2                 MOV     AL,0xd2
0041626b 66 0f d6 85 0c ff ff ff  MOVQ  qword ptr [EBP + 0xffffff0c]=>local_f8,XMM0
00416273 f3 0f 7e 05 54 c9 42 00  MOVQ  XMM0,qword ptr [DAT_0042c954]
0041627b 88 45 fa              MOV     byte ptr [EBP + -0x6]=>local_a,AL
0041627e b0 fa                 MOV     AL,0xfa
00416280 66 0f d6 85 4c ff ff ff  MOVQ  qword ptr [EBP + 0xffffff4c]=>local_b8,XMM0
00416288 f3 0f 7e 05 60 c9 42 00  MOVQ  XMM0,qword ptr [s_document_0042c960]
00416290 88 45 fb              MOV     byte ptr [EBP + -0x5]=>local_9,AL
00416293 0f b6 05 68 c9 42 00  MOVZX   EAX,byte ptr [s__0042c960+8]
0041629a 66 0f d6 45 d0        MOVQ    qword ptr [EBP + -0x30]=>local_34,XMM0
0041629f f3 0f 7e 05 6c c9 42 00  MOVQ  XMM0,qword ptr [DAT_0042c96c]
004162a7 88 45 d8              MOV     byte ptr [EBP + -0x28]=>local_2c,AL
004162aa a1 74 c9 42 00        MOV     EAX,[DAT_0042c974]
004162af 66 0f d6 85 fc fe ff ff  MOVQ  qword ptr [EBP + 0xfffffefc]=>local_108,XMM0
004162b7 f3 0f 7e 05 7c c9 42 00  MOVQ  XMM0,qword ptr [DAT_0042c97c]
004162bf 89 85 04 ff ff ff     MOV     dword ptr [EBP + 0xffffff04]=>local_100,EAX
004162c5 0f b7 05 78 c9 42 00  MOVZX   EAX,word ptr [DAT_0042c978]
```

Rule 2: Give the People What They Want

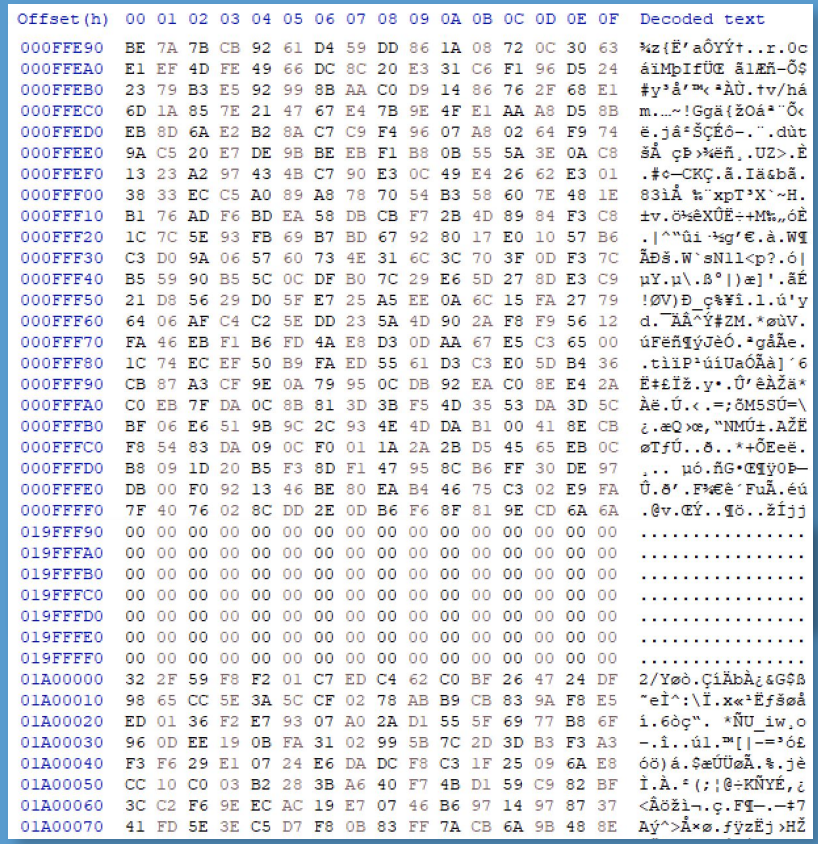
PLAY ransomware uses intermittent encryption

- Encrypts every other 1MB of data
- Feature included from the start in June 2022

Makes encryption faster over large files

- Less “recoverable” than just encrypting first X bytes
- Faster = more damage = profit!

Implemented by other groups like Agenda, ALPHV and BlackBasta



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000FFE90	BE	7A	7B	CB	92	61	D4	59	DD	86	1A	08	72	0C	30	63	¾z{Ë'aÔYÝ†..r.0c
000FFEA0	E1	EF	4D	FE	49	66	DC	8C	20	E3	31	C6	F1	96	D5	24	áïMþIfÜŒ ã1Æñ–Õ$
000FFEB0	23	79	B3	E5	92	99	8B	AA	C0	D9	14	86	76	2F	68	E1	#y³å'™‹ªÀÙ.†v/há
000FFEC0	6D	1A	85	7E	21	47	67	E4	7B	9E	4F	E1	AA	A8	D5	8B	m.…~!Ggä{žOáª¨Õ‹
000FFED0	EB	8D	6A	E2	B2	8A	C7	C9	F4	96	07	A8	02	64	F9	74	ë.jâ²ŠÇÉô–.¨.dùt
000FFEE0	9A	C5	20	E7	DE	9B	BE	EB	F1	B8	0B	55	5A	3E	0A	C8	šÅ çÞ›¾ëñ¸.UZ>.È
000FFEF0	13	23	A2	97	43	4B	C7	90	E3	0C	49	E4	26	62	E3	01	.#¢—CKÇ.ã.Iä&bã.
000FFF00	38	33	EC	C5	A0	89	A8	78	70	54	B3	58	60	7E	48	1E	83ìÅ ‰¨xpT³X`~H.
000FFF10	B1	76	AD	F6	BD	EA	58	DB	CB	F7	2B	4D	89	84	F3	C8	±v.ö½êXÛË÷+M‰„óÈ
000FFF20	1C	7C	5E	93	FB	69	B7	BD	67	92	80	17	E0	10	57	B6	.|^"ûi·½g'€.à.W¶
000FFF30	C3	D0	9A	06	57	60	73	4E	31	6C	3C	70	3F	0D	F3	7C	ÃÐš.W`sN1l<p?.ó|
000FFF40	B5	59	90	B5	5C	0C	DF	B0	7C	29	E6	5D	27	8D	E3	C9	µY.µ\.ß°|)æ]'.ãÉ
000FFF50	21	D8	56	29	D0	5F	E7	25	A5	EE	0A	6C	15	FA	27	79	!ØV)Ð_ç%¥î.l.ú'y
000FFF60	64	06	AF	C4	C2	5E	DD	23	5A	4D	90	2A	F8	F9	56	12	d.¯ÄÂ^Ý#ZM.*øùV.
000FFF70	FA	46	EB	F1	B6	FD	4A	E8	D3	0D	AA	67	E5	C3	65	00	úFëñ¶ýJèÓ.ªgåÃe.
000FFF80	1C	74	EC	EF	50	B9	FA	ED	55	61	D3	C3	E0	5D	B4	36	.tìïP¹úíUaÓÃà]´6
000FFF90	CB	87	A3	CF	9E	0A	79	95	0C	DB	92	EA	C0	8E	E4	2A	Ë‡£Ïž.y•.Û'êÀŽä*
000FFFA0	C0	EB	7F	DA	0C	8B	81	3D	3B	F5	4D	35	53	DA	3D	5C	Àë.Ú.‹.=;õM5SÚ=\
000FFFB0	BF	06	E6	51	9B	9C	2C	93	4E	4D	DA	B1	00	41	8E	CB	¿.æQ›œ,"NMÚ±.AŽË
000FFFC0	F8	54	83	DA	09	0C	F0	01	1A	2A	2B	D5	45	65	EB	0C	øTƒÚ..ð..*+ÕEeë.
000FFFD0	B8	09	1D	20	B5	F3	8D	F1	47	95	8C	B6	FF	30	DE	97	¸.. µó.ñG•Œ¶ÿ0Þ—
000FFFE0	DB	00	F0	92	13	46	BE	80	EA	B4	46	75	C3	02	E9	FA	Û.ð'.F¾€ê´FuÃ.éú
000FFFF0	7F	40	76	02	8C	DD	2E	0D	B6	F6	8F	81	9E	CD	6A	6A	.@v.ŒÝ..¶ö..žÍjj
019FFF90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
019FFFA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
019FFFB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
019FFFC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
019FFFD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
019FFFE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
019FFFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
01A00000	32	2F	59	F8	F2	01	C7	ED	C4	62	C0	BF	26	47	24	DF	2/Yøò.ÇíÄbÀ¿&G$ß
01A00010	98	65	CC	5E	3A	5C	CF	02	78	AB	B9	CB	83	9A	F8	E5	˜eÌ^:\Ï.x«¹Ëƒšøå
01A00020	ED	01	36	F2	E7	93	07	A0	2A	D1	55	5F	69	77	B8	6F	í.6òç“. *ÑU_iw¸o
01A00030	96	0D	EE	19	0B	FA	31	02	99	5B	7C	2D	3D	B3	F3	A3	–.î..ú1.™[|-=³ó£
01A00040	F3	F6	29	E1	07	24	E6	DA	DC	F8	C3	1F	25	09	6A	E8	óö)á.$æÚÜøÃ.%.jè
01A00050	CC	10	C0	03	B2	28	3B	A6	40	F7	4B	D1	59	C9	82	BF	Ì.À.²(;¦@÷KÑYÉ‚¿
01A00060	3C	C2	F6	9E	EC	AC	19	E7	07	46	B6	97	14	97	87	37	<Âöžì¬.ç.F¶—.—‡7
01A00070	41	FD	5E	3E	C5	D7	F8	0B	83	FF	7A	CB	6A	9B	48	8E	Aý^>Å×ø.ƒÿzËj›HŽ

Recorded Future

ALPHV Ransomware

ALPHV ransomware is a rebrand of BlackMatter

Written in Rust

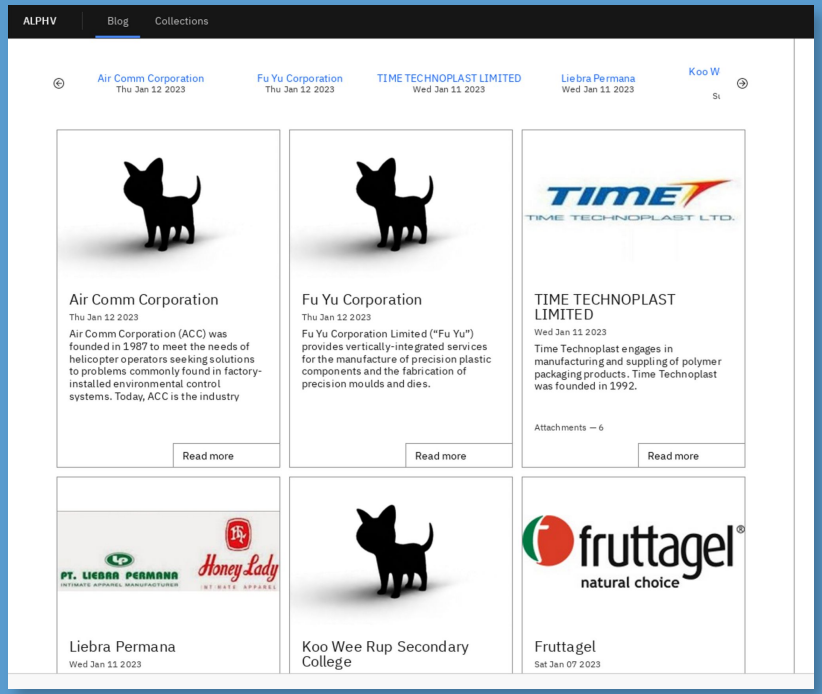
First discovered in December 2021

Multiple attacks on infrastructure

- Colonial Pipeline (as DarkSide)
- Creos Luxembourg European gas pipeline
- Italy's energy agency GSE
- Colombian energy company EPM

Notable Features

- First “big” ransomware in Rust
- First “big” ransomware with ARM locker
- Build-time obfuscation toolkit “MORPH”
- Chat Access Tokens
- ALPHV Collections



Recorded Future

ALPHV Access Tokens

Chat Hijacking (as BlackMatter)

Requirements for victim chat

- Domain Controllers
- Domain Admins

Locker parameter “access-token” required

Created more secure line of communication with victims

- Unable to discover victim page through sample detonation

DS: Why did you add Access tokens and unique domains for every victim?

ALPHV: As adverts of darkmatter [DarkSide / BlackMatter], we suffered from the interception of victims for subsequent decryption by Emsisoft.

ALPHV Interview **(Source: Recorded Future)**

SOFTWARE

The software is written from scratch without using any templates or previously leaked source codes of other ransomware. The choice is offered:

4 encryption modes:
-Full - full file encryption. The safest and the slowest.
-Fast encryption of the first N megabytes. It is not recommended for use, the most insecure of possible solutions, but the fastest.
-Dotpattern - encryption of N megabytes through M step. If configured incorrectly, Fast may work worse both in terms of speed and cryptographic strength.
-Smartpattern - encryption of N megabytes in percentage increments. By default, it encrypts with a 10 megabyte strip every 10% of the file starting from the header. The most optimal mode in the ratio of speed \ cryptographic strength.

2 encryption algorithms:
ChaCha20 and AES
In auto mode, the software detects the presence of hardware support for AES (exists in all modern processors) and uses it. If there is no AES support, the software encrypts ChaCha20 files.
The software is cross-platform, i.e. if you mount Windows disks on Linux or vice versa, the decryptor will be able to decrypt files.

Supported OS:
- The entire line of Windows from 7 and above (tested by us on 7, 8.1, 10, 11; 2008r2, 2012, 2016, 2019, 2022); XP and 2003 can be encrypted by SMB.
- ESXI (tested on 5.5, 6.5, 7.0.2u)
- Debian (tested on 7, 8, 9);
- Ubuntu (tested on 18.04, 20.04)
- ReadyNAS, Synology

Since binaries have been leaking to analysts lately, and premium VT allows you to download samples and get README random people may appear in chats who can disrupt negotiations (hello DarkSide), it is MANDATORY to use the --access-token flag when launching the software. Cmdline arguments are not passed to the AntiVirus, which will allow maintaining the secrecy of correspondence with the victim. For the same reason, each encrypted computer generates its own unique ID used to separate chats.

ALPHV Affiliate Introduction **(Source: Recorded Future)**

Why Rust?

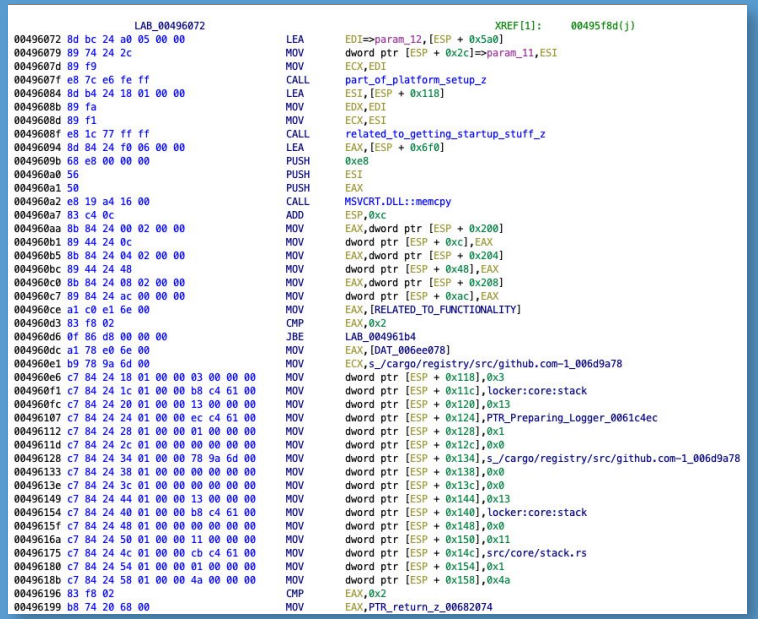
First “big” ransomware written in Rust

- Usually C/C++, Delphi, Golang
- FickerStealer also written in Rust

Cross-compilable to several architectures - get a Windows, Linux, ARM locker from one set of code

Bonus: Reverse engineering is harder (for now)

- Library functions not always identified - look like non-library, interesting code
- Lots of runtime code in the binary (eg: error handling)
- Strings are not null-terminated =(
- “Fixup” tools more nascent, currently



Part of Rust code from ALPHV Windows binary

An ARM locker, you say?

Advertised as being designed to target NAS devices (QNAP, Synology, and more)

- Used in parallel to Windows and Linux/ESXi lockers
- Backups and uncommon file shares
- Increase effectiveness of ransom attack

To date, have not observed ITW use

Not common to see - Chaos ransomware also has one, but no other mainstream groups

```
0c 30 84 e2     add     r3,r4,#0xc
05 00 a0 e1     cpy     r0,r5
0c 20 a0 e3     mov     r2,#0xc
01 10 8f e0     add     r1=>access_token,pc,r1
e6 28 ff eb     bl      copy_string
fc 1f 9f e5     ldr     r1,[DAT_000fabd0]
18 30 84 e2     add     r3,r4,#0x18
05 00 a0 e1     cpy     r0,r5
09 20 a0 e3     mov     r2,#0x9
01 10 8f e0     add     r1=>config_id,pc,r1
e0 28 ff eb     bl      copy_string
```

```
2c 30 88 e2     add     r3,r8,#0x2c
38 40 8d e2     add     r4,sp,#0x38
02 20 8f e0     add     r2,pc,r2
09 eb 8d e2     add     lr,sp,#0x2400
b0 20 88 e5     str     r2=>access_token,[r8,#0xb0]=>local_62c
00 20 a0 e3     mov     r2,#0x0
07 00 83 e8     stmia   r3,{ r0 r1 r2 }=>Access_Token
04 00 a0 e1     cpy     r0,r4
b4 10 88 e5     str     r1,[r8,#0xb4]=>local_628
5e 1f 8e e2     add     r1,lr,#0x178
38 20 88 e5     str     r2,[r8,#0x38]=>local_6a4
0a 20 a0 e1     cpy     r2,r10
ba 10 00 eb     bl      FUN_0007975c
```

ALPHV Access Tokens from ARM locker **(Source: Recorded Future)**

ALPHV MORPH - Windows

Strings deobfuscated with 1-byte XOR using “randomly generated” functions (with garbage code)

```
                    s_[|iz|afo(La{kg~mzmz_00e75894

00e75894 5b 7c 69 7a 7c 61 66 6f 28 4c 61 7b  ds        "[|iz|afo(La{kg~mzmz"
         6b 67 7e 6d 7a 6d 7a 00
```

```
int __fastcall deobfuscate_Starting_Discoverer_z(undefined4 param_1,byte param_2)

{
  int idx;
  byte curr;

  idx = 0;
  while( true ) {
    if (0x12 < idx) break;
    curr = s_Starting_Discoverer_00e75894[idx];
    DAT_00e75cd6 = param_2;
    s_Starting_Discoverer_00e75894[idx] = curr ^ 8;
    idx = idx + 1;
    DAT_00e75cd6 = DAT_00e75cd6 ^ 0xb2;
    DAT_00e75c73 = DAT_00e75c73 ^ 0x2c;
    param_2 = curr ^ 8;
  }
  DAT_00e75cd6 = DAT_00e75cd6 + 0x4e;
  return idx;
}
```

Deobfuscation function for “Starting Discoverer”

Windows binaries over 4 times the size of the “unobfuscated” versions - biggest increase in .text, .data and .reloc sections

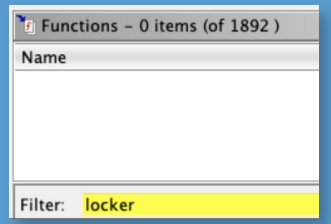
Name	Size
[redacted]_encrypt_app_creds_obfuscated1	14,518 KB
[redacted]_encrypt_app_creds_unobfuscated	3,006 KB

Name	Start	End	Length	R	W	X
Headers	00400000	004003ff	0x400	☑	☐	☐
.text	00401000	00e73fff	0xa73000	☑	☐	☑
.data	00e74000	00f42bff	0xcec00	☑	☑	☐
.rdata	00f43000	00feedff	0xabe00	☑	☐	☐
.eh_fram	00fef000	010b45ff	0xc5600	☑	☐	☐
.bss	010b5000	010b5643	0x644	☑	☑	☐
.idata	010b6000	010b83ff	0x2400	☑	☑	☐
.CRT	010b9000	010b91ff	0x200	☑	☑	☐
.tls	010ba000	010ba1ff	0x200	☑	☑	☐
.reloc	010bb000	012391ff	0x17e200	☑	☐	☐

Morph-Obfuscated Binary Section Information

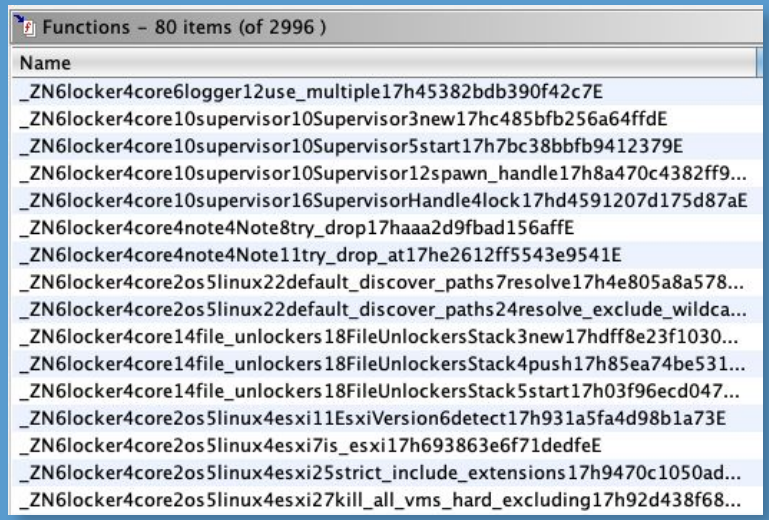
Rule 3: Check Your Work

No string obfuscation was present, but the Linux x64 Morph-obfuscated samples appear to now have the name-mangled function names, versus the unobfuscated ones with scrubbed names



Function names from “unobfuscated” x64 Linux/ESXi Samples

Exported variables from “unobfuscated” x64 Linux/ESXi Samples



Function names from “obfuscated” x64 Linux/ESXi Samples

Exported variables from “obfuscated” x86 Linux/ESXi Samples

Recorded Future

Rule 4: No, Really, Check Your Work

Fully testing encryption and decryption is critical - this is where the money is made

ESXiArgs version 1 (circa 2023)

- Encrypt 1MB and skip X MB where X is ~1% of file size
- Made recovery possible for very large files

Luna ransomware's ESXi locker (circa 2022)

- Encrypt VMs without shutting down
- May be corrupted after decryption

BlackMatter/DarkSide (circa 2020)

- Researchers helped decrypt victims without payment

Ryuk ransomware (circa 2019)

- Buggy decryptor did not work on large files

Luna ransomware, which appeared in July of 2022 and seeks out ESXi instances, does not shut down the virtual machines—a tactic that may lead to file corruption after decryption.

When VM files are not fully shut down during the encryption process, the files themselves become corrupt because they are unable to write data as expected within ESXi, said Betts, leading to "trash" files. Because the talks between guest and host did not finish properly, the virtual files may be left in a misconfigured, unusable state, even after deploying a decryption tool.

"Files are corrupted because they weren't able to shut down gracefully. So, things aren't written into the .vmx and the .vmdks and the .flat like they're supposed to," Betts told IT Brew.

Luna Ransomware VM Corruption **(Source: IT Brew)**

LockBit Ransomware

LockBit ransomware is one of the most active ransomware groups

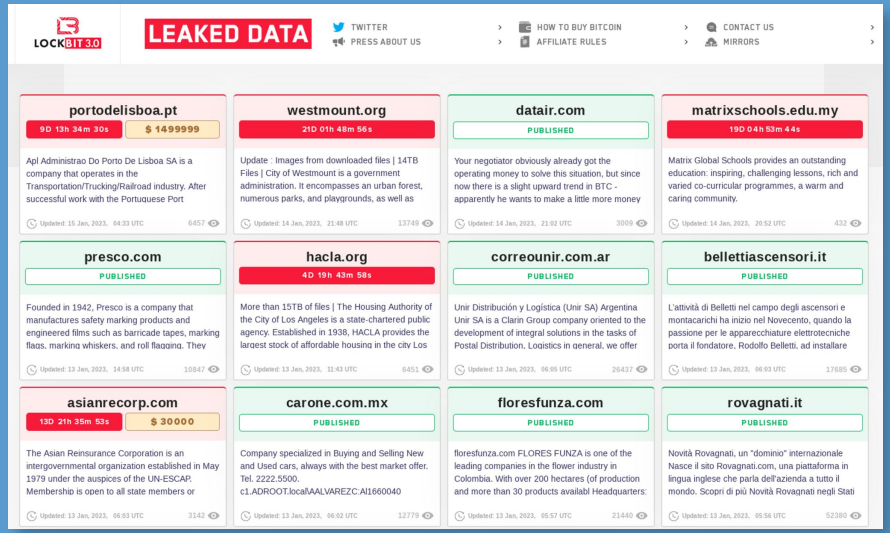
Written in Origin C

First observed in September 2019

- Continental Tire
- California Department of Finance
- FoxConn

Notable Features

- StealBit
- Recruiting Insiders
- Builder Leaked



Recorded Future

LockBit Ransomware

Recruiting “insiders” at companies for initial access

LockBit acquires BlackMatter code from fired developers - in comes LockBit “Black”

Rumor has it that the disgruntled developer leaked Lockbit Black code

- Customizable config allows anyone to modify
- Enables ransomware group spinoffs (eg: Bl00dy Ransomware)

LockBit “Green” based on Conti’s leaked source

- New ESXi variant
- Tor-based URLs belonging to LockBit found within samples

LockBit Builder Leaked **(Source: Recorded Future)**

LockBit Green based on Conti **(Source: ProDaft)**

Recorded Future

LockBit Black

“LockBit Black” looked very much like BlackMatter ransomware

```
setup_fns_z();
call_NtSetInformationThread_z(0);
generates_notename_and_token_membership_rel_z();
process_command_line_and_do_encrypt_z();
(*ExitProcess)(0);
GetCommandLineW();
GetProcAddress(unaff_retaddr,(LPCSTR)param_1);
GetLastError();
SetLastError((DWORD)param_2);
GetCommandLineW();
GetDlgItemTextW((HWND)param_3,(int)param_4,(LPWSTR)param_5,(int)param_6);
GetWindowTextW((HWND)param_7,(LPWSTR)param_8,(int)param_9);
GetDlgItemTextW((HWND)param_10,(int)param_11,(LPWSTR)param_12,(int)param_13);
LoadMenuW((HINSTANCE)param_14,(LPCWSTR)param_15);
LoadMenuW((HINSTANCE)param_16,(LPCWSTR)param_17);
DialogBoxParamW((HINSTANCE)param_18,(LPCWSTR)param_19,(HWND)param_20,(DLGPROC)param_21,
                (LPARAM)param_22);
GetDlgItem((HWND)param_23,(int)param_24);
LoadImageW((HINSTANCE)param_25,(LPCWSTR)param_26,param_27,param_28,param_29,(UINT)param_
GetWindowTextW((HWND)param_31,(LPWSTR)param_32,(int)param_33);
GetWindowTextW((HWND)param_34,(LPWSTR)param_35,(int)param_36);
DialogBoxParamW((HINSTANCE)param_37,(LPCWSTR)param_38,(HWND)param_39,(DLGPROC)param_40,
                (LPARAM)param_41);
LoadMenuW((HINSTANCE)param_42,(LPCWSTR)param_43);
DialogBoxParamW((HINSTANCE)param_44,(LPCWSTR)param_45,(HWND)param_46,(DLGPROC)param_47,
                (LPARAM)param_48);
CreateMenu();
GetDeviceCaps((HDC)param_49,(int)param_50);
```

BlackMatter “entry” function entry

```
do_decrypt_z();
setup_fns_z();
generates_notename_and_token_membership_rel_z();
process_command_line_and_do_encrypt_z();
(*ExitProcess)(0);
GetProcAddress(unaff_retaddr,(LPCSTR)param_1);
GetCommandLineA();
GetTickCount();
GetDateFormatW(param_2,(DWORD)param_3,(SYSTEMTIME *)param_4,param_5,(LPWSTR)param_6,param_7);
FormatMessageW((DWORD)param_8,(LPCVOID)param_9,param_10,(DWORD)param_11,(LPWSTR)param_12,
               (DWORD)param_13,(va_list *)param_14);
GetTickCount();
GetModuleHandleW((LPCWSTR)param_15);
LoadLibraryExA((LPCSTR)param_16,(HANDLE)param_17,param_18);
GetLocaleInfoW(param_19,(LCTYPE)param_20,(LPWSTR)param_21,(int)param_22);
GetCommandLineA();
GetLastError();
GetProcAddress((HMODULE)param_23,(LPCSTR)param_24);
GetLastError();
CreateWindowExW((DWORD)param_25,param_26,(LPCWSTR)param_27,param_28,param_29,param_30,param_31,
                (int)param_32,(HWND)param_33,(HMENU)param_34,(HINSTANCE)param_35,param_36);
GetDlgItem((HWND)param_37,(int)param_38);
GetMessageW((LPMSG)param_39,(HWND)param_40,param_41,(UINT)param_42);
EndDialog((HWND)param_43,(INT_PTR)param_44);
LoadMenuW((HINSTANCE)param_45,(LPCWSTR)param_46);
GetKeyNameTextW((LONG)param_47,(LPWSTR)param_48,param_49);
GetKeyNameTextW((LONG)param_50,(LPWSTR)param_51,(int)param_52);
DialogBoxParamW((HINSTANCE)param_53,(LPCWSTR)param_54,(HWND)param_55,(DLGPROC)param_56,param_57);
CreateWindowExW((DWORD)param_58,param_59,(LPCWSTR)param_60,param_61,param_62,param_63,param_64,
                (int)param_65,(HWND)param_66,(HMENU)param_67,(HINSTANCE)param_68,param_69);
GetClassNameW((HWND)param_70,(LPWSTR)param_71,(int)param_72);
```

LockBit Black “entry” function entry

Rule 5: Borrow, But Improve

Similarities

- High-level structure of the code
- API Hashing technique
- String hashing (eg: command line options)
- Configuration file decryption
- Anti-debugging techniques (eg: crash if breakpoint placed on its thread)

Differences

- Some LockBit Black versions require a password to decrypt
- Accepts additional command line parameters (eg: group policy modification, self-deletion)
- Configuration data flags

```
do {
  curr = *toHash;
  uVar1 = (uint)curr;
  if ((0x40 < curr) && (curr < 0x5b)) {
    uVar1 = uVar1 | 0x20;
  }
  curr = (byte)uVar1 & 0x1f;
  hashedVal._0_4_ = ((uint)hashedVal >> curr | (uint)hashedVal << 0x20 - curr) + uVar1;
  toHash = toHash + 1;
} while (uVar1 != 0);
```

BlackMatter/LockBit Black string hash (top), LockBit Black program arguments (bottom)

```
else {
            /* -pass */
  if (iVar6 == 0x459f1cd7) {
    (*wcscpy)((wchar_t *)&DAT_004271a8,ppWVar9[1]);
    (*RtlEncryptMemory)(&DAT_004271a8,0x48,0);
    (*memset)(ppWVar9[1],0,0x42);
    _DAT_004271a4 = 1;
    ppWVar9 = ppWVar9 + 2;
    numargs = numargs + -2;
    uVar7 = extraout_ECX_01;
    uVar8 = extraout_EDX_00;
  }
  else {
            /* -safe */
    if (iVar6 == 0x452f4997) {
      safe = true;
      ppWVar9 = ppWVar9 + 1;
      numargs = numargs + -1;
    }
    else {
            /* -wall */
      if (iVar6 == 0x45678b17) {
        wall = true;
        ppWVar9 = ppWVar9 + 1;
      }
      else if (iVar6 == 0x69268c17) {
        bVar2 = true;
        ppWVar9 = ppWVar9 + 1;
      }
      else {
            /* -psex */
        if (iVar6 == 0x69c71957) {
          bVar3 = true;
          ppWVar9 = ppWVar9 + 1;
        }
        else if (iVar6 == -0x349d16c0) {
          bVar4 = true;
          ppWVar9 = ppWVar9 + 1;
        }
        else {
            /* -gdel */
          if (iVar6 == 0x4b668957) {
```

And Keep Improving: LockBit Green

Highly similar to Conti - definitely based on leaked code

- API hashing functionality
- String decryption
- Overall structure

Many others using Conti code as well:

- Meow Ransomware
- ScareCrow
- BlueSky
- Putin team
- And More!

One key difference - ransom note is encrypted for LockBit green

- Can use decryption function features as part of signature!

```
                /* CryptImportKey */
pcVar1 = (code *)resolve_function_from_hash_z(extraout_ECX,0x10,0x70d2c0e4,0x37);
iVar2 = (*pcVar1)(hKey,&local_5c,0x2c,0,0,&local_2c);
if (iVar2 == 0) goto LAB_1001c431;
BVar3 = CryptSetKeyParam(local_2c,1,(BYTE *)&local_28,0);
if (BVar3 == 0) goto LAB_1001c44c;
_Size = local_60 * 2;
pdwDataLen = (DWORD *)do_malloc_z(_Size);
_memset(pdwDataLen,0,_Size);
DATA_LEN = pdwDataLen;
if (pdwDataLen != (DWORD *)0x0) {
  _memcpy_call(pdwDataLen,&RANSOM_NOTE,local_60);
  BVar3 = CryptDecrypt(hKey,local_2c,0,1,(BYTE *)0x0,pdwDataLen);
```

Ransom note decryption function for LockBit Green

What's Next?

LockBit Black has been fairly consistent - not as much change over time as other variants, however:

- Some samples have no "decrypt" function and do not require a password to run
- Option the builder provides

LockBit Green could evolve too, but too early to know

Bonus: StealBit used to automate data exfiltration tool of victim files and upload them to the LockBit leak site

```
var_x = 200000000;
do {
  var_x = var_x + -1;
} while (var_x != 0);
curr = (short *)get_command_line_z();
var_x = parse_command_line_arguments_looking_for_password_z
                  (extraout_ECX,extraout_EDX,curr,(byte *)pass_out);
if (var_x != 0) {
  string_rel_z(local_64,pass_out);
  local_68 = do_RC4_KSA_z((int)local_64,(int)local_44,(int)local_178);
  var_x = get_img_base_rel_z();
  var_x = *(int *)(var_x + 8);
  iVar3 = *(int *)(var_x + 0x3c) + var_x;
  uVar4 = (uint)*(ushort *)(iVar3 + 6);
  pbVar4 = (CSTRING *)(iVar3 + 0xf8);
  uVar1 = extraout_ECX_00;
  uVar2 = extraout_EDX_00;
  do {
    uVar5 = hash_string_z(uVar1,uVar2,&pbVar4->field0_0x0,0);
    uVar2 = (undefined4)((ulonglong)uVar5 >> 0x20);
    iVar3 = (int)uVar5;
    if (((iVar3 == 0x76918075) || (iVar3 == 0x4a41b)) ||
       (uVar1 = extraout_ECX_01, iVar3 == 0xb84b49b)) {
      do_decrypt_z((byte *)(pbVar4->field12_0xc + var_x),pbVar4->contents,local_178,local_68);
      uVar1 = extraout_ECX_02;
      uVar2 = extraout_EDX_01;
    }
    pbVar4 = pbVar4 + 1;
    uVar4 = uVar4 - 1;
  } while (uVar4 != 0);
```

LockBit decryption function

Lost in Translation

LockBit told Recorded Future that they live in China and that none of their affiliates live in the United States or Russia

Threat actors post machine-translated Chinese asking about ransomware

RAMP forum welcomes Chinese speakers

- Aim to attract Chinese threat actors and ransomware gangs

DS: After the US and Russian presidents met in June everyone is looking for signs of change. And I see some change – the attacks have increased after a temporary slowdown in summer. Are these events related or did the affiliates just go for a long vacation?

LB: It's just a summer vacation. Like all people on the planet, no one wants to work in the summer, and even more so when you have millions of dollars. The meetings of the presidents will not affect anything, everyone who works seriously does not live in the United States or Russia. Personally, I live in China and feel completely safe.

LockBit Interview **(Source: Recorded Future)**

User posts machine translated post on XSS forum **(Source: Flashpoint)**

Recorded Future

….And Found in Translation

When chats belonging to Yanluowang ransomware were leaked, it was discovered that they, too are Russian-speakers - not Chinese

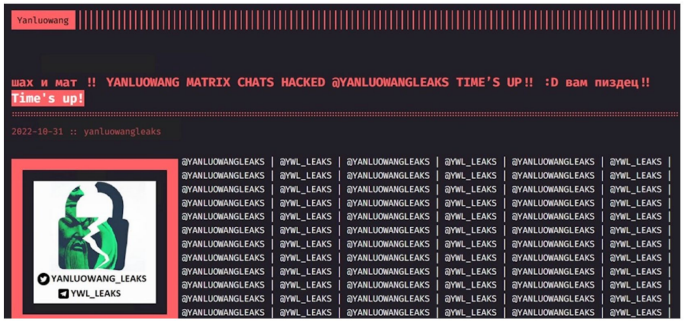


IMAGE: SCREENSHOT OF THE YANLUOWANG LEAK SITE

Dina Temple-Raston
November 22, 2022
News

The Yanluowang ransomware group in their own words

It announced that the contents of one of the group's discussion channels – some 2,700 messages sent between January and September 2022 – had been breached and was now uploaded to a leak site that allowed researchers, law enforcement, and even competitors to understand how the group was organized, how it interacted with other ransomware actors, and who might be in charge.

"We wanted to dig into the internal chats and figure out what we could locate there — what their TTPs [tactics, techniques, and procedures] tradecraft is, was there any collaboration with other ransomware families, "said Jambul Tologonov, a researcher at the cybersecurity firm Trellix. "That's what my mindset was when I started the investigation, and the first thing I noticed was that their conversations were all in Russian."

Rule 6: Oh, and Don't Cut Corners

```
Hi, since you are reading this it means you have been hacked.
In addition to encrypting all your systems, deleting backups, we also downloaded some of confidential information.
Here's what you shouldn't do:
1) Contact the police, fbi or other authorities before the end of our deal
2) Contact the recovery company so that they would conduct dialogues with us. (This can slow down the recovery, and generally put our communication
to naught)
3) Do not try to decrypt the files yourself, as well as do not change the file extension yourself !!! This can lead to the impossibility of their
decryption.
4) Keep us for fools)
We will also stop any communication with you, and continue DDoS, calls to employees and business partners.
In a few weeks, we will simply repeat our attack and delete all your data from your networks, WHICH WILL LEAD TO THEIR UNAVAILABILITY!
Here's what you should do right after reading it:
1) If you are an ordinary employee, send our message to the CEO of the company, as well as to the IT department
2) If you are a CEO, or a specialist in the IT department, or another person who has weight in the company, you should contact us within 24 hours by
email.
We are ready to confirm all our intentions regarding DDOS, calls, and deletion of the date at your first request.
As a guarantee that we can decrypt the files, we suggest that you send several files for free decryption.
Mails to contact us:
1)son.goku@mailfence.com
2)leen.cang@mailfence.com
Our leak site :
crptd5sv5bdz6hovrbkac6mnp3rt7zij62njsqwh5a6ldd3asxdd22qd.onion
```

Shao Ransomware note

Not Shao

```
root@ubuntu:/home/user/Desktop# ./revz
Revix 1.1c
Usage example: elf.exe --path /vmfs/ --threads 5
Without --path - it encrypts current dir
--silent (-s) use for not stoping VMs mode
!!!BY DEFAULT THIS SOFTWARE USES 50 THREADS!!!
Path: .
killing vmx-*
esxcli --formatter=csv --format-param=fields=="WorldID,DisplayName" vm process list | awk -F "\"*,\"*" '{system("esxcli vm process kill --type=force
--world-id=" $1)}'
```

Shao Ransomware command line output

Recorded Future

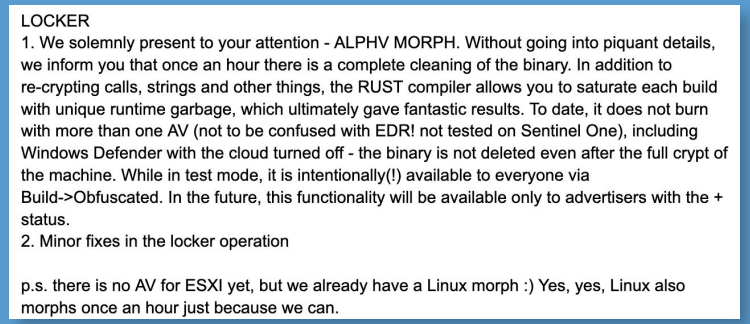
Hunt ‘Em

Face the Strange!

- Automated obfuscation techniques often leave artifacts; better if custom
- Inconsistencies in language/strings, ransom notes especially!
- Anti-RE/anti-debugging/anti-analysis techniques
- Implementation of crypto algorithms
- “Buggy” anomalies
- Stay up on the latest affiliate news

Look for the similarities

- Code reuse between families
- Overlap in ransom note language (eg: “What Happened?”, “your network”, “torproject.org”)


LOCKER
1. We solemnly present to your attention - ALPHV MORPH. Without going into piquant details, we inform you that once an hour there is a complete cleaning of the binary. In addition to re-crypting calls, strings and other things, the RUST compiler allows you to saturate each build with unique runtime garbage, which ultimately gave fantastic results. To date, it does not burn with more than one AV (not to be confused with EDR! not tested on Sentinel One), including Windows Defender with the cloud turned off - the binary is not deleted even after the full crypt of the machine. While in test mode, it is intentionally(!) available to everyone via Build->Obfuscated. In the future, this functionality will be available only to advertisers with the + status.
2. Minor fixes in the locker operation

p.s. there is no AV for ESXI yet, but we already have a Linux morph :) Yes, yes, Linux also morphs once an hour just because we can.

ALPHV announce MORPH **(Source: Recorded Future)**

Defend the Net

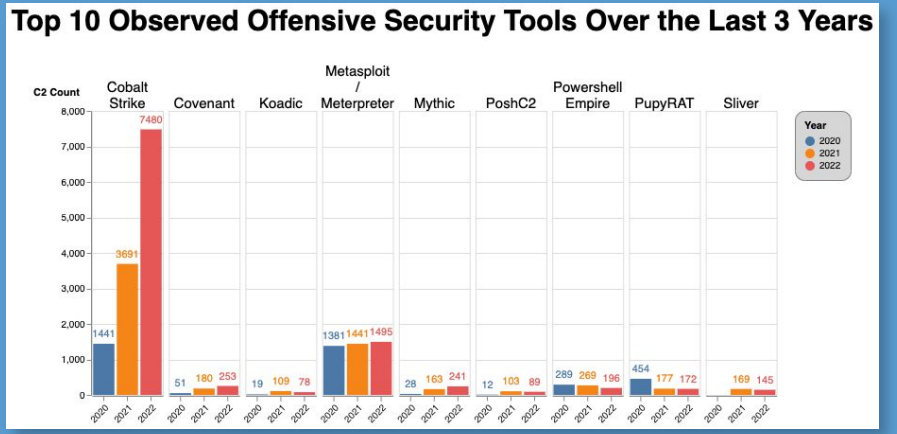
The ransomware evolves, but tried and true techniques are still used - they just keep working

Implement best practices

- Strong passwords and MFA
- Patching systems wherever possible, prioritizing externally facing
- Disaster Recovery and Backup Plan
- Pruning accounts
- Active Directory cleanup

Focus on the pre-ransomware tools first

- Stealers such as RedLine, Raccoon, Vidar
- Openly available tools like Cobalt Strike, OST, bots and trojans
- Active Directory enumeration, password spraying, lateral movement techniques



Excerpt from Adversary Infrastructure Trends 2022 Report showing top OST over last 3 years
(Source: Recorded Future)

Recorded Future

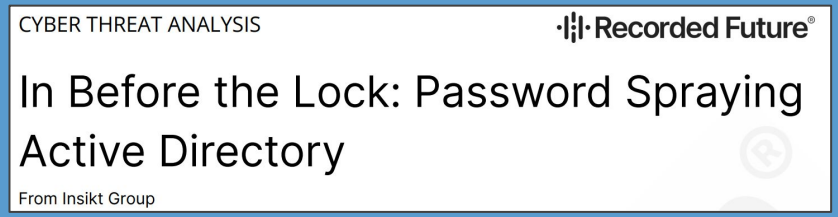
Defend the Net: Active Directory

Active Directory is still an effective target for threat actors looking to escalate an attack

- Enumeration: identify possible paths from compromised systems to obtaining a higher privilege level, such as Domain Administrator access
- Password Spraying: post-enumeration, can be used to gain access to systems of interest

Largely possible using openly available, “red team” tools - often used with Cobalt Strike

- Lowers barrier to entry
- Lessens risk of attribution
- Challenge to detect increases with Cobalt Strike



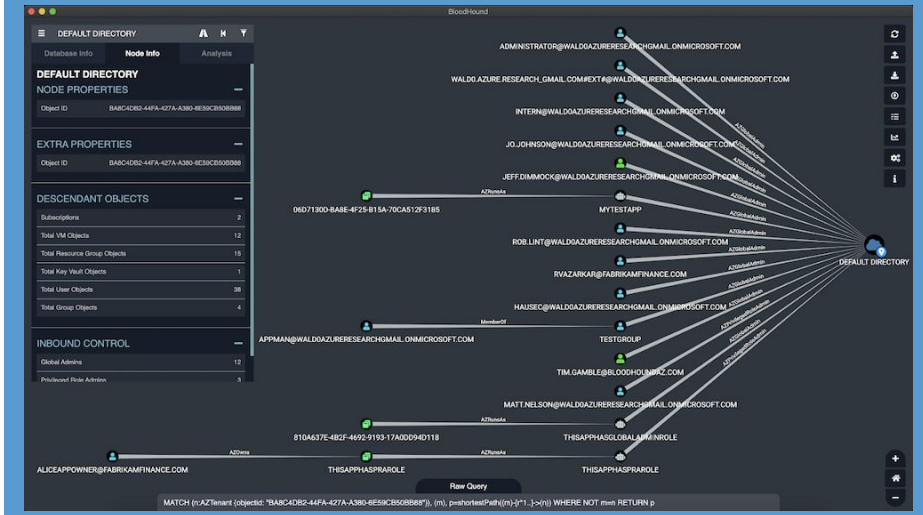
Active Directory Enumeration

Evaluated 3 common tools

- SharpHound/BloodHound: collect and visualize AD information, including active sessions on machines, Group Policy details, access control entries
- ADFind: command line tool that is used to query Active Directory
- LACheck: C# tool used to enumerate administrative rights, sessions, logged-on users, etc

Detection Opportunities

- Any tool run with Cobalt Strike: look for Beacon activity instead of tool-specific
- SharpHound: Sigma rules for process/file creation events, PowerShell (“Invoke-BloodHound”, compressed tool bytes)
- ADFind, LACheck: Sigma rules for command line options/parameters in combination with general AD enumeration mitigations
- General: many DNS requests (Sysmon EventID: 22) and network requests (Sysmon EventID: 3) for LDAP over port 389AD HoneyTokens



Bloodhound visualization (Source: SpecterOps.io)

```
index="main" ((TargetFilename="*_domains.json*" OR TargetFilename="*_users.json*" OR
TargetFilename="*_groups.json*" OR TargetFilename="*_ous.json*" OR
TargetFilename="*_computers.json*" OR TargetFilename="*_BloodHound.zip*" OR
TargetFilename="*_BloodHoundLoopResults.zip*" OR TargetFilename="*_gpos.json*")
((Image="*SharpHound.exe*" OR Image="*rundll32.exe*") OR (CommandLine="*SharpHound.exe*") OR
(Description="*SharpHound*") OR (OriginalFileName="*SharpHound.exe*"))) OR
((OriginalFileName="*SharpHound.exe*") ((CommandLine="*nosavecache*" OR
CommandLine="*excludeddomaincontrollers*" OR CommandLine="*stealth*" OR CommandLine="*jitter*"
OR CommandLine="*throttle*") OR (Image="*SharpHound.exe*") OR (CommandLine="*SharpHound.exe*")
OR (Description="*SharpHound*")))| transaction startswith=(EventCode="1") endswith=
(TargetFilename="*zip") maxspan=600s
```

Detects the Sharphound process creation event in combination with the file creation events within a time span of 600s

Recorded Future

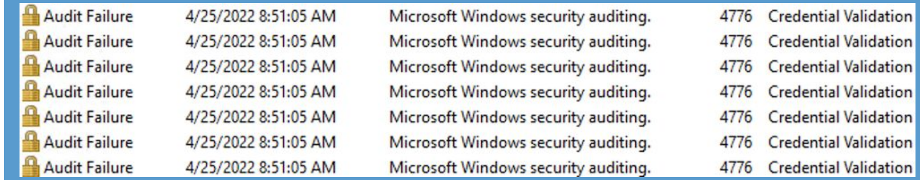
Password Spraying

Evaluated 3 C# password spraying tools

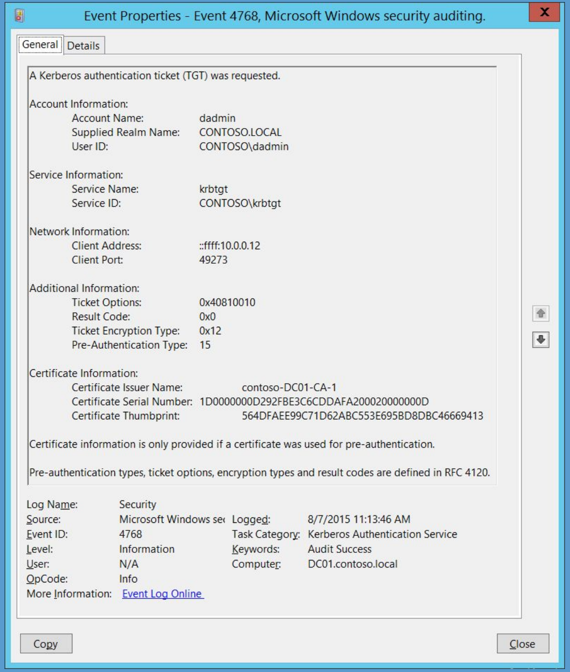
- SharpHose/SharpSpray: C# implementation of DomainPasswordSpray, designed to perform password spraying against Active Directory objects
- SharpMapExec: Scan for access to SMB shares, PsRemote, and vulnerable JEA endpoints, perform domain password spraying, execute local C# assemblies in memory (such as Rubeus or Cobalt Strike Beacon)

Detection Opportunities

- Any tool run with Cobalt Strike: look for Beacon activity instead of tool-specific (Again)
- Largely, Sigma rules for command line parameters, default configuration (eg: defined password list)
- Windows Event IDs for password spraying include:
  - 4625: An account failed to log on
  - 4648: A logon was attempted using explicit credentials
  - 4768: A Kerberos authentication ticket (TGT) was requested
  - 4771: Kerberos pre-authentication failed
  - 4776: The computer attempted to validate the credentials for an account



Security Logs for Failed Password Attempts



Event ID 4768 triggered by password spraying

Recorded Future®

Thank You!

Recorded Future®

Lindsay Kaye @TheQueenofELF