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
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

ReadMe - Notepad

File Edit Format View Help

PLAY
boitelswaniruxl@gmx.com
ROP in Ransomware??

Increment ESP by $0\times32$ and RET causes a “jump” to real code

“Garbage” code bytes

Earlier PLAY code without ROP
(Source: Recorded Future)

Newer PLAY code using ROP
(Source: 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
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
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
**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

---

**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’s 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.
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
Strings deobfuscated with 1-byte XOR using "randomly generated" functions (with garbage code)

```
s_\[\|iz\]afo\(\{a|\}kg-nmz_00e75894
00e75894 5b 7c 69 7a 7c 61 66 6f 28 4c 61 7b 6d 67 7e 6d 7a 6d 7a 00
"\|iz\]afo\(\{a|\}kg-nmz"
```

Deobfuscation function for "Starting Discoverer"

```
int _fastcall deobfuscate_starting_discoverer_2(undefi\ened4 param_1, byte param_2)
{
    int idx;
    byte curr;
    idx = 0;
    while (true) { break;
        if (0x12 < idx) break;
        curr = s_starting_discoverer_00e75894[idx];
        DAT_0\[8e758d6 = param_2;
        s_starting_discoverer_00e75894[idx] = curr ^ 8;
        idx = idx + 1;
        DAT_0\[8e75cd6 = DAT_0\[8e75cd6 ^ 0xb2;
        DAT_0\[8e75cd7 = DAT_0\[8e75cd7 ^ 0x2c;
        param_2 = curr ^ 8;
    }
    DAT_0\[8e75cd6 = DAT_0\[8e75cd6 + 0x4e;
    return idx;
}
```

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

- Exported variables from "unobfuscated" x64 Linux/ESXi Samples
- Function names from "unobfuscated" x64 Linux/ESXi Samples
- Exported variables from "obfuscated" x64 Linux/ESXi Samples
- Function names from "obfuscated" x64 Linux/ESXi Samples
- Exported variables from "obfuscated" x86 Linux/ESXi Samples
- Function names from "obfuscated" x86 Linux/ESXi Samples
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
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: Bi00dy Ransomware)

LockBit “Green” based on Conti’s leaked source

• New ESXi variant
• Tor-based URLs belonging to LockBit found within samples
“LockBit Black” looked very much like BlackMatter ransomware.
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
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!

```c
/* 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,6,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;
d0 { 
    var x = var x + 1;
} while (var x != 0);
curr = (short *)get_command_line_z();
var x = parse_command_line_arguments_locking_for_password_z
                (extrout_ECK, extrout_EKM, curr, (byte *)pass_out);
if (var x != 0) {
    string rel2[(local_64,(int)local_64,(int)local_64,(int)local_170);
    var x = get_long_base_rel_2();
    var x = (int *)&(var x + 0);
    var x = (int *)&(var x + 0x4e) + var x;
    uVar4 = (uint)*((ushort *)&(var x + 0);
    pVar4 = ((cstring *))(var x + 0xf8);
    uVar4 = extrout_ECK_00;
    uVar4 = extrout_ECK_00;
    do {
        uVar3 = hash_string_z(uVar1, uVar2, uVar3, 0x08, 0);
        uVar2 = (undefined4)((long long)uVar3 >> 0x20);
        uVar2 = (int)uVar2;
        if (((Var3 < 0x70910075)) || (var x = 0x4a4f1)) ||
            (uVar3 = extrout_ECK_01, uVar3 = 0x5b84b69a) (do_decrypt_z((byte *)pVar4 = field12_0xc + var x), pVar4 = contents, local_170, local_60);
        uVar2 = extrout_ECK_00;
        uVar2 = extrout_ECK_00;
    };
    pVar4 = pVar4 + 1;
    uVar4 = uVar4 + 1;
} while (uVar4 != 0);
```

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

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 communications 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 (im)possibility 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@hallfence.com
2)leen.cang@hallfence.com

Our leak site: crptdsvsvbdzehovrbkacemnp3rt7zi62nsqwh5a6ld3asxd22qo.onton

---

root@ubuntu:/home/user/Desktop# ./revz
ReviX 1.1c
Usage example: elf.exe --path /vmfs/ --threads 5
Without --path - it encrypts current dtr
--silent (-s) use for not stopping VMs mode
!!!BY DEFAULT THIS SOFTWARE USES 50 THREADS!!!
Path:
kl1ing vmx-*
esxcli --formatter=csv --format-param=fields="WorldID,DisplayName" vm process list | awk -F "|",|" '{system("esxcli vm process kll --type=force --world id="$1";)}'
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-encrypting 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
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 389 AD HoneyTokens

Bloodhound visualization (Source: SpecterOps.io)

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

Lindsay Kaye @TheQueenofELF