Under Lock & Key:
A Deep Dive into Ransomware Encryption

Jasper Bongertz & Melissa Eckardt

G DATA Advanced Analytics

G DATA advanced analytics

Working in the **C**omputer **S**ecurity **I**ncident **R**esponse **T**eam (CSIRT) at G Data Advanced Analytics, Bochum

- Computer/Memory/Network Forensics
- Malware Analysis & In-Depth Reversing
- Incident Handling

Helping customers in dozens of cases per year

- Mostly Ransomware
- From 100+ to >40.000 users

RANSOMWARE ENCRYPTION

Ransomware Execution

Ransomware encryption software is usually run “manually”

- Threat Actors trigger the execution as a last step
- Sometimes “by hand” on each system

G DATA advanced analytics

Ransomware Operations

- Encryption tools usually rely on mature OS-internal cryptographic routines
  - In those cases, decryption requires paying the ransom
- After successful execution, the encryption binary sometimes deletes itself
  - Probably an attempt to make forensic analysis harder
  - In most of our cases we still found it

Ransomware Perception

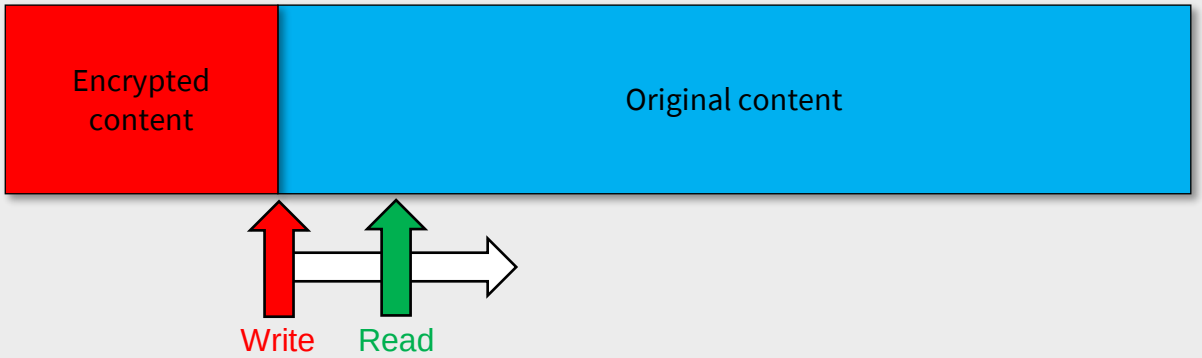
- Victims of ransomware often confuse “encryption” with “infection”
- In the last couple of years, encryption software wasn’t seen “worming” itself through the network anymore
- Fear of automatic “reinfection/encryption” is unfounded

Ransomware File Targeting

- Selection of files usually targets non-OS elements, e.g.
  - Documents
  - Databases
  - Pictures & Movies
  - …
- Most encryption tools have “exclude lists” to prevent harm to vital OS files and folders
- Usually, mounted shares are encrypted, too

File Operation

- Files are read & overwritten with encrypted content “in place”
  - This means “undelete” operations can’t work
  - It also avoids most disk space problems



File Operation

- To prevent decryption by leaked/forensically acquired keys, each file is encrypted with a unique key
  - This means that even memory forensics usually doesn’t help decrypting files
- The encryption key is built from two parts:
  - A public/private key pair, valid for all files of the victim
  - A symmetric key, locally generated, just for the current file
  - The symmetric key is wrapped using asymmetric encryption

File Operation

- The file specific part is kept in either
  - an additional, separate file (usually a text file)

  - Appended in binary form to the encrypted file

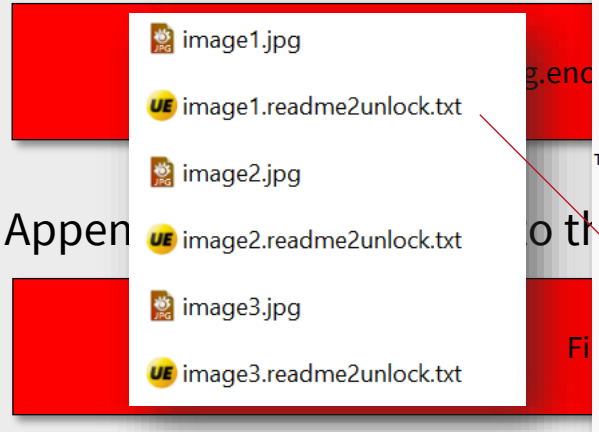
File Operation

- The file specific part is kept in either
  - an additional, separate file (usually a text file)

  - Appen o the encrypted file

File Operation

- The file specific part is kept in
  - an additional, separate file (u



  - Appen

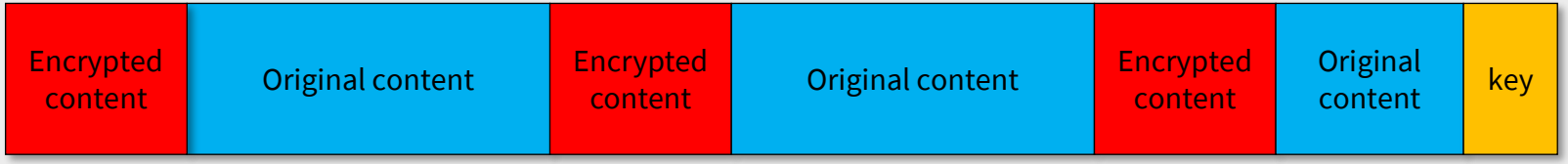
```
Your network has been penetrated.

All files on each host in the network have been encrypted with a strong algorythm.

Backups were either encrypted or deleted or backup disks were formatted.
Shadow copies also removed, so F8 or any other methods may damage encrypted data but not recover.

We exclusively have decryption software for your situation
No decryption software is available in the public.

                    DO NOT RESET OR SHUTDOWN - files may be damaged
                    DO NOT RENAME OR MOVE the encrypted and readme files
                    DO NOT DELETE readme files
                    DO NOT use any recovery software with restoring files overwriting encrypted
                    This may lead to the impossibility of recovery of the certain files.


To get info (decrypt your files) contact us at your personal page:

                    1. Download and install Tor Browser: https://www.torproject.org/download/
                    2. After a successful installation, run the browser and wait for initialization
                    3. Type in the address bar:

                            [URL REDACTED]

                    4. Follow the instructions on the site
                    5. You should get in contact in 48 HOURS since your systems have been infected.
                    6. The link above is valid for 7 days.
                            After that period if you not get in contact
                            your local data would be lost completely.


The faster you get in contact - the lower price you can expect.

DATA
ABoBAQADAQEBBBQ2gAMAwEAAhADEAAAAfKEJDuLhOLhnzI6OBBlzI6Fzc6lxI6Pzc6PzR2XIjtJhuTxggWX
Kv5AAkEQuGQEEQEEQOGgcMuZhkzidiETEImcRMYzohOnGJGeo6ih0B36AY6SEkhAYGZj9oJ516Z5r6yXzdg
OPLpyOFdN4mXym5ojM2PXuVEabWEDTZ65PQdF55ozVdKCUXFLKpjHdG6BO0Q6vQsXzLqavS0N8ExgH5j6f5
SZJ0YcjjIOsuNMJU6PYB937AF1I5P2RHr7aAZbKarLkbqHYm2EKyPKnZx0zhOzhOJBEBBEBBEBBOJBOLjuy
DIHDYUODcxwTDOnETEO6cchId04kkLoHQd04nRCJiE6IRMQnRCdIUjjMPVrurtTs
```

Encryption Speed

- Ransomware encryption speed is vital
  - Victims may interrupt encryption if running too long
  - Threat actors usually trigger the encryption in the nighttime to delay detection
  - In some cases, weekends or public holidays were chosen
- Encrypting TByte of data still takes a long time
  - To improve speeds, stripe encryption is used
  - We have seen speeds of >100TB/h

Encryption Speed

- Stripe encryption happens for large files (usually > ~10MB)



- Depending on number of stripes, stripe size and file size, large portions of a file may be recoverable
- Carving content from virtual disks has been proven successful in some cases

Decryption process

- Decryption also happens in-place
- Decryption tools are usually less well coded than the encryption tools
  - This may cause files to be broken without a chance to repair
  - Backups of encrypted files are necessary to avoid problems
- Needs access to the specific key material for each file

RANSOMWARE ENCRYPTION FAILURES

Crypto Fails: SAN Performance

- During encryption of VMs in a virtual environment in a hospital, the SAN storage failed
  - Further encryption was prevented by the I/O overload situation
- IT staff assumed an IT outage and began restoring from backup
  - Found the ransom note after restoring ~10 VMs

Crypto Fails: Wiped Storage

- In at least one case, encryption failed for unknown reasons
- Storage was left completely wiped clean
- Threat actor claimed encryption worked and demanded ransom payment

Crypto Fails: Weak key generation

- Sometimes a weak random number generator is chosen
- Seed depends on…
  - Current time
  - System uptime (high resolution)
  - Low entropy details such as process id
- In the end: Just around 56 bits of entropy (in a 128-bit seed)
  - Problem: “Confirming” a seed requires computing ~250 SHA-256 rounds

```
1  void rand_init()
2  {
3    struct _SYSTEMTIME SystemTime; // [esp+8h] [ebp-20h] BYREF
4    struct _FILETIME FileTime; // [esp+18h] [ebp-10h] BYREF
5    LARGE_INTEGER PerformanceCount; // [esp+20h] [ebp-8h] BYREF
6
7    InitializeCriticalSectionAndSpinCount(&csRand, 0xFA0u);
8    EnterCriticalSection(&csRand);
9    QueryPerformanceCounter(&PerformanceCount);
10   aRandom[0] ^= GetTickCount();
11   aRandom[1] ^= PerformanceCount.HighPart;
12   aRandom[2] ^= PerformanceCount.LowPart;
13   aRandom[3] ^= GetCurrentProcessId();
14   aRandom[4] ^= GetCurrentThreadId();
15   GetLocalTime(&SystemTime);
16   SystemTimeToFileTime(&SystemTime, &FileTime);
17   aRandom[5] ^= FileTime.dwHighDateTime;
18   aRandom[6] ^= FileTime.dwLowDateTime;
19   QueryPerformanceCounter(&PerformanceCount);
20   aRandom[7] ^= PerformanceCount.HighPart;
21   aRandom[0] ^= PerformanceCount.LowPart;
22   do
23     rand_next();
24   while ( LOBYTE(aRandom[0]) );
25   bRandIntialized = 1;
26   LeaveCriticalSection(&csRand);
27 }
```

Crypto Fails: Known Plaintext Attacks

- When threat actors invent their own custom “encryption” they often make mistakes
- This means that decryption of files may be possible without paying the ransom
- Specifically, in one case cyclic XOR was used instead of a cipher

```
40  for ( i = 0LL; dataSize_ > i; ++i )
41  {
42    if ( dataSize <= i )
43      runtime_panicIndex(i);
44    fileByte = inData[i];
45    if ( !keySize )
46      runtime_panicdivide(outData, 0LL, dataSize_, dataSize, inData, i, fileByte);
47    if ( i % keySize >= (unsigned __int64)keySize )
48      runtime_panicIndex(i % keySize);
49    outData[i] = key[i % keySize] ^ fileByte;
50  }
51  return outData;
52}
```

Decompiled Go code; key size was 0x500 bytes

Crypto Fails: Programming Errors

- Threat actor chose XChaCha20 as symmetric cipher ✔️
- Crypto++ was used as cryptographic library ✔️
- A mistake was made with handling the cipher object
  - Internal state was reset for each stripe iteration
  - All stripes in the file were encrypted with the same nonce 💥
  - Known plaintext attack possible and feasible

Details:
https://www.srlabs.de/blog-post/black-basta-buster-decrypting-files-without-paying-the-ransom

THANK YOU! QUESTIONS?