DEFENCE TECH
Terra, Cielo, Mare, Spazio, Spazio cibernetico.
PROTEGGIAMOLI

Mimic Ransomware

Malware Lab Analysis Report

Summary

DEFENCE TECH

1

Our Malware Lab

1. Our Malware Lab

Defence Tech Malware Lab daily performs dissection of malware with the aim of timely understanding the technological evolutions of attacks, consolidating the knowledge of necessary to make more effective and faster the process of incidents responding, contributing to spreading information about emerging threats into the expert's community and among its clients.

Malware Lab analysts are continuously engaged in searching and experimenting new analysis tools, for increasing accuracy and scope of action with regard to the proliferation of new evasion and anti-analysis techniques adopted by malwares.

The Malware Lab is also committed to the development of proprietary tools for malware analysis and supporting the management and response of incidents.

Besides malware analysis, Malware Lab ideated and implemented an automatic process of extraction of Indicators of Compromise (IOC) that is daily run on dozens of new malwares, intercepted in the wide for populating our Knowledge Base.

CORRADO AARON VISAGGIO
Group Chief Scientist Officer & Malware Lab Director
a.visaggio@defencetech.it

2

Executive Summary

2. Executive Summary

We analysed a malware family after capturing a sample from an infected machine after a successful attack, in this case we could only handle the post-incident stage. According to the IoCs left on the machine, the sample was version 4.3 of Mimic Ransomware.
This is a rather uncommon ransomware family but nonetheless very effective, in this report we analyse its features and file encryption algorithm which has not been publicly documented until now.

This family implements many malicious techniques and offers support for running custom commands defined at build time making it very dangerous and hard to predict with exact behaviour configurable by the attacker. Among its many features we observed support for disabling Windows Defender, deleting backup catalogs, disable automatic recovery features using wbadmin.exe and clearing Windows event logs, these are all common techniques in ransomware.

Mimic Ransomware drops a ransom note in the temp folder on the C:\ drive and creates an autorun registry key with the purpose of showing the ransom upon every boot. It is also able to bypass the Windows User Account Control (UAC) settings to stealthily execute code with elevated permissions, abusing the ICMLuaUtil COM interface* **.

As already documented by Trend Micro***, it possesses the ability of abusing the application Everything's**** APIs, a file name search engine, to speed up the encryption. The sample typically carries Everything standalone executable files, this way it's also able to attack machines where the software is not installed. In this case the encrypted file extension was different than any documented online producing no search results, we assume it was randomly generated when the sample was built.

* https://attack.mitre.org/tactics/TA0004/

** https://attack.mitre.org/techniques/T1548/

*** https://www.trendmicro.com/en_us/research/23/a/new-mimic-ransomware-abuses-everything-apis-for-its-encryption-p.html

**** https://www.voidtools.com/

Moreover, according to Recorded Future's* Threat Intelligence platform, Mimic Ransomware reuses code from the leaked Conti Ransomware builder, including functions for enumerating encryption modes, enumeration of Windows shares and port scanning.

We found through our independent research that the file encryption code is completely custom and based on OpenSSL, unlike Conti which used Windows's cryptography API.

* *https://www.recordedfuture.com/platform/threat-intelligence*

3

Analysis

3. Analysis

3.1 Everything for file enumeration

The payload includes two executables: everything.exe and everything32.dll, which are parts of a legitimate file indexing and search software called Everything; Mimic uses Everything indexing capabilities to efficiently enumerate the files on the local computer and, if enabled in the configuration, will enumerate also the network shares in order to encrypt them.

The configuration of the sample includes a list of files and folders that are always excluded from encryption, this essentially boils down to installed programs, system files and the malware itself.

For post-incident scenarios the configuration can be easily retrieved from the log file produced by the malware without the need of analysing the sample, the following table shows the list of excluded files and directories in our case.

Excluded files list	*boot.ini, bootfont.bin, desktop.ini, iconcache.db, io.sys, ntdetect.com, ntldr, ntuser.dat, ntuser.ini, thumbs.db, *.exe*
Excluded folders list	*steamapps, Cache, Boot, Chrome, Firefox, Mozilla, Mozilla Firefox, MicrosoftEdge, Internet Explorer, Tor Browser, Opera, Opera Software, Common Files, Config.Msi, Intel, Microsoft, Microsoft Shared, Microsoft.NET, MSBuild, MSOCache, Packages, PerfLogs, Program Files, Program Files (x86), ProgramData, System Volume Information, tmp, Temp, USOShared, Windows, Windows Defender, Windows Journal, Windows NT, Windows Photo Viewer, Windows Security, Windows.old, WindowsApps, WindowsPowerShell, WINNT, $RECYCLE.BIN, $WINDOWS.~BT, $Windows.~WS, C:\Users\Public\, C:\Users\Default\*

3.2 Malware features

Before starting the encryption process, Mimic can perform several actions that can be scripted in its configuration, these include running arbitrary commands, deleting Windows shadow copies backups, and disabling Windows Defender.

The malware includes a User Account Control (UAC) bypass technique to escalate privileges; UAC bypasses are well documented but not considered a security boundary according to Microsoft*, however EDR software will usually detect this kind of behaviour.

Then it will try to terminate a hardcoded list of services and processes, in the case of our infected machine the configuration captured from log files is reported in the following table.

Processes to terminate	*agntsvc.exe, AutodeskDesktopApp.exe, axlbridge.exe, bedbh.exe, benetns.exe, bengien.exe, beserver.exe, CoreSync.exe, Creative Cloud.exe, dbeng50.exe, dbsnmp.exe, encsvc.exe, EnterpriseClient.exe, fbguard.exe, fbserver.exe, fdhost.exe, fdlauncher.exe, httpd.exe, isqlplussvc.exe, msaccess.exe, MsDtSrvr.exe, msftesql.exe, mspub.exe, mydesktopqos.exe, mydesktopservice.exe, mysqld.exe, mysqld-nt.exe, mysqld-opt.exe, ocautoupds.exe, ocomm.exe, ocssd.exe, oracle.exe, pvlsvr.exe, node.exe, java.exe, python.exe, wpython.exe, QBDBMgr.exe, QBDBMgrN.exe, QBIDPService.exe, qbupdate.exe, QBW32.exe, QBW64.exe, Raccine.exe, Raccine_x86.exe, RaccineElevatedCfg.exe, RaccineSettings.exe, VeeamDeploymentSvc.exe, RAgui.exe, raw_agent_svc.exe, SimplyConnectionManager.exe, sqbcoreservice.exe, sql.exe, sqlagent.exe, sqlbrowser.exe, sqlmangr.exe, sqlservr.exe, sqlwriter.exe, Ssms.exe, Sysmon.exe, Sysmon64.exe, tbirdconfig.exe, TeamViewer.exe, TeamViewer_Service.exe, tv_w32.exe, tv_x64.exe, tomcat6.exe, vsnapvss.exe, vxmon.exe, wdswfsafe.exe, wsa_service.exe, wxServer.exe, wxServerView.exe, xfssvccon.exe*
Services to terminate	*AcronisAgent, ARSM, backup, BackupExecAgentAccelerator, BackupExecAgentBrowser, BackupExecDiveciMediaService, BackupExecJobEngine, BackupExecManagementService, BackupExecRPCService, BackupExecVSSProvider, CAARCUpdateSvc, CASAD2DWebSvc, ccEvtMgr, ccSetMgr, Culserver, dbeng8, dbsrv12, DefWatch, FishbowlMySQL, GxBlr, GxCIMgr, GxCVD, GxFWD, GxVss, memtas, mepocs, msexchange, MSExchange$, msftesql-Exchange, msmdsrv, MSSQL, MSSQL$, MSSQL$KAV_CS_ADMIN_KIT, MSSQL$MICROSOFT##SSEE, MSSQL$MICROSOFT##WID, MSSQL$SBSMONITORING, MSSQL$SHAREPOINT, MSSQL$VEEAMSQL2012, MSSQLFDLauncher$SBSMONITORING, MSSQLFDLauncher$SHAREPOINT, MSSQLServerADHelper100, MVArmor, MVarmor64, svc$, sophos, RTVscan, MySQL57, PDVFSService, QBCFMonitorService, QBFCService, QBIDPService, QBVSS, SavRoam, SQL, SQLADHLP, sqlagent, SQLAgent$KAV_CS_ADMIN_KIT, SQLAgent$SBSMONITORING, SQLAgent$SHAREPOINT, SQLAgent$VEEAMSQL2012, sqlbrowser, Sqlservr, SQLWriter, stc_raw_agent, tomcat6, veeam, VeeamDeploymentService, VeeamNFSSvc, VeeamTransportSvc, vmware-converter, vmware-usbarbitator64, VSNAPVSS, vss, wrapper, WSBExchange, YooBackup, YooIT*

* *https://www.microsoft.com/en-us/msrc/windows-security-servicing-criteria*

This is a common list of both end-user and server-side applications that may keep target files in use preventing their encryption. Finally, Mimic also includes code to terminate Hyper-V virtual machines using its PowerShell API.

3.3 File encryption scheme

Mimic Ransomware uses the well-known ChaCha20* algorithm to encrypt files and then protects the file encryption key with a second key, this time using asymmetric cryptography; this means that it is impossible to recover the encrypted files without the private key owned by the attackers.

The asymmetric mechanism used by this family is Diffie-Hellman key exchange over elliptic curves (ECDH**): it is an algorithm that enables two parties to establish a shared secret key over an insecure channel.
Each party generates a random key pair and exchanges their public keys over the insecure channel, then each party computes the shared secret key by multiplying the other party's public key by their own private key.

When referring to asymmetric cryptography, typically RSA comes to mind. The difference between RSA and Diffie-Hellman is that the latter only allows the two parties to derive the same secret key, which can subsequently be used for symmetric encryption. RSA, on the other hand, allows full encryption of a message with a public key in a way that can only be decrypted with the matching private key, however due to the complexity of calculations needed, it is typically used for key exchange.

On first launch the ransomware picks a public key at random from a hardcoded list and stores it in a file called *session.tmp*, we will refer to this key as the session key. The chosen session key is stored on disk so that the malware can resume encryption in case it gets interrupted halfway through. This key is also encoded as Base64 in the ID shown in the ransom note, this way the attacker knows which private key to use when producing a decryptor.

* *https://xilinx.github.io/Vitis_Libraries/security/2019.2/guide_L1/internals/chacha20.html*

** *https://datatracker.ietf.org/doc/html/rfc8418*

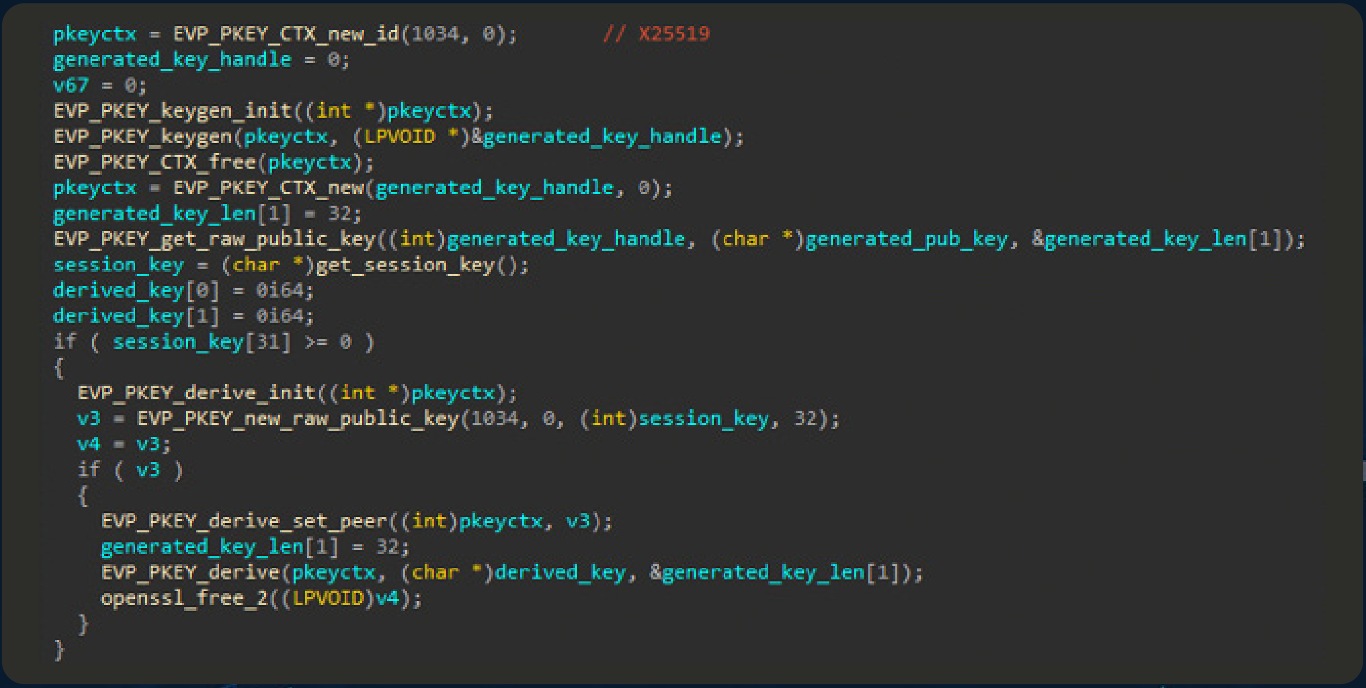
The malware proceeds to encrypt the files using multiple threads and using functions from the open-source OpenSSL library, which is statically linked in the binary. To reconstruct the meaning of each operation, we used a mix of function signatures, debugging and comparing the decompiler output with the source code of OpenSSL from the official repository.

For each file it generates a random 32-byte key using C++'s standard library random implementation, we'll refer to this key as the file key. Although this key is not generated using a cryptographic random number generator, recovering the files through a cryptographic attack is unlikely. The file key is used to encrypt the content of the file using the ChaCha20 algorithm.

To allow the decryption, once the ransom has been paid, an 80-byte block of custom metadata is appended to the end of the file. Before describing the metadata format, we need to analyse how the file encryption key is protected.

For each file, after the content has been encrypted, the malware generates a new *X25519* key pair, this is referred to as the *generated keypair*. *X25519* is the name of a standard elliptic curve. For a successful ECDH key exchange both parties must know the reference curve.

The sample performs ECDH using the session key (which is the attacker's public key) and the private part of the generated keypair, this produces a new key that we will call *derived key*.

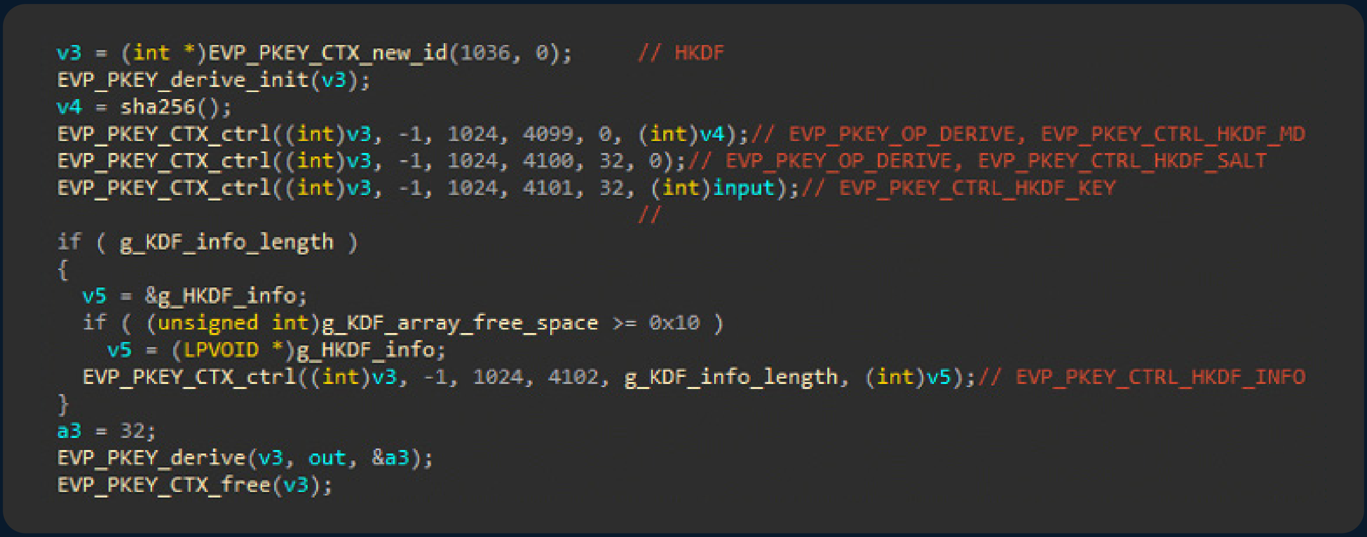


```
pkeyctx = EVP_PKEY_CTX_new_id(1034, 0);       // X25519
generated_key_handle = 0;
v67 = 0;
EVP_PKEY_keygen_init((int *)pkeyctx);
EVP_PKEY_keygen(pkeyctx, (LPVOID *)&generated_key_handle);
EVP_PKEY_CTX_free(pkeyctx);
pkeyctx = EVP_PKEY_CTX_new(generated_key_handle, 0);
generated_key_len[1] = 32;
EVP_PKEY_get_raw_public_key((int)generated_key_handle, (char *)generated_pub_key, &generated_key_len[1]);
session_key = (char *)get_session_key();
derived_key[0] = 0i64;
derived_key[1] = 0i64;
if ( session_key[31] >= 0 )
{
  EVP_PKEY_derive_init((int *)pkeyctx);
  v3 = EVP_PKEY_new_raw_public_key(1034, 0, (int)session_key, 32);
  v4 = v3;
  if ( v3 )
  {
    EVP_PKEY_derive_set_peer((int)pkeyctx, v3);
    generated_key_len[1] = 32;
    EVP_PKEY_derive(pkeyctx, (char *)derived_key, &generated_key_len[1]);
    openssl_free_2((LPVOID)v4);
  }
}
```

Figure 1. ECDH key exchange code

The derived key is used to generate the final key using HKDF* with SHA256 as key derivation function (KDF). This final key is used to protect the custom meta-data, we call this the meta key.

Key derivation functions are typically used after ECDH, as a random number generator seeded with the shared key to produce multiple keys from a single key exchange. In practice, it's not necessarily needed for this ransomware, it's safe to assume that the developers copied this step from an example showing good practices when working with ECDH.



```
v3 = (int *)EVP_PKEY_CTX_new_id(1036, 0);     // HKDF
EVP_PKEY_derive_init(v3);
v4 = sha256();
EVP_PKEY_CTX_ctrl((int)v3, -1, 1024, 4099, 0, (int)v4);// EVP_PKEY_OP_DERIVE, EVP_PKEY_CTRL_HKDF_MD
EVP_PKEY_CTX_ctrl((int)v3, -1, 1024, 4100, 32, 0);// EVP_PKEY_OP_DERIVE, EVP_PKEY_CTRL_HKDF_SALT
EVP_PKEY_CTX_ctrl((int)v3, -1, 1024, 4101, 32, (int)input);// EVP_PKEY_CTRL_HKDF_KEY
                                              //
if ( g_KDF_info_length )
{
  v5 = &g_HKDF_info;
  if ( (unsigned int)g_KDF_array_free_space >= 0x10 )
    v5 = (LPVOID *)g_HKDF_info;
  EVP_PKEY_CTX_ctrl((int)v3, -1, 1024, 4102, g_KDF_info_length, (int)v5);// EVP_PKEY_CTRL_HKDF_INFO
}
a3 = 32;
EVP_PKEY_derive(v3, out, &a3);
EVP_PKEY_CTX_free(v3);
```

Figure 2. HKDF Derivation

The HKDF can take additional parameters (called info in OpenSSL) to increase entropy; the build of Mimic that we analysed also adds the string "DontDecompileMePlease", since this code path is guarded by an "if", it is likely to be a configurable option when compiling the malware. Knowing this extra string is also needed to decrypt the files.

* https://su21.cs161.org/proj2/crypto/hkdf.html

At this point the malware generates and writes the metadata to the file, the format is described in the following table:

Offset (hex)	Length (hex)	Description
+ 0x00	0x20	Unencrypted public key of the generated keypair
+ 0x20	0x20	File key encrypted using ChaCha20 and the meta key
+ 0x40	0x8	Original length of the file as a Windows LARGE_INTEGER structure encrypted using ChaCha20 and the meta key
+ 0x48	0x1	Single byte ranging from 0 to 100 representing the progress of the encryption process of the system encrypted using ChaCha20 and the meta key
+ 0x49	0x1	Single byte representing the ASCII character '2' encrypted using ChaCha20 and the meta key
+ 0x4A	0x6	ASCII string "isKey" encrypted using ChaCha20 and the meta key

Table 1. Format of metadata

Every field shown in Table 1 is encrypted using an individual call to the same encryption function used for the actual file data, this may be a cryptographic flaw, but it's unlikely that it can be exploited to recover the whole 32 bytes of the meta key needed to decrypt the file key and subsequently the actual file data.

We don't believe recovery of encrypted data is possible without the attackers' private key. Moreover, since the file key is randomised for each file, dumping the malware while the encryption process is running does not allow to recover the keys used in files already encrypted previously.

3.4 IOC

Mimic ransomware will download the following legitimate signed files:

- **Everything64.dll**
- **Everything.exe**

These are parts of the file search tool Everything and its SDK.

Mimic will associate the extension used to encrypt files to a link opening the ransom note text file, this is done by defining a class named "mimicfile" in the registry, the full path of the key is *"HKEY_CLASSES_ROOT\mimicfile"*.

The malware deletes itself after the encryption has finished, only the following files are left:

- ***Session.tmp*****, containing the public key used by the ransomware.**
- **Mimic_log.txt, containing a log of all the files that were encrypted.**

DEFENCE TECH

4

Conclusions

4. Conclusions

Asymmetric encryption has become common practice in ransomware to force victims to pay the ransom rather than relying on security experts to recover the files.

Mimic in particular features the use of multiple threads and the "Everything" file search engine considerably speeding up the data encryption process and backup deleting instructions making it a dangerous malware for end-users and companies alike.

Our recommendation is to implement some best practices in order to guarantee data safety against encryption, such as isolated backup and recovery measures.

It's also important to conduct vulnerability assessments regularly to mitigate known CVEs and detect system misconfigurations.

Furthermore, in this case it seems the initial access to the machine was gained through phishing, we suggest putting in place tools and measures to monitor ingoing emails, including suspicious attachment, for example, password protected archives.

Finally, it is fundamental to perform security and phishing awareness training campaigns, in order to improve the security posture in the company.

DEFENCE TECH

Terra, Cielo, Mare, Spazio, Spazio cibernetico.
PROTEGGIAMOLI

DONEXIT IT SECURITY

NEXT INGEGNERIA DEI SISTEMI

FORAMIL RADAR TECHNOLOGIES & DEFENCE SYSTEMS

INNODESI electronic systems

Defence Tech Holding S.p.A Società Benefit

Via Giacomo Peroni, 452 - 00131 Roma
tel. 06.45752720 - fax 06.45752721
info@defencetech.it - www.defencetech.it