

# AtomSilo Ransomware Enters the League of Double Extortion | Zscaler

By Rajdeepsinh Dodia

Published: 2021-10-15 · Archived: 2026-04-05 21:05:38 UTC

Ransomware is used widely in cyberattacks to disrupt the victim's organization. Over the last two years, many attackers have evolved their ransomware tactics to include data exfiltration. This tactic is known as "[double-extortion](#)": attackers demand ransom for the data decryption in addition to the ransom to prevent public release of the stolen data. ThreatLabz monitors these threat actors and analyzes the attack sequences of double extortion attacks. AtomSilo is a new player on the scene, and in this blog, we'll break down the details of their attacks.

## Introduction

AtomSilo ransomware emerged around September 2021, with their tactics including exfiltrating and publishing their first victim's data.

We'll break down one of their attacks, which started with initial access through exploiting a vulnerability in [Atlassian's Confluence collaboration software](#). The ransomware operators planted a back door using legitimate software via a dll side loading technique. The backdoor allowed remote code execution of Windows Shell commands through WMI (Windows Management Interface), which operators exploited using compromised administrative accounts before dropping AtomSilo.

## Technical Analysis

The AtomSilo payload is 64-bit and packed with a modified UPX packer. Once executed, it enumerates each drive and drops a ransom note in each folder except the few listed in *Table1*. The ransom note is named "README-FILE-{COMPUTER\_Name}-{DateTime}.hta".

**Atom Silo**  
Instructions

---

**WARNING! YOUR FILES ARE ENCRYPTED AND LEAKED!**

---

We are AtomSilo.Sorry to inform you that your files has been obtained and encrypted by us.

But don't worry, your files are safe, provided that you are willing to pay the ransom.

Any forced shutdown or attempts to restore your files with the thrid-party software will be **damage your files permanently!**

The only way to decrypt your files safely is to buy the special decryption software from us.

The price of decryption software is **1000000 dollars.**

If you pay within 48 hours, you only need to pay **500000 dollars.** No price reduction is accepted.

We only accept Bitcoin payment,you can buy it from bitpay,coinbase,binance or others.

You have five days to decide whether to pay or not. After a week, we will no longer provide decryption tools and publish your files

---

**Time starts at 0:00 on September 11**

---

**Survival time: -23 Day -1 Hour -55 Min -24 Sec**

---

You can contact us with the following email:

Email: [atomsilo@atomsilo.com](mailto:atomsilo@atomsilo.com)

If this email can't be contacted, you can find the latest email address on the following website:

<http://www.atomsilo.com>

Figure 1: AtomSilo ransom note

It enumerates each file and encrypts all folders and files EXCEPT those that contain the below names:

Folder name	File name
Boot	autorun.inf
Windows	index.html
Windows.old	boot.ini
Tor Browser	bootfont.bin
Internet Explorer	bootsect.bak
Google	bootmgr
Opera	bootmgr.efi
Opera Software	bootmgfw.efi
Mozilla	desktop.ini
Mozilla Firefox	iconcache.db

Folder name	File name
\$recycle.Bin	ntldr
ProgramData	ntuser.dat
All Users	ntuser.dat.log
	#recycle
	thumbs.db
	ntuser.ini

Table1: List of files and folders

It also does not encrypt files with the following extensions:

.hta	.idx
.hlp	.ini
.html	.sys
.icl	.cab
.exe	.spl
.icns	.cur
.dll	.ocx
.ico	.cpl
.cpl	.drv

Table2: List of extensions

## File Encryption

Ransomware appends “.atomsilo” extensions to files after encryption. Ransomware uses “CreateFileMappingA” and “MapViewOfFile” APIs to map the file in memory and moves the pointer to the start of the mapped file. AtomSilo uses XOR and AES Encryption algorithms to encrypt files. It generates AES round keys using the “AESKEYGENASSIST” instruction as shown in the below figure.

```

000013F76F826 89 4B 2C mov dword ptr ds:[rbx+2C], ecx
000013F76F829 66 0F 3A 22 F1 03 pinsrd xmm6, ecx, 3
000013F76F82F  EB 46 jmp d9f7bb98ad01c4775ec71ec6.13F76F877
000013F76F831 48 83 FF 20 cmp rdi, 20
000013F76F835 75 39 jne d9f7bb98ad01c4775ec71ec6.13F76F870
000013F76F837 66 0F 3A 22 73 2C pinsrd xmm6, dword ptr ds:[rbx+2C], 3
000013F76F83E 8B 4B 14 mov ecx, dword ptr ds:[rbx+14]
000013F76F841 66 0F 3A DF C6 00 aeskeygenassist xmm0, xmm6, 0
000013F76F847 66 0F 3A 16 C0 02 pextrd eax, xmm0, 2
000013F76F84D 33 43 10 xor eax, dword ptr ds:[rbx+10]
000013F76F850 33 C8 xor ecx, eax
000013F76F852 89 43 30 mov dword ptr ds:[rbx+30], eax
000013F76F855 8B 43 18 mov eax, dword ptr ds:[rbx+18]
000013F76F858 33 C1 xor eax, ecx
000013F76F85A 89 4B 34 mov dword ptr ds:[rbx+34], ecx
000013F76F85D 8B 4B 1C mov ecx, dword ptr ds:[rbx+1C]
000013F76F860 33 C8 xor ecx, eax
000013F76F862 89 43 38 mov dword ptr ds:[rbx+38], eax
000013F76F865 89 4B 3C mov dword ptr ds:[rbx+3C], ecx
000013F76F868 66 0F 3A 22 F1 03 pinsrd xmm6, ecx, 3
000013F76F86E  EB 07 jmp d9f7bb98ad01c4775ec71ec6.13F76F877
000013F76F870 66 0F 3A 22 73 1C pinsrd xmm6, dword ptr ds:[rbx+1C], 3
000013F76F877 49 8B D9 mov rbx, r9
000013F76F87A 66 0F 3A DF C6 00 aeskeygenassist xmm0, xmm6, 0
000013F76F880 66 0F 3A 16 C1 03 pextrd ecx, xmm0, 3
000013F76F886 33 0A xor ecx, dword ptr ds:[rdx]
000013F76F888 48 83 C2 04 add rdx, 4
000013F76F88C 41 33 09 xor ecx, dword ptr ds:[r9]
000013F76F88F 4D 03 CA add r9, r10
000013F76F892 41 89 09 mov dword ptr ds:[r9], ecx
    
```

41 d9f7bb98ad01c4775ec71ec6.exe:\$3F841 #0

Hex	ASCII
66 68 7A AD F8 62 BD 77 6C 8F C1 8B 8	fhz.ob½wl.A...
08 97 14 85 6E E2 33 B3 90 2A 59 1D 0	....nâ3³.*Y..)%
A8 CD 45 7A 50 AF F8 0D 3C 20 39 86 B	IEZPø.<9.²¿!
3F 9F BD A1 51 7D 8E 12 C1 57 D7 0F C	?½jQ}..Awx.İ.b*
9A 76 A0 31 CA D9 58 3C F6 F9 61 BA 4	.v 1EUX<oua°DFÖ.
0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0	0.ð.º.ð.º.ð.º.ð.º
0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0	0.ð.º.ð.º.ð.º.ð.º
0D F0 AD BA 0D F0 AD BA 0D F0 AD BA 0	0.ð.º.ð.º.ð.º.ð.º

Generating key

Figure 2: AtomSilo generates encryption keys using AESKEYGENASSIST

The encryption key is 240 bytes. The first 32 bytes are randomly generated by the payload, and other 208 bytes are generated using the “AESKEYGENASSIST” instruction. In the file , it takes 16 bytes of plain text and does XOR as a first stage encryption. Then, it encrypts it with 14 rounds of AES encryption. It uses “AESENC” instruction for the first 13 rounds and the last round uses “AESENCLAST” instruction.

The screenshot displays assembly code for an AES encryption routine. Key instructions include:

- `movdq xmm0, xmmword ptr ds:[rcx]` (Load data into XMM0)
- `lea r11d, qword ptr ds:[r8-1]` (Calculate address for first round XOR)
- `pxor xmm0, xmmword ptr ds:[rdx]` (XOR with plain text)
- `mov r9d, 1` (Set round counter)
- `movdq xmmword ptr ds:[rcx], xmm0` (Store XMM0 back to memory)
- `mov r10, rcx` (Set pointer to current block)
- `cmp r11d, r9d` (Compare round counter)
- `jbe d9f7bb98ad01c4775ec71ec6.13F76F67C` (Jump if below or equal, indicating end of rounds)
- `mov eax, r9d` (Get current round)
- `lea ecx, qword ptr ds:[r9+1]` (Increment round counter)
- `add rax, rax` (Increment pointer to next block)
- `add rcx, rcx` (Increment pointer to current block)
- `add r9d, 2` (Increment round counter)
- `aesenc xmm0, xmmword ptr ds:[rdx+rcx*8]` (Perform AES encryption on current block)
- `movdq xmmword ptr ds:[r10], xmm0` (Load next block)
- `aesenc xmm0, xmmword ptr ds:[rdx+rcx*8]` (Perform AES encryption)
- `movdq xmmword ptr ds:[r10], xmm0` (Store encrypted block)
- `cmp r9d, r11d` (Compare round counter)
- `jb d9f7bb98ad01c4775ec71ec6.13F76F650` (Jump if below, indicating end of rounds)
- `mov eax, r11d` (Get current round)
- `add rax, rax` (Increment pointer to next block)
- `aesenc xmm0, xmmword ptr ds:[rdx+rcx*8]` (Perform AES encryption)
- `movdq xmmword ptr ds:[r10], xmm0` (Load next block)
- `mov eax, r8d` (Get pointer to current block)
- `add rax, rax` (Increment pointer to next block)
- `aesencld xmm0, xmmword ptr ds:[rdx+rcx*8]` (Perform AES encryption)
- `movdq xmmword ptr ds:[r10], xmm0` (Store encrypted block)
- `ret` (Return)

The register window shows XMM0-XMM15, and the register dump shows:

```

1: rcx 0000000000000004
2: rdx 0000000001E2D20 "L2F90sauT06vjF8
3: r8 000000000000000E
4: r9 0000000000000003
5: [rsp+28] 0000000000000000
    
```

Figure 3: Encrypting data using AES algorithm

It encrypts chunks of the file, not the complete file. It encrypts the first 16 bytes, leaves the next 32 bytes as-is, encrypts the next 16 bytes, and so on. The below screenshot shows the comparison of the normal file and encrypted file, where we can see that chunks of files are not encrypted. The encryption key and other information are encrypted and appended at the end of the encrypted file.

The screenshot compares the original file content with the encrypted file content. The original file contains text such as:

```

75 69 74 61 62 6C 65 20 28 61 6D 6F 6E 67 73 74
20 6F 74 68 65 72 20 75 73 65 73 29 20 66 6F 72
20 64 69 73 74 72 69 62 75 74 65 64 20 61 70 70
6C 69 63 61 74 69 6F 6E 0D 0A 64 65 76 65 6C 6F
70 6D 65 6E 74 2C 20 73 63 72 69 70 74 69 6E 67
2C 20 6E 75 6D 65 72 69 63 20 63 6F 6D 70 75 74
69 6E 67 20 61 6E 64 20 73 79 73 74 65 6D 20 74
65 73 74 69 6E 67 2E 20 50 79 74 68 6F 6E 0D
0A 69 73 20 6F 66 74 65 6E 20 63 6F 6D 70 61 72
65 64 20 74 6F 20 54 63 6C 2C 20 50 65 72 6C 2C
20 4A 61 76 61 2C 20 4A 61 76 61 53 63 72 69 70
74 2C 20 56 69 73 75 61 6C 20 42 61 73 69 63 20
6F 72 0D 0A 53 63 68 65 6D 65 2E 20 20 54 6F 20
66 69 6E 64 20 6F 75 74 20 6D 6F 72 65 20 61 62
6F 75 74 20 77 68 61 74 20 50 79 74 68 6F 6E 20
63 61 6E 20 64 6F 20 66 6F 72 20 79 6F 75 2C 20
70 6F 69 6E 74 20 79 6F 75 72 0D 0A 62 72 6F 77
73 65 72 20 74 6F 20 68 74 74 70 3A 2F 2F 77 77
77 2E 70 79 74 68 6F 6E 2E 6F 72 67 2F 2E
    
```

The encrypted file shows the same text but with some parts (like the key and other information) encrypted into gibberish. A legend at the bottom indicates:

- Yellow: Not encrypted
- Red: Encrypted
- Blue: Encrypted Key and other information

Figure 4: Original vs Encrypted file

## Data Leak site

According to their leak sites, AtomSilo actors won't attack the following types of organizations:

- Hospitals.
- Critical infrastructure facilities (nuclear power plants, power plants, water treatment facilities).
- Oil and gas industry (pipelines, oil refineries).
- Educational unit.
- Non-profit companies.

They also promise to provide free decryption if the victim company is on the above list.

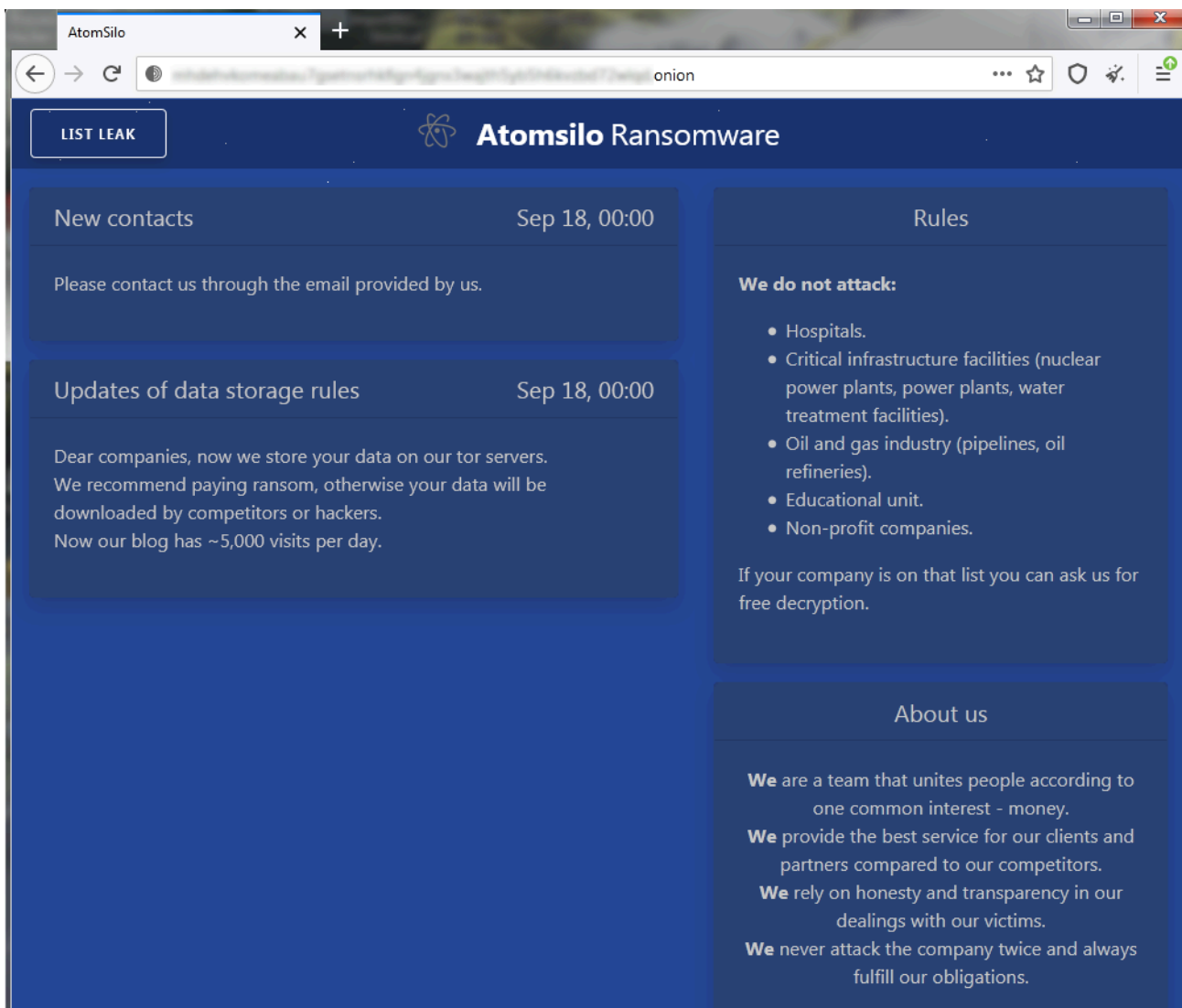


Figure 5: Data leak site

The first data leak was from a Brazilian Pharmaceutical company. AtomSilo published around 900 GB data as shown in the below screenshot:

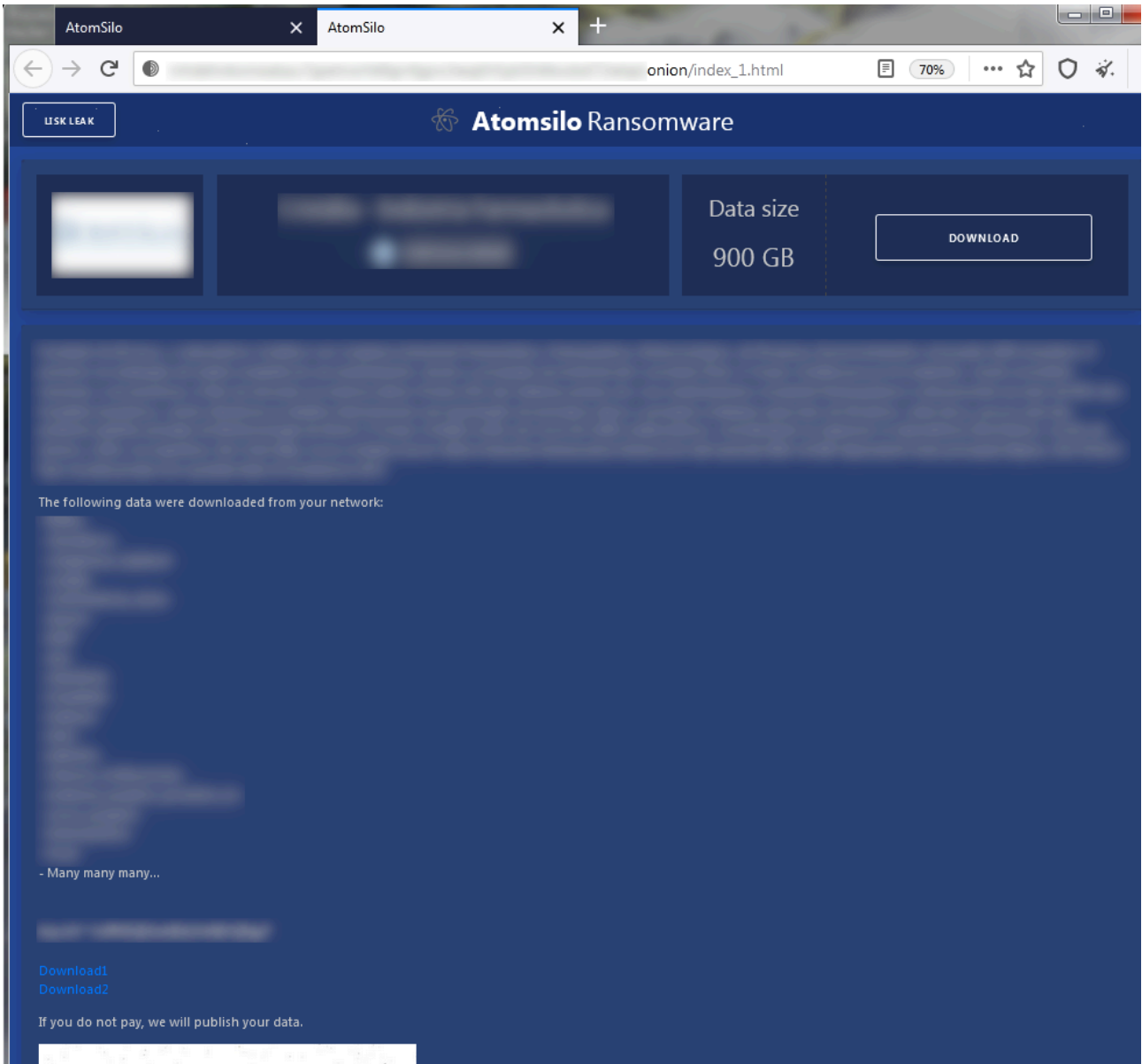


Figure 6: Victim data published on data leak site

## Cloud Sandbox Detection

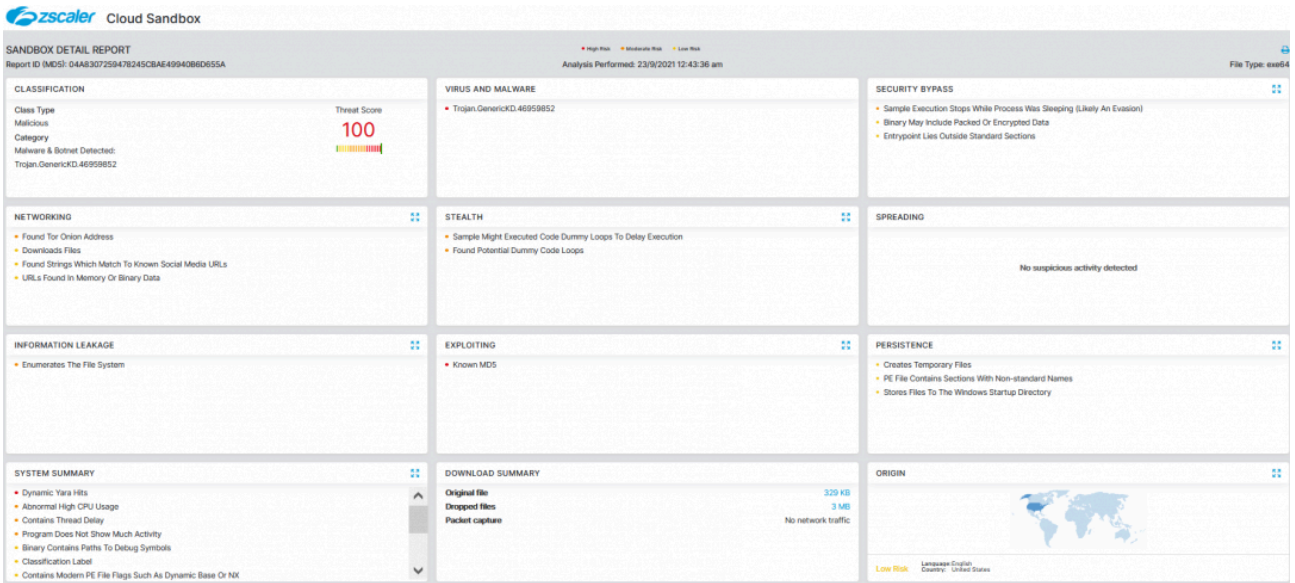


Figure 7: Zscaler Cloud Sandbox detection of AtomSilo ransomware

In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects indicators at various levels.

[Win64.Ransom.AtomSilo](#)

## IOC

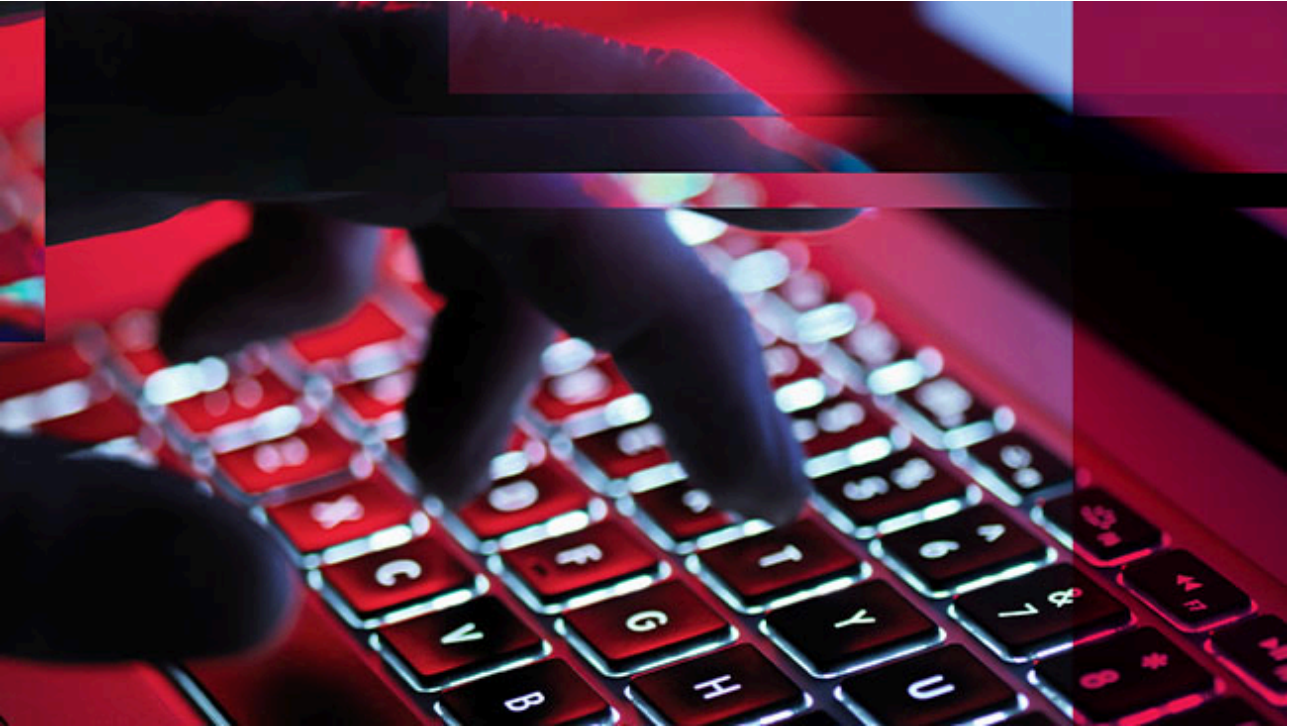
### Md5

04a8307259478245cbae49940b6d655a

## Explore more Zscaler blogs

Zscaler ThreatLabz 2024 Phishing Report

The Threat Prevention Buyer's Guide



---

Source: <https://www.zscaler.com/blogs/security-research/atomsilo-ransomware-enters-league-double-extortion>