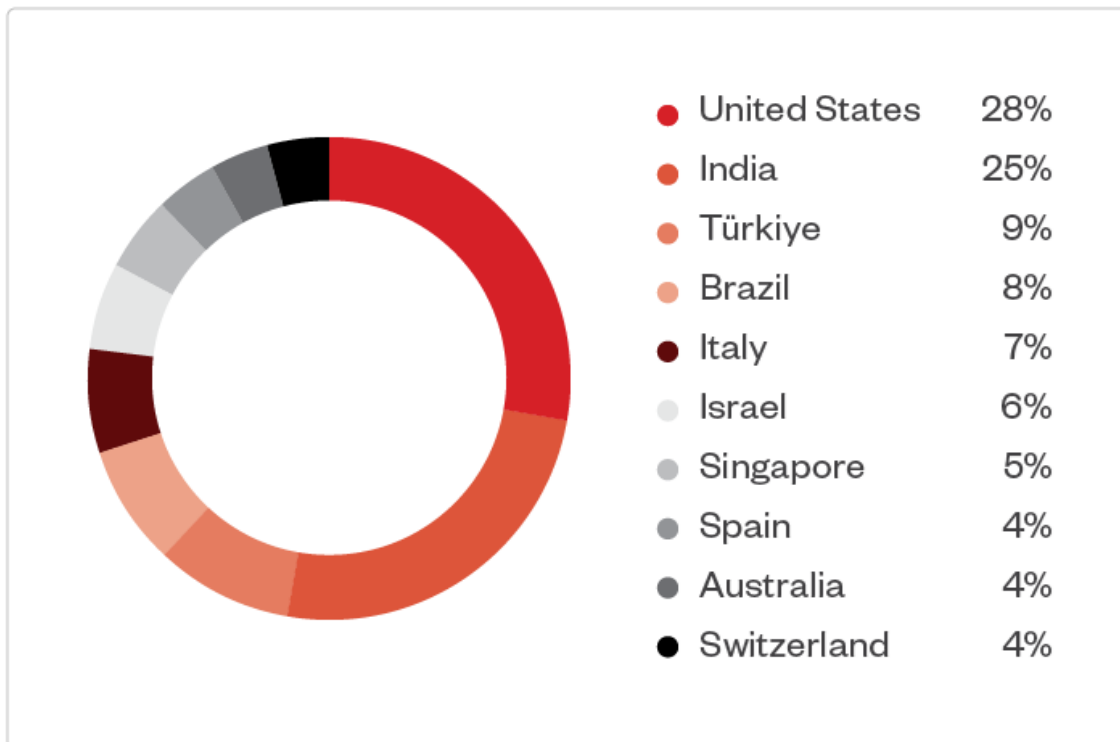


# An Overview of the Different Versions of the Trigona Ransomware



©2023 TREND MICRO

Figure 1. Trigona ransomware detections based on country

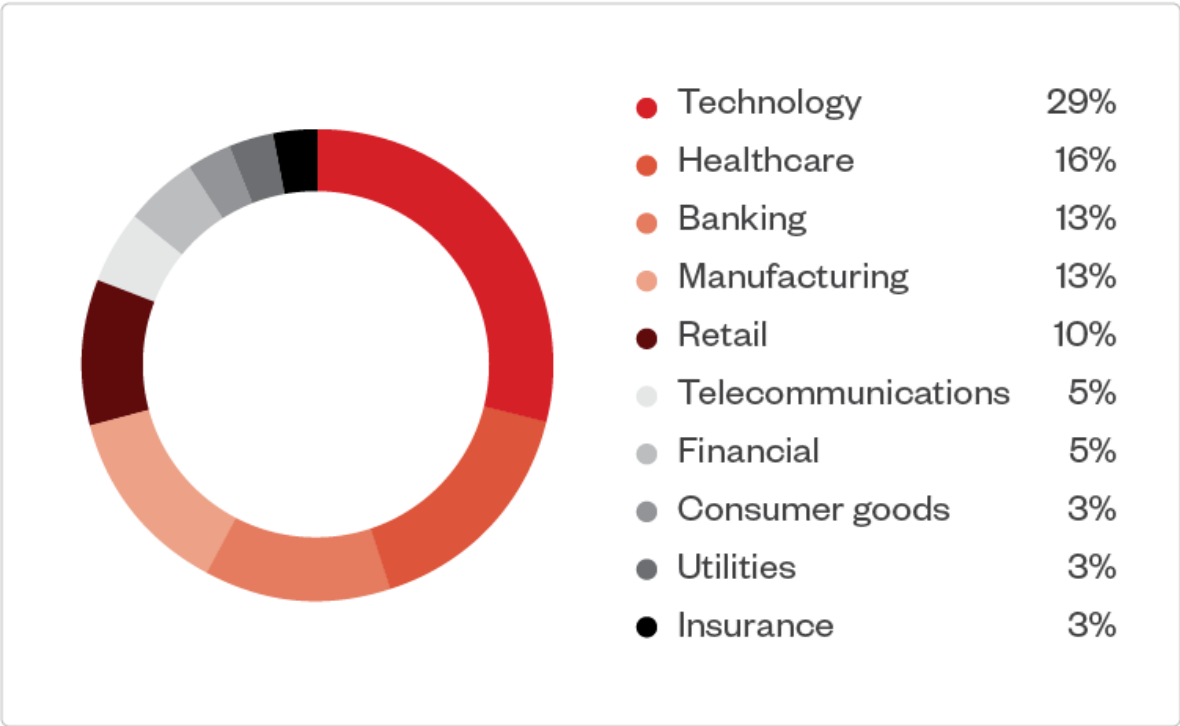
The Trigona ransomware is a relatively new ransomware family that began activities around late October 2022 — although samples of it existed as early as June 2022. Since then, Trigona’s operators have remained highly active, and in fact have been continuously updating their ransomware binaries. By April 2023, Trigona began targeting compromised MSSQL servers by stealing credentials via brute force methods. In May 2023, we found a Linux version of Trigona that shared similarities with its Windows counterpart.

The threat actors behind Trigona are allegedly the same group behind the CryLock ransomware due to similarities in tools, tactics, and procedures (TTPs). It has also been linked to the ALPHV group (also known as BlackCat), though we believe that any similarities between Trigona and BlackCat ransomware are only circumstantial at best (one possibility is that ALPHV collaborated with the threat actors deploying Trigona but were not actually involved with its development and operation).

## Target countries and industries

Based on Trend Micro™ Smart Protection Network™ data, US and India were the countries with the highest number of Trigona ransomware detections, with Israel, Turkey, Brazil, and Italy also having a significant count.

Meanwhile, attacks focused mainly on the technology and healthcare industries, which had the highest number of detections.



©2023 TREND MICRO

Figure 1. Trigona ransomware detections based on industry  
Infection chain

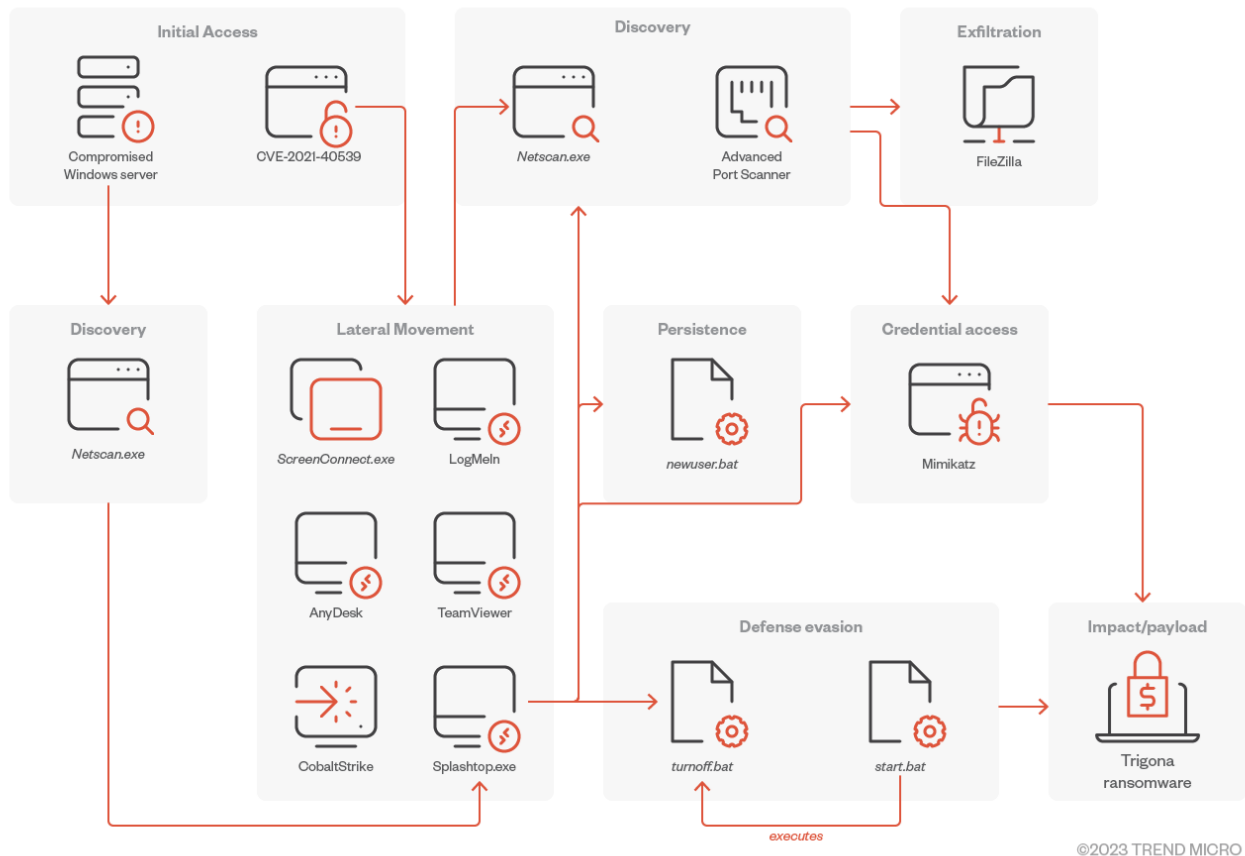


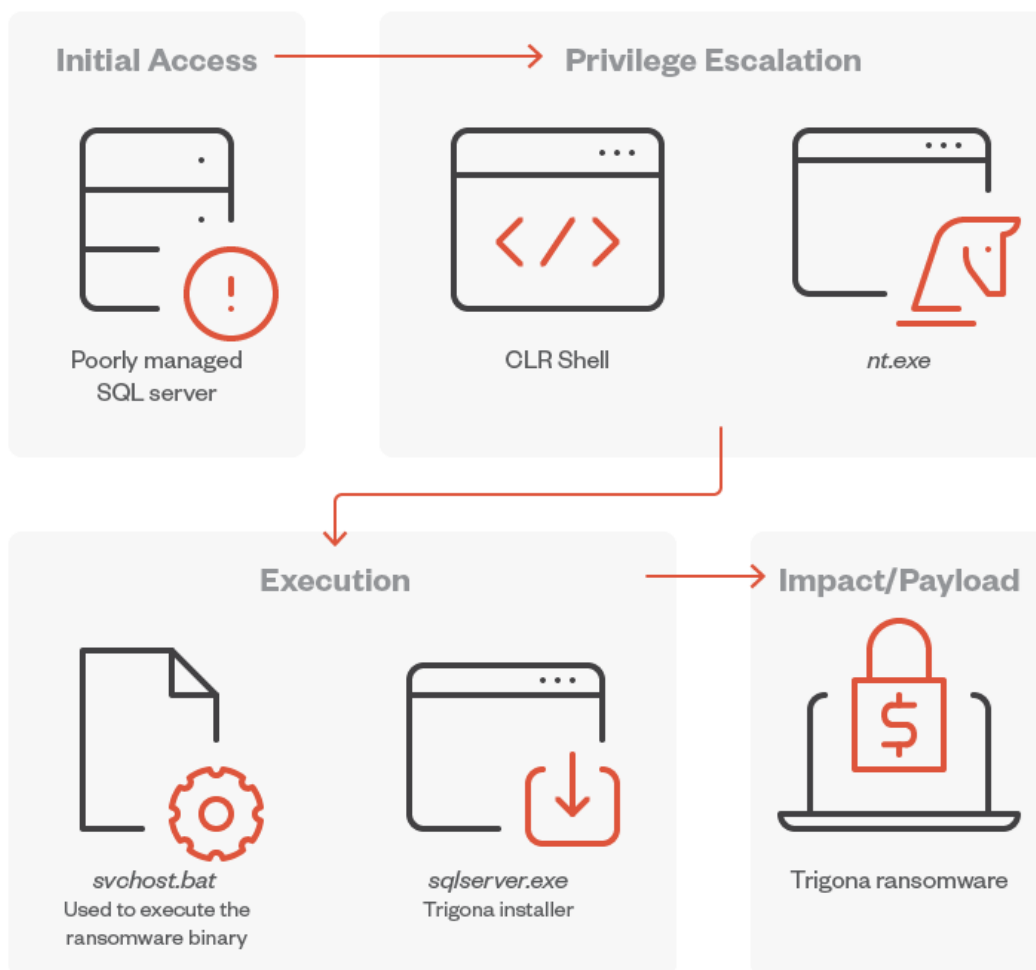
Figure 3. The Trigona ransomware infection chain (based on Palo Alto’s analysis of Trigona)

Trigona was found to be exploiting the ManageEngine vulnerability [CVE-2021-40539](#) for initial access based on a [report from Arete](#). In addition, the threat actors used previously compromised accounts by obtaining access from network access brokers.

It uses a variety of tools for lateral movement, including Splashtop (a legitimate remote access tool), which is used to drop further additional tools on a compromised machine.

Trigona drops a file called *turnoff.bat* (detected as Trojan.BAT.TASKILL.AE) to terminate AV-related services and processes. It also uses Network Scanner and Advanced Port Scanner to identify network connections.

Based on [AhnLab’s analysis](#), Trigona’s operators use CLR shell on attacks launched against MS-SQL servers. This tool is capable of multiple commands, including one that drops additional executables for privilege escalation (*nt.exe*).



©2023 TREND MICRO

Figure 4. Infection chain for compromised SQL server (Based on AhnLab's analysis)

Trigona encrypts files in infected machines using AES encryption. Furthermore, the ransomware contains an encrypted configuration in its resource section which is decrypted upon execution. However, it will only use certain strings within its configuration. Trigona also randomizes the file names of encrypted files and appends the `._locked` extension upon encryption.

Trigona's operators employ the credential dumper Mimikatz to gather the passwords and credentials found on the machines of the victims.

## Linux Version

In May 2023, our threat hunting team found a Linux ransomware binary that had a sparse number of detections. Upon further verification, we confirmed these binaries to be a Linux version of Trigona. Like its 32-bit Windows counterpart, this binary accepts command-line arguments for execution.

```

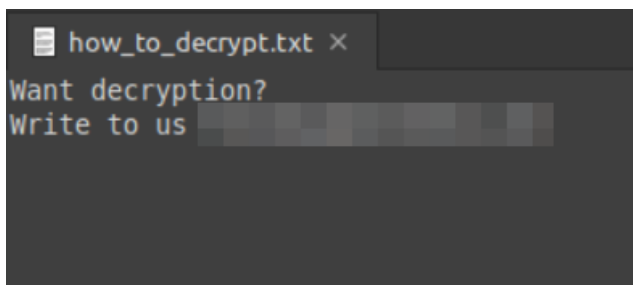
if ( !LOADCONFIGSFROMRES() )
{
    v11 = fpc_get_output();
    fpc_write_text_shortstr(0LL, v11, _PROJECT1__Ld3);
    fpc_iocheck();
    fpc_writeln_end(v11);
    fpc_iocheck();
    goto LABEL_58;
}
v12 = fpc_get_output();
fpc_write_text_shortstr(0LL, v12, &_PROJECT1__Ld4);
fpc_iocheck();
fpc_writeln_end(v12);
fpc_iocheck();
if ( ASLINUX_PARAMCOUNT() == 2 )
{
    fpc_ansi_str_decr_ref(PARAM);
    *PARAM = 0LL;
    ASLINUX_PARAMSTR(PARAM);
    if ( fpc_ansi_str_compare_equal(*PARAM, _PROJECT1__Ld5) )// /full
    {
        fpc_ansi_str_decr_ref(PARAM);
        *PARAM = 0LL;
        ASLINUX_PARAMSTR(PARAM);
        if ( fpc_ansi_str_compare_equal(*PARAM, _PROJECT1__Ld6) )// /erase
        {
            fpc_ansi_str_decr_ref(PARAM);
            *PARAM = 0LL;
            ASLINUX_PARAMSTR(PARAM);
            if ( fpc_ansi_str_compare_equal(*PARAM, _PROJECT1__Ld7) )// /is_testing
            {
                fpc_ansi_str_decr_ref(PARAM);
                *PARAM = 0LL;
                ASLINUX_PARAMSTR(PARAM);
                if ( fpc_ansi_str_compare_equal(*PARAM, _PROJECT1__Ld8) )// /test_cid
                {
                    fpc_ansi_str_decr_ref(PARAM);
                    *PARAM = 0LL;
                    ASLINUX_PARAMSTR(PARAM);
                    if ( fpc_ansi_str_compare_equal(*PARAM, _PROJECT1__Ld9) )// /test_vid
                    {
                        fpc_ansi_str_decr_ref(PARAM);
                        *PARAM = 0LL;
                    }
                }
            }
        }
    }
}

```

\_Figure 5. Code snippet showing

command-line arguments from the Linux version of Trigona

The ransom note dropped by the binary (*how\_to\_decrypt.txt*) contains only an email address of the threat actor behind the attack. This may indicate that the Linux version is still a work in progress.



\_Figure 6. Ransom note dropped by the Linux version of

Trigona

## Windows 64-bit version

In June 2023, we encountered a new version of Trigona ransomware, this time designed for Windows 64-bit platforms. This version implements additional command-line arguments that were not present with the Linux version and the original 32-bit version (such as */sleep* and */debug*).

```

if ( *(off_584B48 + 308) )
{
    LOBYTE(v7) = 1;
    sub_542890(L"Read options.", v7);
}
if ( sub_4098A0() == 1 )
{
    getcmdline_409920(&vars1E8, 1i64);
    if ( parsecmdline_412D10(vars1E8, L"/r" ) )
    {
        getcmdline_409920(&vars1E0, 1i64);
        if ( parsecmdline_412D10(vars1E0, L"/sleep" ) )
        {
            getcmdline_409920(&vars1D8, 1i64);
            if ( parsecmdline_412D10(vars1D8, L"/full" ) )
            {
                getcmdline_409920(&vars1D0, 1i64);
                if ( parsecmdline_412D10(vars1D0, L"/debug" ) )
                {
                    getcmdline_409920(&vars1C8, 1i64);
                    if ( parsecmdline_412D10(vars1C8, L"/log_f" ) )
                    {
                        getcmdline_409920(&vars1C0, 1i64);
                        if ( parsecmdline_412D10(vars1C0, L"/fast" ) )
                        {
                            getcmdline_409920(&vars1B8, 1i64);
                            if ( parsecmdline_412D10(vars1B8, L"/erase" ) )
                            {
                                getcmdline_409920(&vars1B0, 1i64);
                                if ( parsecmdline_412D10(vars1B0, L"/!autorun" ) )

```

Figure 7. Snippet showing

command-line arguments from the 64-bit Windows version of Trigona

## Command-line arguments

Table 1 summarizes the command-line arguments used by each of the different versions of Trigona:

| 32-bit Windows | 64-bit Windows | Linux       | Description   |
|----------------|----------------|-------------|---|
| /r             | /r             |             | Allows the encryption of files in a random order  |
| /full          | /full          | /full       | Encrypt the whole content of the target file (if not used, only the first 0x80000 bytes/512kb are encrypted)            |
| /erase         | /erase         | /erase      | Deletes the content of the target files. (By default, only the first 512kb is erased unless the argument /full is used) |
| /!autorun      | /!autorun      |             | Does not create the autorun registry entry.   |
| /is_testing    | /is_testing    | /is_testing | Used with /test_cid and /test_vid for testing purposes  |
| /test_cid      | /test_cid      | /test_cid   | Uses the specified Computer ID instead of generating one  |
| /test_vid      | /test_vid      | /test_vid   | Uses the specified Victim ID instead of the one in the configurations   |
| /p             | /p             | /p          | Specifies the path to encrypt   |

|               |               |           |  |
|---------------|---------------|-----------|--|
| /path         | /path         | /path     | Specifies the path to encrypt  |
| !/local       | !/local       |           | Avoids encrypting local files  |
| !/lan         | !/lan         |           | Avoids encrypting network shares   |
| /shdwn        | /shdwn        | /shutdown | Forces shutdown of the machine after encryption  |
| /autorun_only | /autorun_only |           | Creates an autorun registry that will execute the ransomware upon logon. This will not perform the encryption yet. |
|               | /sleep        |           | Sleeps for n seconds before execution  |
|               | /debug        |           | Executes in debug mode, need to be executed with /p  |
|               | /log_f        |           | specifies the log file for logging   |
|               | /fast         |           |  |
|               | /allow_system |           | Allows encryption of files in the system directory   |

Table 1. Command-line arguments used by each Trigona version

## Encryption

All versions of Trigona employ *TDCP\_rijndael* (AES) to encrypt the target files depending on the configurations set in its resource section.

```

if ( !v17 )
{
    EXTENDCONTROLDATAIFWANTED(p_CDa, *p_CDLEN);
    SYSTEM_FILLCHAR_formal_INT64_BYTE(*p_BLOCK, PARTSIZEa, 0LL);
    RWB = 0;
    if ( ASLINUX_SETFILEPOINTEREX(*p_FAa, CFPa, 0) )
        ASLINUX_READFILE(*p_FAa, *p_BLOCK, PARTSIZEa, &RWB);
    for ( I = RWB - 1; I >= 0 && !>(*p_BLOCK + I); --I )
        ;
    RWB = ++I;
    if ( !I )
        goto LABEL_9;
    PARTSIZEa = RWB;
    AES_OFB_ENCRYPT(p_BLOCK, p_PASSa, p_IVa, p_EBLOCK, RWB, p_DCP);
    RWB = 0;
    if ( ASLINUX_SETFILEPOINTEREX(*p_FAa, CFPa, 0) )
        ASLINUX_WRITEFILE(*p_FAa, *p_EBLOCK, PARTSIZEa, &RWB);
    if ( RWB )
    {
        ENCRYPTION_LOG.ENCRYPTED_IN_FILE += RWB;
        PARTSIZEa = RWB;
        v15 = NUMTOARRAY(RWB);
        fpc_dynarray_decr_ref(&LAR, &INIT_BASETYPES_TCHANGEBLEARRAY);
        LAR = v15;
        *(*p_CDa + *p_CDLEN) = **p_BLOCK;
        *(*p_CDa + *p_CDLEN + 1) = *(*p_BLOCK + PARTSIZEa - 1);
        *(*p_CDa + *p_CDLEN + 2) = *(LAR + 1);
        *(*p_CDa + *p_CDLEN + 3) = *(LAR + 2);
        *(*p_CDa + *p_CDLEN + 4) = *(LAR + 3);
        *p_CDLEN += 5;
        v16 = 0LL;
        fpc_dynarray_setlength(&LAR, &INIT_BASETYPES_TCHANGEBLEARRAY, 1LL, &v16);
    }
}

```

\_Figure 8. The Linux version of

Trigona using AES for encryption

Encrypted files are either renamed with encrypted strings or with an additional prepended string *available\_for\_trial*, then appended by the *.\_locked* extension.

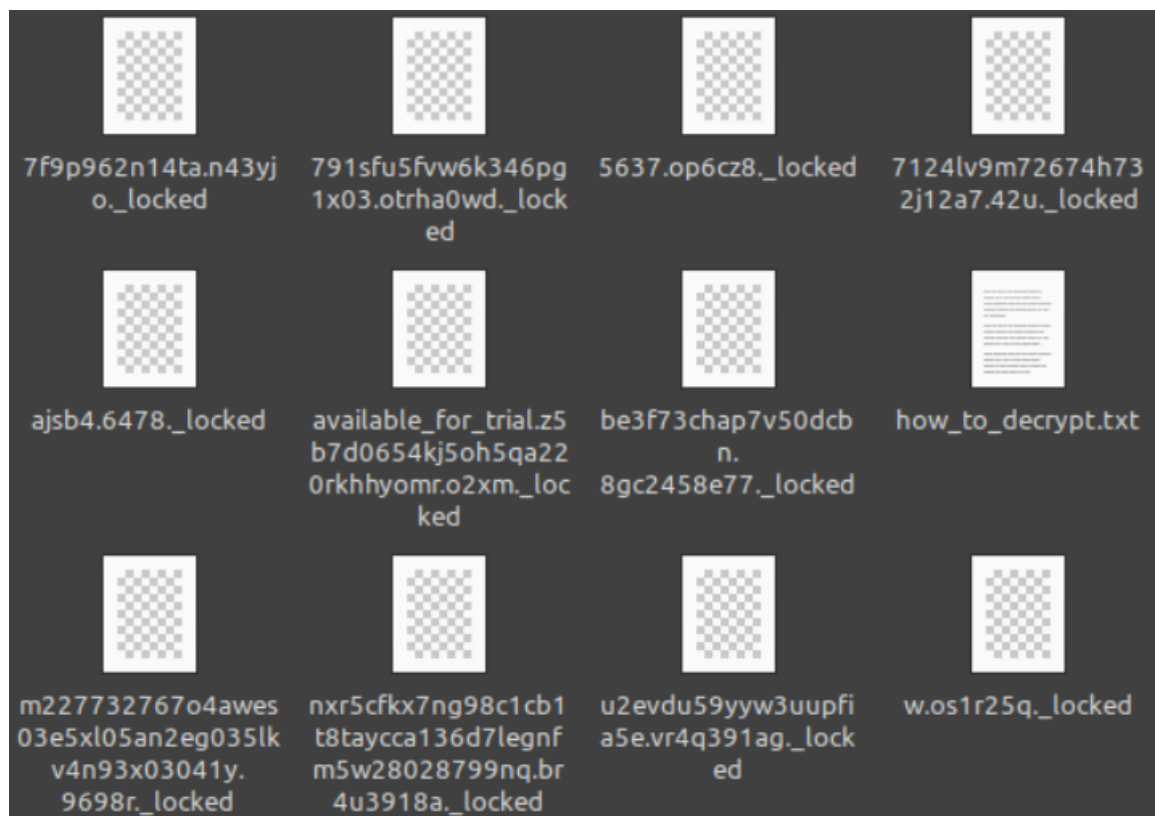


Figure 9. Files

encrypted by Trigona

To pressure victims into paying the ransom, the Trigona leak site contains a countdown timer and bidding options for parties interested in acquiring access to the leaked data. The attackers provide each victim with an authorization key that they can use to register on the negotiation portal provided by Trigona.

## Trigona leak site update

The Trigona ransomware group employs a double extortion scheme. In addition to the main leak site which displays the list of victim companies, Trigona's operators also use a Tor site where victims can communicate with the threat actor group to negotiate for the decryption tool. Interestingly, they also flag those victims that have already paid.

The report from Palo Alto revealed that an IP address hosting the leak site under the name "Trigona Leaks" and using port 8000. Additionally, another IP address titled "Leaks" was uncovered, which also employed port 8000 and shared the same IP range as the previously mentioned leak site-connected IP address.

During our investigation, we found another IP address on June 3 that was still active at the time of writing. This IP address, which uses port 3000 and the title *Blog*, is within the IP range of the previous addresses. We surmise that the threat actor relocates some of its infrastructure when their IP address is exposed. Using this third leak site, we were able to find their file storage site ([aeey7hxzgl6zowiwhiteo5xjbf6sb36tkbn5hptykgmbsjrbiygv4c4id\[.\]onion](http://aeey7hxzgl6zowiwhiteo5xjbf6sb36tkbn5hptykgmbsjrbiygv4c4id[.]onion)). This site hosts critical data stolen from victims such as documents, contracts, and other large amounts of data.

The Trigona ransomware group has poor operational security when it comes to the implementation of Tor sites — although their aim of targeting poorly-managed SQL servers is not something we usually see with less technically-proficient threat actors. Our [ransomware spotlight on TargetCompany](#) shows another group using a similar technique of targeting SQL servers.



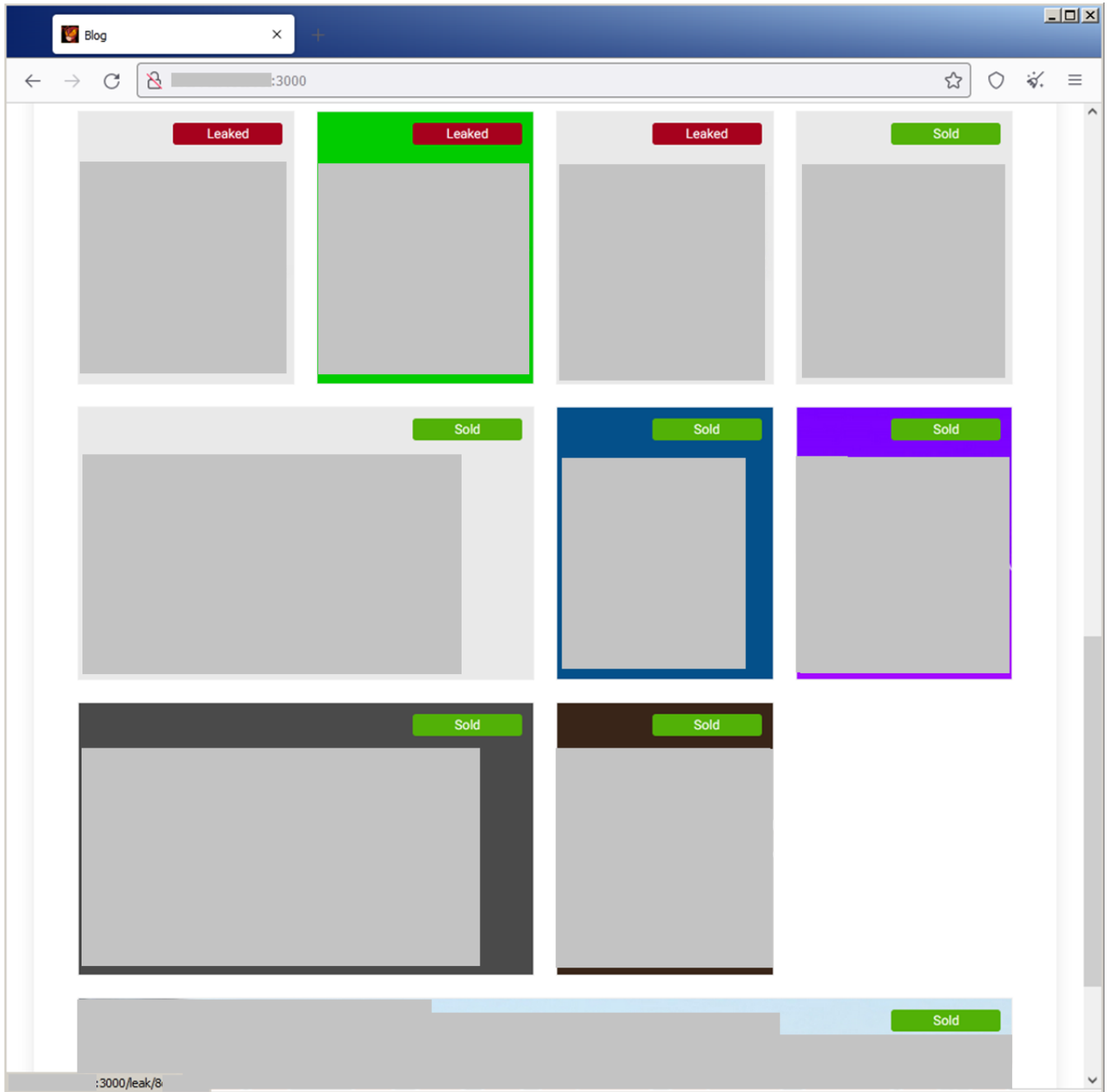


Figure 10. Main leak site of Trigona

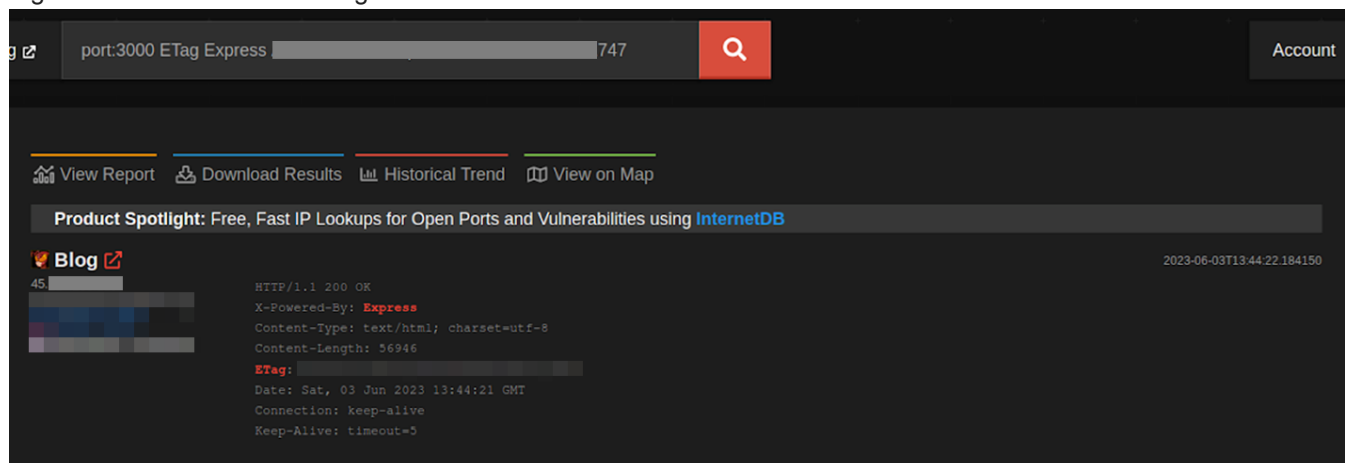


Figure 11. Trigona leak site found via Shodan on June 3, 2023

| File Name  | Size     | Timestamp           |
|------------|----------|---------------------|
| [Redacted] | 172.00kB | 2023-05-06 16:59:04 |
| [Redacted] | 4.24MB   | 2023-05-07 01:53:48 |
| [Redacted] | 107.43MB | 2023-05-07 01:24:56 |
| [Redacted] | 1.74GB   | 2023-05-07 01:08:24 |
| [Redacted] | 349.52MB | 2023-05-07 01:34:54 |
| [Redacted] | 865.74MB | 2023-05-07 01:24:28 |
| [Redacted] | 12.71GB  | 2023-05-06 23:34:24 |

Figure 12. The file storage Tor site of Trigona using the title “test”

## Conclusion and recommendations

The Trigona ransomware currently maintains a relatively low profile when compared to more widespread families, allowing it to operate covertly. Nonetheless, due to its continuous evolution and increased activity, we anticipate that Trigona will gain prominence in the near future. Furthermore, it joins the [growing list of ransomware groups that have developed a Linux version](#) to try and capitalize on the expanding high-value Linux market, adding evidence that Trigona’s operators are trying to expand their reach as much as possible. Therefore, it is crucial for individuals and organizations to familiarize themselves with this ransomware to prevent potential harm.

To safeguard systems against ransomware attacks, it is advisable for organizations to adopt effective measures. These include implementing data protection protocols and establishing backup and recovery procedures to ensure that data remains secure and can be restored in case of encryption or even deletion. Conducting routine vulnerability assessments and promptly patching systems can significantly reduce the impact of ransomware attacks that exploit vulnerabilities.

We recommend the following security precautions:

1. Enable multifactor authentication (MFA) to hinder attackers from moving laterally within a network and accessing sensitive information.
2. Follow the 3-2-1 rule when creating backups for important files. This involves generating three backup copies stored in two different file formats, with one copy stored in a separate location. This ensures redundancy and minimizes the risk of data loss.
3. Update and patch systems regularly. It is important to keep applications and operating systems up to date and establish robust patch management protocols to prevent malicious actors from exploiting software vulnerabilities.

## Indicators of Compromise

| SHA256   | Detection name               |
|--|------------------------------|
| f1e2a7f5fd6ee0c21928b1cae6e66724c4537052f8676feaa18e84cf3c0c663  | Ransom.Linux.TRIGONA.THCBBBC |
| 951fad30e91adae94ded90c60b80d29654918f90e76b05491b014b8810269f74 | Ransom.Linux.TRIGONA.THEAFBC |
| d0268d29e6d26d726adb848eff991754486880ebfd7affb3bb2a9e91a1dbb7c  | Ransom.Win64.TRIGONA.YXDFIZ  |
| a891d24823796a4ffa2fac76d92fec2c7fae1ac1c3665be0d4f85e13acd33f9  | Ransom.Win64.TRIGONA.THFOIBC |

---

|  |                             |
|--|-----------------------------|
| 2b40a804a6fc99f6643f8320d2668ebd2544f34833701300e34960b048485357 | Ransom.Win64.TRIGONA.YXDFOZ |
| 8cbe32f31befe7c4169f25614afd1778006e4bda6c6091531bc7b4ff4bf62376 | Ransom.Win32.TRIGONA.YPDDZ  |
| fb128dbd4e945574a2795c2089340467fcf61bb3232cc0886df98d86ff328d1b | Ransom.Win32.TRIGONA.YMDBJ  |
| 41c9080f9c90e00a431b2fb04b461584abe68576996379a97469a71be42fc6ff | Ransom.Win64.TRIGONA.YXDFUZ |
| c7a930f1ca5670978aa6d323d16c03a97d897c77f5cff68185c8393830a6083f | Trojan.MSIL.TRIGONA.YCDCT   |

---