New Ransomware Variant Uses Golang Packer | CrowdStrike

By Alexandru Ghita

Archived: 2026-04-05 13:53:20 UTC

CrowdStrike recently observed a ransomware sample borrowing implementations from previous HelloKitty and FiveHands variants and using a Golang packer compiled with the most recent version of Golang (Go1.16, released mid-February 2021). These ransomware families have been active since late 2019 and analyzed by the research community under different names based on various code overlaps. The similarities of this recent sample with previous HelloKitty and FiveHands variants involve similar ransomware functions written in C++, accepting CLI arguments, the use of four magic bytes appended to the encrypted files, and using an embedded public key.

Ransomware Sample Analysis

Similar to FiveHands ransomware, this variant uses a unique executable packer that requires a key value to decrypt the payload in memory using a command-line switch `“-key”` . This key is used to decrypt the embedded payload ransomware binary directly into memory. This method of using a memory-only dropper prevents security solutions from detecting the final payload without the unique key used to execute the packer. What’s new about this ransomware variant is the use of a Golang packer to encrypt the C++ written payload. Although Golang-written malware and packers are not new, compiling it with the latest Golang (Go1.16) makes it challenging to debug for malware researchers. That's because all necessary libraries are statically linked and included in the compiler binary and the function name recovery is difficult.

```
C:\VIR>eeb51dce12f243b332b51d7b1b11ecff155dd823ff8f9b79d6ad486cc49098ba.exe  -key                    --help
Registered ctrl handlers
Help usage:
          -path | select path; bin.exe -path "c:\temp"
         -limit | limit file offset; bin.exe -limit 10 (limit first 10 megabytes of file)
        -blocks | use random blocks encryption; bin.exe -blocks
       -maxrand | maximum random block range[0..maxrand); bin.exe -maxrand 10
         -nosum | no use checksum in enc/dec; bin.exe -nosum
     -norecycle | do not clear recycle; bin.exe -norecycle
     -noshadows | do not remove shadows; bin.exe -noshadows
           -key |
       -visible | do not hide console window
          -test | run self tests
         -tests | run self tests
      -tempfile | set temp file name
         --help | print this help
          -help | print this help
```

Figure 1. Golang Encryptor help menu (Click to enlarge)

The sample accepts different CLI arguments, suggesting it can limit the encryption to a specified path. After executing with the right key parameter (`Command execution bin.exe -key ""`), it starts decrypting the payload that is reflectively loaded into memory. The payload is the actual ransomware written in C++.

```
C:\                          >eeb51dce12f243b332b51d7b1b11ecff155dd823ff8f9b79d6ad486cc49098ba.exe -key              YM3S"
Registered ctrl handlers
pool created 240
pool created 244
Clear recycle
Clear shadows
ignore dir c:\\$recycle.bin, $recycle
ignore dir c:\\boot, boot
ignore file bootsect.bak, bootsect.bak
ignore file pagefile.sys, .sys
ignore file read_me_lock.txt, read_me_lock.txt
ignore dir c:\\program files, program files
ignore dir c:\\program files (x86), program files
ignore dir c:\\programdata, programdata
ignore file read_me_lock.txt, read_me_lock.txt
ignore file read_me_lock.txt, read_me_lock.txt
ignore dir c:\\recovery\windowsre, windows
ignore file swapfile.sys, .sys
C:\\System Volume Information\read_me_lock.txt - error drop readme EPERM
ignore file cglpt64.sys, .sys
```

Figure 2. Executable packer and key on the command-line (Click to enlarge)

The ransomware has the capability to also clear RecycleBin and to delete each Shadow Copy by ID ("Win32_ShadowCopy.ID") using WMI functions (Figure 3) similar to the other ransomware variants like FiveHands and HelloKitty. We have also identified an implementation of "IoCompletionPorts" for a better threading model in the encryption process, similar to FiveHands.

```
if ( CoCreateInstance(&stru_457BEC, 0, 0x4401u, &stru_457B1C, &v13) >= 0 && v13 )
{
  v4 = SysAllocString;
  v9 = SysAllocString(L"ROOT\\cimv2");
  if ( (*(int (__stdcall **)(LPVOID, OLECHAR *, _DWORD, _DWORD, _DWORD, _DWORD, _DWORD, LPVOID, IUnknown **))(*(_DWORD *)v13 + 12))(
         v13,
         v9,
         0,
         0,
         0,
         0,
         0,
         ppv,
         &pProxy) >= 0
    && pProxy )
  {
    if ( CoSetProxyBlanket(pProxy, 0xAu, 0, 0, 3u, 3u, 0, 0) >= 0 )
    {
      v12 = 0;
      bstrString = SysAllocString(L"WQL");
      v8 = SysAllocString(L"select * from Win32_ShadowCopy");
      if ( ((int (__stdcall *)(IUnknown *, OLECHAR *, OLECHAR *, int, _DWORD, int *))pProxy->lpVtbl[6].Release)(
             pProxy,
             bstrString,
             v8,
             48,
             0,
             &v12) >= 0 )
      {
        v16 = 0;
        (*(void (__stdcall **)(int, int, int, int *, int *))(*(_DWORD *)v12 + 16))(v12, -1, 1, &v15, &v16);
        while ( v16 )
        {
          if ( (*(int (__stdcall **)(int, const wchar_t *, _DWORD, VARIANTARG *, _DWORD, _DWORD))(*(_DWORD *)v15 + 16))(
                 v15,
                 L"id",
                 0,
                 &pvarg,
                 0,
                 0) >= 0
            && pvarg.vt == 8 )
          {
            wsprintfW(psz, L"Win32_ShadowCopy.ID='%s'", pvarg.lVal);
            v5 = v4(psz);
            v6 = v5;
```

Figure 3. WMI functions for deleting shadow copies (Click to enlarge)

A RSA public key is hard-coded in the code block. This is used to encrypt each symmetric key per file and append it at the end of the encrypted file along with the four bytes `D0 BA AD DE`. The last four bytes are used to check if the targeted file was previously encrypted. In the symmetric key generation process, we also found code implementation for the use of the Salsa20 algorithm. At the end, the final files are renamed using the `. locked` extension. A ransom note is placed into each folder and directory, including the root path, after the files are

encrypted. The ransom note file is named `read_me_lock.txt` and provides instructions to the victim on how to recover their encrypted files.

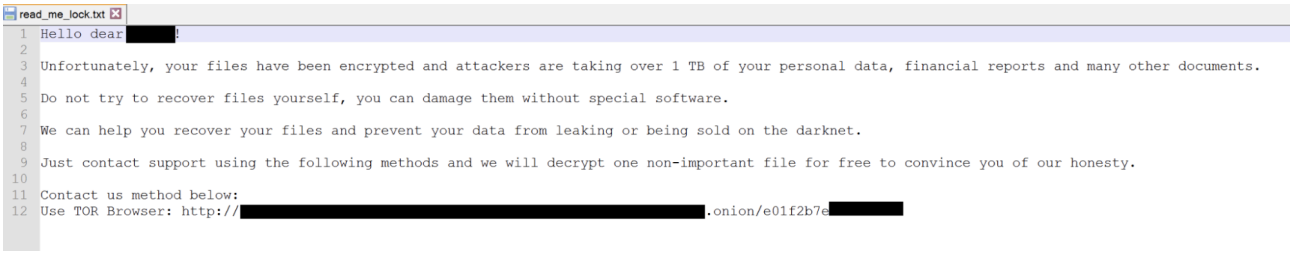

Figure 4. Ransom note (Click to enlarge)

The message does not offer any bitcoin wallet in which payment should be made. Instead, it offers a TOR link where victims can contact the ransom operators. The message also claims to have extracted over 1 TB of personal and sensitive data from the victim, potentially threatening extortion. Visiting the TOR address, victims will be directed to a temporary chat session where they are encouraged to engage with the ransomware operators to negotiate decryption fees.

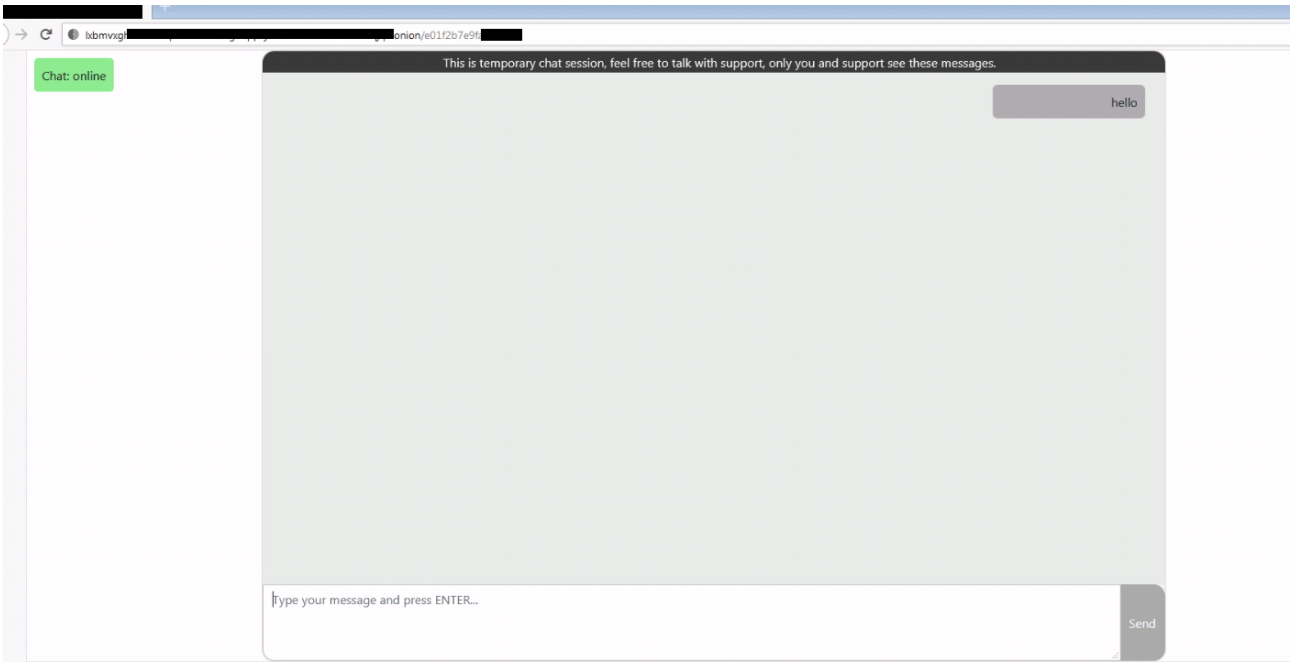

Figure 5. TOR chat box for communicating with operators (Click to enlarge)

CrowdStrike Falcon® Protection

The CrowdStrike Falcon® sensor has the ability to detect the execution of the Golang packer using machine learning (ML), identifying it during the very early stage of execution before it can deliver the ransomware payload. Falcon's ML algorithm can protect customers by providing coverage against this analyzed threat, as illustrated below.

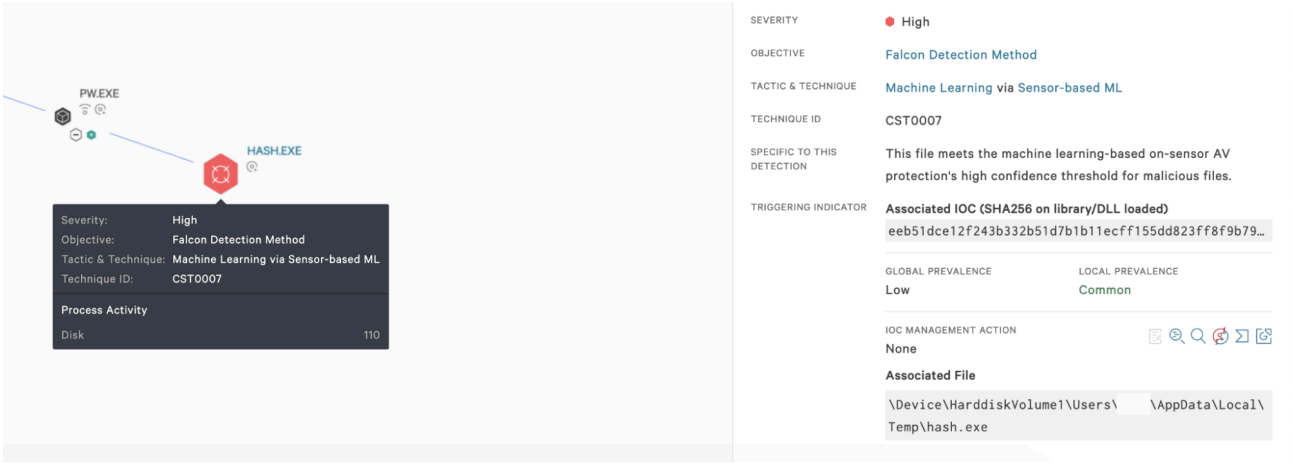

Figure 6. Packer process blocked by ML algorithm (Click to enlarge)

Indicators of Compromise (IOCs)

File	SHA256
Ransomware Payload	b24dcfdda948b339637fe507cf032ec233288691b700e1585cb34b4190704858
Golang Packer	eeb51dce12f243b332b51d7b1b11ecff155dd823ff8f9b79d6ad486cc49098ba
Hardcoded Asymmetric public key extracted from strings	-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAye/wtpovm5pDdZvFRrpj uob4f2bMN9/Ws3TR4MwR0Pngsvpf2b1iO0wjCZNx9wrut+7s5myMPFpE50Qw6Q7o oIFNIguxDyyC3saLlUvty+eohxY0JBv1ljMads9PzjtHvYjlaiB9/HCDNQhucGt3 SlCKQ0bW3Sx2t4yEXHM2T0Qz+pEM2XG2Lkm7HATW34JHyKkJcdm850vxKvDX/QIN C9obv4bvpUgBZq836aT9Uu5B7LBZuMeUNJPq5WYwQOgPhitjCpXZTP1OJrT6Fh6V 0+pnupgv/NqzFCbSkqa96fXM0Lo+EMzI4sWfPhTlZ+qKynr/nw0VCw7G+T1wRC7M 0wIDAQAB -----END PUBLIC KEY-----

Additional Resources

- *Learn more about ransomware adversaries in the CrowdStrike Adversary Universe.*
- *Download the CrowdStrike 2021 Global Threat Report for more information about adversaries tracked by CrowdStrike Intelligence in 2020.*
- *See how the powerful, cloud-native CrowdStrike Falcon® platform protects customers from the latest variants of ransomware in these blogs: DarkSide Goes Dark: How CrowdStrike Falcon® Customers Were Protected and Under Attack: Protecting Against Conti, DarkSide, REvil and Other Ransomware.*
- *Get a full-featured free trial of CrowdStrike Falcon® Prevent™ and learn how true next-gen AV performs against today's most sophisticated threats.*
