

How to Decrypt the PartyTicket Ransomware Targeting Ukraine | CrowdStrike

By CrowdStrike Intelligence Team

Archived: 2026-04-06 00:52:31 UTC

Summary

On Feb. 23, 2022, destructive attacks were conducted against Ukrainian entities. Industry reporting has claimed the Go-based ransomware dubbed *PartyTicket* (or *HermeticRansom*) was identified at several organizations affected by the attack,¹ among other families including a sophisticated wiper CrowdStrike Intelligence tracks as *DriveSlayer* (*HermeticWiper*).

Analysis of the *PartyTicket* ransomware indicates it superficially encrypts files and does not properly initialize the encryption key, making the encrypted file with the associated `.encryptedJB` extension recoverable.

Technical Analysis

A *PartyTicket* ransomware sample has a SHA256 hash of

`4dc13bb83a16d4ff9865a51b3e4d24112327c526c1392e14d56f20d6f4eaf382`. It has been observed associated with the file names `cdir.exe`, `cname.exe`, `connh.exe` and `intpub.exe`.

The ransomware sample — written using Go version 1.10.1 — contains many symbols that reference the U.S. political system, including `voteFor403`, `C:/projects/403forBiden/WhiteHouseE` and `primaryElectionProcess`.

The ransomware iterates over all drive letters and recursively enumerates the files in each drive and its subfolders, excluding file paths that contain the strings `Windows` and `Program Files` and the folder path `C:\Documents and Settings` (the latter folder was replaced in Windows versions later than Windows XP with `C:\Users`). Files with the following extensions are selected for encryption:

`acl, avi, bat, bmp, cab, cfg, chm, cmd, com, contact, crt, css, dat, dip, dll, doc, docx, dot, encryptedjb, epub, exe, gif, htm, html, ico, in, iso, jpeg, jpg, mp3, msi, odt, one, ova, pdf, pgsql, png, ppt, pptx, pub, rar, rtf, sfx, sql, txt, url, vdi, vsd, wma, wmv, wtv, xls, xlsx, xml, xps, zip`

For each file path that passes the previously described path and extension checks, the ransomware copies an instance of itself to the same directory it was executed from and executes via the command line, passing the file path as an argument. The parent ransomware process names its clones with a random UUID generated by a public library² that uses the current timestamp and MAC addresses of the infected host's network adapters. The malware developer attempted to use Go's `WaitGroup` types to implement concurrency; however, due to a likely coding error, the ransomware creates a very large number of threads (one per enumerated file path) and copies its own

binary into the current directory as many times as there are selected files. After all encryption threads have ended, the original binary deletes itself via the command line.

When the sample receives a file path as an argument, it encrypts the file using AES in Galois/Counter Mode (GCM). The AES key is generated using the Go `rand` package's `Intn` function to select offsets in the character array `1234567890ABCDEFGHIJKLMN0PQRSTUVWXYZ`, generating a 32-byte key. Due to another likely coding error, the seed for the `Intn` function is updated after the key is generated, meaning the same AES key is generated each time the binary and its clones are run. All of the files encrypted on a host are encrypted with the same key, and knowledge of the corresponding *PartyTicket* sample's key enables their decryption. A script using this flaw to recover the encrypted files is available on the [CrowdStrike Git Repository](#).

For each file, the AES encryption key is itself encrypted with RSA-OAEP, using a public RSA key that has the following parameters:

```
Modulus (N): 0xcbb94cb189a638b51e7cfe161cd92edb7145ecbd93989e78c94f8c15c61829286fd834d80c931daed4ac4aba14835fd:  
Exponent (E): 0x10001
```

Before encryption, the ransomware renames the file using the format `<original file name>.com>.encryptedJB` ("JB" very likely stands for the initials of the United States president Joseph Biden, given the other political content in the binary). The ransomware then overwrites the content with the encrypted data. *PartyTicket* will only encrypt the first 9437184 bytes (9.44 MB) of a file. If the file passed as an argument is larger than this limit, any data above it is left unencrypted. After the file contents are encrypted, *PartyTicket* appends the RSA-encrypted AES key at the end of the file.

The ransomware also writes an HTML ransom note on the user's desktop directory with the name `read_me.html` before the file encryption starts (Figure 1). Unless they are intentional mistakes, grammar constructs within the note suggest it was likely not written or proofread by a fluent English speaker.

"The only thing that we learn from new elections is we learned nothing from the old!"

Thank you for your vote! All your files, documents, photoes, videos, databases etc. have been successfully encrypted!

Now your computer has a special ID: **XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX**

Do not try to decrypt then by yourself - it's impossible!

It's just a business and we care only about getting benefits. The only way to get your files back is to contact us and get further instuctions.

To prove that we have a decryptor send us any encrypted file (less than 650 kbytes) and we'll send you it back being decrypted. This is our guarantee.

NOTE: *Do not send file with sensitive content. In the email write us your computer's special ID (mentioned above).*

So if you want to get your files back contact us:

1) `vote2024forjb@protonmail[.]com`

2) `stephanie.jones2024@protonmail[.]com` - if we dont't answer you during 3 days

Have a nice day!

Figure

1. Ransom note

Assessment

CrowdStrike Intelligence does not attribute the *PartyTicket* activity to a named adversary at the time of writing.

The ransomware contains implementation errors, making its encryption breakable and slow. This flaw suggests that the malware author was either inexperienced writing in Go or invested limited efforts in testing the malware, possibly because the available development time was limited. In particular, *PartyTicket* is not as advanced as *DriveSlayer*, which implements low-level NTFS parsing logic. The relative immaturity and political messaging of the ransomware, the deployment timing and the targeting of Ukrainian entities are consistent with its use as an additional payload alongside *DriveSlayer* activity, rather than as a legitimate ransomware extortion attempt.

YARA Signatures

The following YARA rule can be used to detect *PartyTicket*:

```
rule CrowdStrike_PartyTicket_01 : ransomware golang
{
  meta:
    copyright = "(c) 2022 CrowdStrike Inc."
    description = "Detects Golang-based crypter"
    version = "202202250130"
    last_modified = "2022-02-25"
  strings:
    $ = ".encryptedJB" ascii
    $start = { ff 20 47 6f 20 62 75 69 6c 64 20 49 44 3a 20 22 }
```

```
    $end = { 0a 20 ff }
condition:
    uint16(0) == 0x5A4D and uint32(uint32(0x3C)) == 0x00004550 and
    for 1 of ($end) : ( @start < @ and @start + 1024 > @) and
    all of them
}

rule CrowdStrike_PartyTicket_02 : PartyTicket golang
{
    meta:
        copyright = "(c) 2022 CrowdStrike Inc."
        description = "Detects Golang-based PartyTicket ransomware"
        version = "202202250130"
        last_modified = "2022-02-25"
    strings:
        $s1 = "voteFor403"
        $s2 = "highWay60"
        $s3 = "randomiseDuration"
        $s4 = "subscribeNewPartyMember"
        $s5 = "primaryElectionProces"
        $s6 = "baggageGatherings"
        $s7 = "getBoo"
        $s8 = "selfElect"
        $s9 = "wHiteHousE"
        $s10 = "encryptedJB"
        $goid = { ff 20 47 6f 20 62 75 69 6c 64 20 49 44 3a 20 22 71 62 30 48 37 41 64 57 41 59 44 7a 66 4d 41 }
        $pdb = "C://projects//403forBiden//wHiteHousE"
    condition:
        (uint32(0) == 0x464c457f or (uint16(0) == 0x5a4d and uint16(uint32(0x3c)) == 0x4550)) and 4 of ($s*) or
}
}
```

Script to Decrypt *PartyTicket* Encrypted Files

Due to the previously discussed implementation errors in the AES key generation, it is possible to recover the AES key used for encryption by *PartyTicket*. The below Go script decrypts files encrypted by *PartyTicket* sample `4dc13bb83a16d4ff9865a51b3e4d24112327c526c1392e14d56f20d6f4eaf382`. The script takes the file to be decrypted as an argument via the “-p” flag and saves the decrypted output to “decrypted.bin” in the same directory. The script can be built as an executable or run via the Go run package; it was tested using Go version go1.16.6.

```
package main

import (
    "crypto/aes"
    "crypto/cipher"
    "encoding/hex"
    "fmt"
}
```

```
"os"
"flag"
)

func main() {

    encrypted_filepath := flag.String("p", "encrypted.bin", "Path to encrypted file")
    flag.Parse()

    fmt.Printf("Decrypting file : %s\n", *encrypted_filepath)
    key_bytes := <>byte("6FBB7P950E8UT5QRTTEBIWAR88S74D0")
    key := hex.EncodeToString(key_bytes)
    fmt.Printf("Decryption key : %s\n", key_bytes)

    dat, err := os.ReadFile(*encrypted_filepath)
    if err != nil {
        fmt.Println("Unable to open file, please supply path of encrypted file with flag -p, default f
        os.Exit(3)
    }

    decrypted_filepath := "decrypted.bin"
    filecontents := dat
    encrypted_contents := filecontents<:len(filecontents) - 288>
    enc_size := len(encrypted_contents)
    bsize := 1048604
    cycles := enc_size / bsize

    if cycles == 0{

        encrypted := hex.EncodeToString(encrypted_contents)
        decrypted := decrypt(encrypted, key)
        write_output(decrypted_filepath, decrypted)
    } else {
        for i:=0; i<cycles; i++ { if i >= 9 {
            start := 9 * bsize
            end := enc_size
            data := string(encrypted_contents)
            write_output(decrypted_filepath, data)
            break
        }
        block_start := i * bsize
        block_end := (i+1) * bsize
        if block_end > enc_size{
            block_end := enc_size
            encrypted:=hex.EncodeToString(encrypted_contents)
            decrypted := decrypt(encrypted, key)
            write_output(decrypted_filepath, decrypted)
        }
    }
}
```

```
        }

        encrypted:=hex.EncodeToString(encrypted_contents)
        decrypted := decrypt(encrypted, key)
        write_output(decrypted_filepath, decrypted)
    }
}

fmt.Printf("Decrypted file written to : %s\n", decrypted_filepath)

}

func write_output(filepath string, data string) {
    f, err := os.OpenFile(filepath, os.O_APPEND|os.O_CREATE|os.O_WRONLY, 0644)
    if err != nil {
        panic(err)
    }
    byte_data := <>byte(data)
    f.Write(byte_data)
    f.Close()
}

func decrypt(encryptedString string, keyString string) (decryptedString string) {

    key, _ := hex.DecodeString(keyString)
    enc, _ := hex.DecodeString(encryptedString)

    block, err := aes.NewCipher(key)
    if err != nil {
        panic(err.Error())
    }
    aesGCM, err := cipher.NewGCM(block)
    if err != nil {
        panic(err.Error())
    }
    nonceSize := aesGCM.NonceSize()
    nonce, ciphertext := enc<:nonceSize>, enc
    plaintext, err := aesGCM.Open(nil, nonce, ciphertext, nil)
    if err != nil {
        panic(err.Error())
    }

    return fmt.Sprintf("%s", plaintext)
}
```

Endnotes

1. <https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/ukraine-wiper-malware-russia>
2. <https://pkg.go.dev/github.com/satori/go.uuid#NewV1>

Additional Resources

- Read more about use of offensive cyber operations against Ukraine: [Lessons Learned From Successful Use of Offensive Cyber Operations Against Ukraine and What May Be Next](#).
- Learn how CrowdStrike Falcon® provides continuous protection against DriveSlayer and wiper-style threats: [CrowdStrike Falcon® Protects from New Wiper Malware Used in Ukraine Cyberattacks](#)
- Read about WhisperGate in this CrowdStrike Intelligence blog: [Technical Analysis of the WhisperGate Malicious Bootloader](#).
- Learn about the powerful, cloud-native [CrowdStrike Falcon® platform by visiting the product webpage](#).
- [Get a full-featured free trial of CrowdStrike Falcon® Prevent](#) to see for yourself how true next-gen AV performs against today's most sophisticated threats.

Source: <https://www.crowdstrike.com/blog/how-to-decrypt-the-partyticket-ransomware-targeting-ukraine/>