# CrowdStrike Discovers New DoppelPaymer Ransomware & Dridex Variant

By bsg.sf.bh



CrowdStrike® Intelligence has identified a new ransomware variant identifying itself as *BitPaymer*. This new variant was behind a series of ransomware campaigns beginning in June 2019, including attacks against the City of Edcouch, Texas and the Chilean Ministry of Agriculture. We have dubbed this new ransomware *DoppelPaymer* because it shares most of its code with the BitPaymer ransomware operated by INDRIK SPIDER. However, there are a number of differences between DoppelPaymer and BitPaymer, which may signify that one or more members of INDRIK SPIDER have split from the group and forked the source code of both *Dridex* and BitPaymer to start their own Big Game Hunting ransomware operation.

## INDRIK SPIDER Origins

INDRIK SPIDER was formed in 2014 by former affiliates of the GameOver Zeus criminal network who internally referred to themselves as “The Business Club.” Shortly after the group’s inception, INDRIK SPIDER developed their own custom malware known as Dridex. Early versions of Dridex were primitive, but over the years the malware became increasingly professional and sophisticated. In fact, Dridex operations were significant throughout 2015 and 2016, making it one of the most prevalent eCrime malware families. At this time, INDRIK SPIDER was primarily conducting wire fraud, resulting in the loss of millions of dollars globally. Over time, INDRIK SPIDER encountered a number of obstacles to their wire fraud operations. First, in 2015 the group had to overcome a takedown operation, which resulted in the arrest of one of its affiliates, who used the alias “Smilex.” This setback was followed by a law enforcement operation in the U.K. designed to break up the money laundering network supporting INDRIK SPIDER's monetization of Dridex campaigns. The dismantling of this network also coincided with the arrest, and subsequent imprisonment, of a U.K. bank employee who helped set up fake accounts. Perhaps as a result of these obstacles, INDRIK SPIDER changed their methods of operation in 2017, conducting smaller Dridex distribution campaigns. In August 2017, the group introduced BitPaymer ransomware and began to focus on leveraging access within a victim organization to demand a high ransom payment.

## BitPaymer Origins

CrowdStrike Intelligence, has tracked the original BitPaymer since it was first identified in August 2017. In its first iteration, the BitPaymer ransom note included the ransom demand and a URL for a TOR-based payment portal. The payment portal included the title “Bit paymer” along with a reference ID, a Bitcoin (BTC) wallet, and a contact email address. An example of this portal is shown in Figure 1. Within the first month of operation, the ransom amount was dropped from the ransom note. In July 2018, the payment portal URL was also removed. From July 2018 until present, the ransom note has only included two contact emails, which are used to negotiate the ransom.


Figure 1. Original BitPaymer Payment Portal via a TOR Hidden Service

## Latest BitPaymer Version

In November 2018, there was a significant update to BitPaymer. The ransom note was updated to include the victim's name, and the file extension appended to encrypted files was also customized to use a representation of the victim's name. An example of the new ransom note is shown below in Figure 2.


Figure 2. Latest BitPaymer Ransom Note

In addition to the updated ransom note and encrypted file extension, BitPaymer's file encryption routine was updated to use 256-bit AES in cipher block chaining (CBC) mode with a randomly generated key and a NULL initialization vector. Previous versions of BitPaymer had used 128-bit RC4. Since AES is a block cipher, the implementation requires padding in cases where the data is not a multiple of the block size. Typically, this is implemented by adding zeros or the number *n* of padding bytes *n* times (also known as PKCS#7). However, INDRIK SPIDER chose to generate *n* bytes randomly for padding. As a result, the malware developer had to preserve the random padding bytes in order to correctly decrypt the last data block of an encrypted file. This is reflected in the BitPaymer ransom note with a new field of `TAIL`, as shown above in Figure 2, which contains the Base64-encoded `TAIL` padding and encrypted AES `KEY.` Interestingly, the BitPaymer developers implemented an encryption initialization function in the ransomware code that selects one of three desired encryption algorithms. The algorithm is chosen by an argument that is passed as an integer parameter to the function. The current values supported are 1, 2, and 3 for 128-bit RC4, 128-bit AES and 256-bit AES, respectively. Newer versions of BitPaymer pass the hard-coded value of 3 for 256-bit AES encryption into the function, as shown in Figure 3.

```
if ( CryptoAlgorithm == 1 )
{
  crypto->CryptoAlgorithm = CALG_RC4;
}
else
{
  if ( CryptoAlgorithm != 2 )
  {
    if ( CryptoAlgorithm == 3 )
    {
      crypto->CryptoAlgorithm = CALG_AES_256;
      crypto->KeySize = 32;
    }
    else
    {
      crypto->CryptoAlgorithm = 0;
      crypto->KeySize = 0;
    }
    goto CRYPTO;
  }
  crypto->CryptoAlgorithm = CALG_AES_128;
}
crypto->KeySize = 16;
```

Figure 3. Latest BitPaymer Encryption Selection Pseudocode

Along with the updated file encryption routine, the size of the victim-specific RSA public key has also been increased from 1,024-bit to 4,096-bit. This asymmetric key is used to encrypt the generated symmetric file encryption keys. If the ransom is paid, INDRIK SPIDER will provide a decryption tool that contains the corresponding victim's RSA private key. It is unclear why INDRIK SPIDER moved from RC4 to AES encryption, but it may be due to concerns about the relative weakness of

RC4 in comparison to AES. The increase in the RSA key size also greatly augments the cryptographic strength protecting the file encryption keys. However, there is no evidence that BitPaymer's prior or current encryption has been broken. Since the update in November 2018, INDRIK SPIDER has actively used the latest version of BitPaymer in at least 15 confirmed ransomware attacks. These attacks have continued throughout 2019, with multiple incidents occurring in June and July of 2019 alone. While the first known victims of DoppelPaymer were targeted in June 2019, we were able to recover earlier builds of the malware dating back to April 2019. These earlier builds are missing many of the new features found in later variants, so it is not clear if they were deployed to victims or if they were simply built for testing. To date, we have identified eight distinct malware builds and three confirmed victims with ransom amounts of 2 BTC, 40 BTC and 100 BTC. Based on the USD to BTC exchange rate at the time of this writing, these ransom amounts vary from approximately $25,000 to over $1,200,000. The ransom note used by DoppelPaymer is similar to those used by the original BitPaymer in 2018. The note does not include the ransom amount; however, it does contain a URL for a TOR-based payment portal, and instead of using the keyword `KEY` to identify the encrypted key, the note uses the keyword `DATA` as shown in Figure 4.

```
Your network has been penetrated.

All files on each host in the network have been encrypted with a strong algorythm.

Backups were either encrypted or deleted or backup disks were formatted.
Shadow copies also removed, so F8 or any other methods may damage encrypted data but not recover.

We exclusively have decryption software for your situation
No decryption software is available in the public.

        DO NOT RESET OR SHUTDOWN - files may be damaged.
        DO NOT RENAME OR MOVE the encrypted and readme files.
        DO NOT DELETE readme files.
        DO NOT use any recovery software with restoring files overwriting encrypted.
        This may lead to the impossibility of recovery of the certain files.


To get info (decrypt your files) contact us at your personal page:

        1. Download and install Tor Browser: https://www.torproject.org/download/
        2. After a successful installation, run the browser and wait for initialization.
        3. Type in the address bar:

                        REDACTED

        4. Follow the instructions on the site
        5. You should get in contact in 48 HOURS since your systems been infected.
        6. The link above is valid for 7 days.
           After that period if you not get in contact
           your local data would be lost completely.


The faster you get in contact - the lower price you can expect.

DATA
AQAAAD0BAgAAEGYAAACkAAAVZbpNets6EP1bQXd7Gb8IcODGmeKDm5FmsMe1p/RYzI0ljRcE2tH4
jZ2CksvKFzlBu1Rwa7P5l6dvX5VhxEHyj0TeLTwSFpIsBbJyRHNb1/G6biex/0RKKmkCkJ9gqIvi
vy8o9Ulz2c6jdeqr+ViaYpYYODwOwCa2AJso1FYqJ4B9ek7TCOBdjNKMSAyBz+M5gQrlNeOmYgGs
itXGyCwiwTN3rGDdXFINkSTRw1mM3bg6D8gxOHUnfbjIi1VA3ikHO3ORs/9kQ0CliOfF32owhwLQ
iE66ds59Dq/aSby/3RKuFrPSatuwf6TqLhXTKn6CnCqT1fNJY0d1zZiMxJSV
```

Figure 4. DoppelPaymer Ransom Note

The payment portal for DoppelPaymer is almost identical to the original BitPaymer portal. The "Bit paymer" title is still present on the web page and a unique ID is still used to identify the victim. The portal provides a ransom amount, a countdown timer and a BTC address where the ransom payment can be sent. An example of the DoppelPaymer ransom portal web page is shown below in Figure 5.

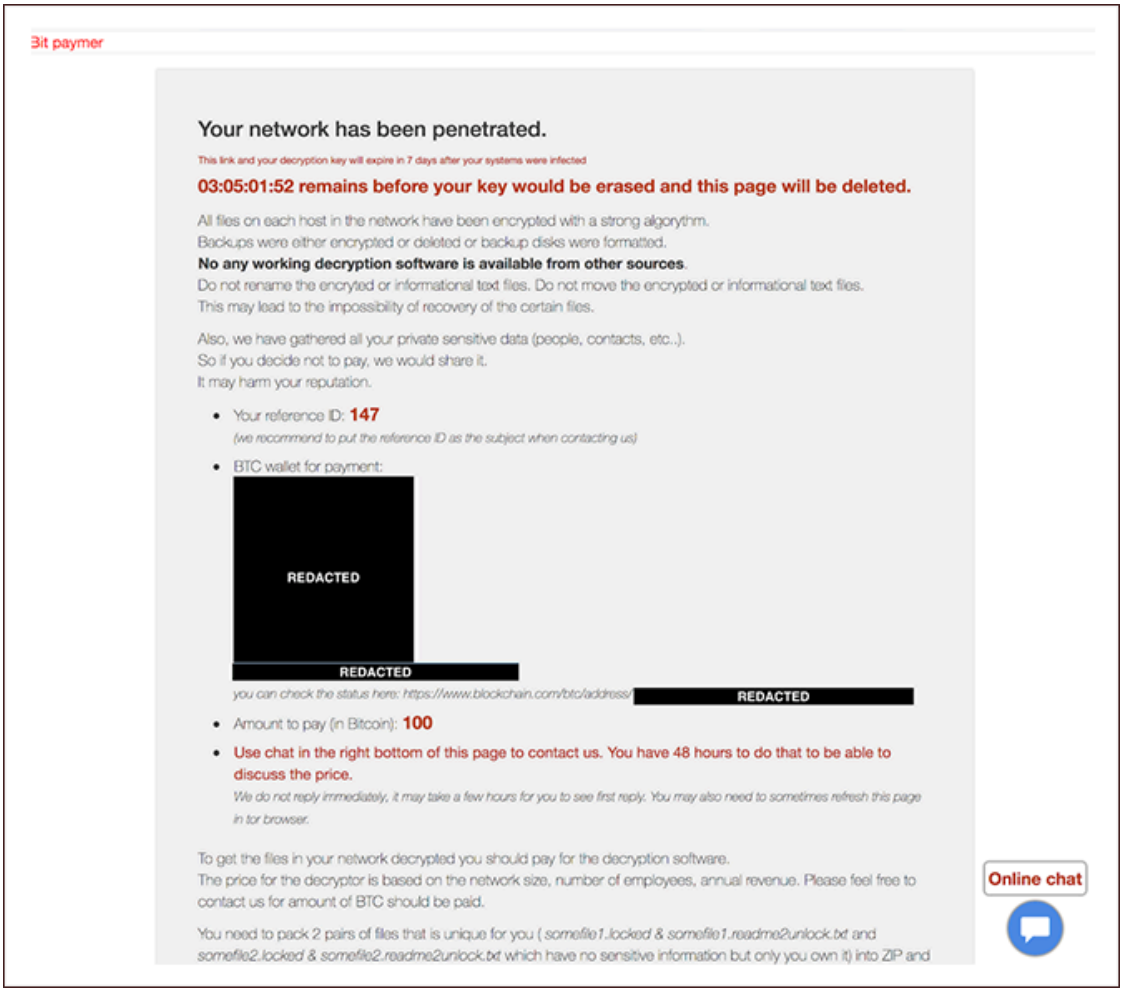


Figure 5. DoppelPaymer Ransomware Payment Portal

## DoppelPaymer and BitPaymer Encryption Comparison

Although DoppelPaymer and BitPaymer share significant amounts of code, there are some notable encryption differences, which are described in Table 1.

| | **DoppelPaymer** | **BitPaymer** |
|---|---|---|
| **Ransom note** | Each readme file contains an encrypted 256-bit AES key in a field named `DATA.` | Each readme file contains an encrypted 256-bit AES key in a field named `KEY.` Older versions contained an encrypted 128-bit RC4 key in the `KEY` field. Current versions use anonymous email services such as ProtonMail for ransom payment negotiations. |
| **Encryption** | 2048-bit RSA + 256-bit AES | 4096-bit RSA + 256-bit AES. Older versions used 1024-bit RSA + 128-bit RC4. |
| **Encryption (AES) padding scheme** | Standard padding (PKCS#7) | Random bytes specified in a field named `TAIL` |
| **Ransom filename** | Encrypted files are renamed with a `.locked` extension. | Encrypted files are renamed with the victim name as the extension. Older versions are appended the suffix `.locked` to the names of encrypted files. |

**Table 1. Encryption-Related Differences Between DoppelPaymer and BitPaymer** There are obvious similarities between the tactics, techniques and procedures (TTPs) used by DoppelPaymer and prior TTPs of BitPaymer, such as the use

of TOR for ransom payment and the `.locked` extension. However, the code overlaps suggest that DoppelPaymer is a more recent fork of the latest version of BitPaymer. For example, in the latest version of BitPaymer, the code for RC4 string obfuscation reverses the bytes prior to encryption, and includes a helper function that provides support for multiple forms of symmetric encryption (i.e., RC4, 128-bit AES, and 256-bit AES), as shown in Figure 3.

## New DoppelPaymer Features and the Use of ProcessHacker

In addition to the changes discussed above, numerous modifications were made to the BitPaymer source code to improve and enhance DoppelPaymer's functionality. For instance, file encryption is now threaded, which can increase the rate at which files are encrypted. The network enumeration code was updated to parse the victim system's Address Resolution Protocol (ARP) table, retrieved with the command `arp.exe -a .` The resulting IP addresses of other hosts on the local network are combined with domain resolution results via `nslookup.exe`. (In a similar approach, previous versions of BitPaymer made use of the command `net.exe view` to enumerate network shares.) In addition, DoppelPaymer is designed to run only after a specific command line argument is provided. The malware computes a CRC32 checksum of the first argument passed on the command line and adds it with a constant value that is hard-coded in the binary. The malware then adds the instruction pointer address to this result, which becomes the destination for a jmp used to continue the malware execution. The hard-coded constant value is unique to each build. In the sampled analyzed, this value was `0x672e6eb7`, as shown below in Figure 6.

```
.text:01132254    sub     esp, 48h
.text:01132257    lea     eax, [esp+48h+String1]
.text:0113225B    push    offset String2  ; "Setup run\n"
.text:01132260    push    eax             ; lpString1
.text:01132261    call    ds:lstrcpyW
.text:01132267    call    ds:GetCommandLineW
.text:0113226D    mov     edx, eax
.text:0113226F    lea     eax, [esp+48h+pNumArgs]
.text:01132273    push    eax             ; pNumArgs
.text:01132274    push    edx             ; lpCmdLine
.text:01132275    call    ds:CommandLineToArgvW
.text:0113227B    mov     ebx, eax
.text:0113227D    mov     edx, 7FFFFFFFh
.text:01132282    mov     ecx, [ebx+4]
.text:01132285    call    malware_StrLen
.text:0113228A    add     eax, eax
.text:0113228C    push    eax             ; Length
.text:0113228D    push    dword ptr [ebx+4] ; Buffer
.text:01132290    push    0               ; InitialCrc
.text:01132292    call    ds:RtlComputeCrc32
.text:01132298    add     eax, 672E6EB7h
.text:0113229D    mov     [esp+48h+var_48], eax
.text:011322A0    call    sub_11445C0
.text:011322A5    add     [esp+48h+var_48], eax
.text:011322A8    jmp     [esp+48h+var_48]
```

Figure 6. DoppelPaymer Control Flow Obfuscation

If no arguments are provided, or if an incorrect value is provided on the command line, DoppelPaymer will crash. This design was likely intended to hinder automated malware analysis environments. Perhaps the most interesting change that the DoppelPaymer author made is to terminate processes and services that may interfere with file encryption. DoppelPaymer contains several lists of CRC32 checksums of process and service names that are blacklisted. The malware author included CRC32 checksums rather than strings to hinder reverse engineering efforts. However, it is possible to brute-force all of the checksums and recover the respective strings, as shown in Tables 7-11 found in the Appendix.

### ProcessHacker

In order to terminate some of these processes and services, DopplePaymer uses an interesting technique that leverages ProcessHacker, a legitimate open-source administrative utility. This application is bundled with a kernel driver that can be used to terminate processes and services. DoppelPaymer is bundled with six portable executable (PE) files that are encrypted and compressed in the malware's `sdata` section. These PE files contain 32-bit and 64-bit versions of the following:

- ProcessHacker application
- ProcessHacker kernel driver
- A custom stager DLL that is used to exploit ProcessHacker

The modules are extracted by using the first 16 bytes of the `sdata` section as an RC4 key to decrypt the next 4 bytes of data, which is the size (big endian) of the subsequent encrypted data. The encrypted data that follows also uses the first 16 bytes as an RC4 key to decrypt the remaining data. The format is shown below in Table 2.

| **16 Bytes** | **4 Bytes** | **16 Bytes** | **M Bytes** |
|---|---|---|---|
| RC4 key | Encrypted data size (M) | RC4 key | Encrypted data |

**Table 2. Format of Encrypted DoppelPaymer ProcessHacker Related Modules** After decryption, the first 4 bytes are the size of the compressed data, and the next 4 bytes are the size of the uncompressed data, followed by the compressed data as shown in Table 3.

| **4 Bytes** | **4 Bytes** | **N Bytes** |
|---|---|---|
| Compressed size (N bytes) | Uncompressed size | Compressed 32-bit and 64-bit Process Hacker modules |

**Table 3. Format of Encrypted DoppelPaymer ProcessHacker Related Modules Header and Data** The data is decompressed using aPLib, which produces the PE files in a custom structured format, where each PE contains an 8-byte header consisting of a magic 4-byte value, followed by another 4-byte value that specifies the size of the following PE data as shown in Table 4.

| **4 Bytes** | **4 Bytes** | **N Bytes** | **4 Bytes** | **4 Bytes** | **N Bytes** | |
|---|---|---|---|---|---|---|
| Magic value 1 | Size of Module 1 | Module 1 | Magic value 2 | Size of Module 2 | Module 2 | ... |

**Table 4. DoppelPaymer ProcessHacker Packed Module Format** Table 5 contains the magic value and SHA256 hash for each ProcessHacker component.

| **Magic Value** | **SHA256** | **Description** |
|---|---|---|
| `0xf03d9386` | `51d8618ec86159327e883615ad8989c7638172cf801f65ab0367e5b2e6af596a` | DoppelPaymer's ProcessHacker Stager DLL (32-bit) |
| `0xa68d9640` | `d4a0fe56316a2c45b9ba9ac1005363309a3edc7acf9e4df64d326a0ff273e80f` | ProcessHacker3 (32-bit) |
| `0x53e9cd92` | `0f97f6d53fff47914174bc3a05fb016e2c02ed0b43c827e5e5aadba2d244aecc` | KProcessHacker3 Kernel Driver (32-bit) |
| `0x2fb0f795` | `bfb7e62ba4ad5975e68a1beefb045cb72e056911fd7a8b070a15029dfcbbefe1` | DoppelPaymer's ProcessHacker Stager DLL (64-bit) |
| `0x7900f253` | `bd2c2cf0631d881ed382817afcce2b093f4e412ffb170a719e2762f250abfea4` | ProcessHacker3 (64-bit) |
| `0x8c64a981` | `70211a3f90376bbc61f49c22a63075d1d4ddd53f0aefa976216c46e6ba39a9f4` | KProcessHacker3 Kernel Driver (64-bit) |

**Table 5. Encrypted PE Files Embedded in DoppelPaymer** After decompression, all three binaries are written to the same directory. Both ProcessHacker and the kernel driver are written as random filenames, but the stager DLL filename is chosen to be one of the DLL names imported by ProcessHacker. DoppelPaymer then executes ProcessHacker which loads the stager

DLL via DLL search order hijacking. Once loaded, ProcessHacker's kernel driver is leveraged to kill the blacklisted processes.

## DoppelPaymer Links to "Dridex 2.0"

A Dridex loader sample, identified by SHA256 hash `813d8020f32fefe01b66bea0ce63834adef2e725801b4b761f5ea90ac4facd3a`, was distributed through the *Emotet* malware on June 4, 2019. The Dridex sample contained code to decrypt either a 32-bit or a 64-bit core bot module from its `sdata` section using the exact same encryption, compression, and data format (previously described) that DoppelPaymer uses to extract PEs from its `sdata` section. This observation ties this Dridex variant directly with DoppelPaymer. The Dridex sample was also unusual; not only because the Dridex loader was bundled with the bot core module (rather than dynamically retrieving it from a C2 server), but also because the bot core module had a version number of 2.0.0.78. We have seen subsequent updates to this new variant of the Dridex bot core module with the latest version being 2.0.0.80 at the time of writing. Of note, prior samples of Dridex had a version number of 4.0.0.87. It's unclear why the malware author decided to use lower version numbers, but one explanation is that the threat actor views this new creation as "Dridex 2.0."

## Conclusion

Both BitPaymer and DoppelPaymer continue to be operated in parallel and new victims of both ransomware families have been identified in June and July 2019. The parallel operations, coupled with the significant code overlap between BitPaymer and DoppelPaymer, indicate not only a fork of the BitPaymer code base, but an entirely separate operation. This may suggest that the threat actor who is operating DoppelPaymer has splintered from INDRIK SPIDER and is now using the forked code to run their own Big Game Hunting ransomware operations.

#### Additional Resources

- *For more information on how to incorporate intelligence on dangerous threat actors into your security strategy, please visit the CrowdStrike Falcon® Intelligence product page.*
- *Download the CrowdStrike 2021 Global Threat Report*
- *Test CrowdStrike next-gen AV for yourself. Start your free trial of Falcon Prevent™ today.*

## Appendix/Indicators

| Indicator | Description |
|---|---|
| `801b04a1504f167c25f568f8d7cbac13bdde6440a609d0dcd64ebe225c197f9b` | DoppelPaymer SHA256 hash |
| `813d8020f32fefe01b66bea0ce63834adef2e725801b4b761f5ea90ac4facd3a` | Dridex 2.0 SHA256 hash |

**Table 6. DoppelPaymer and Dridex 2.0 IOCs**

| CRC32 | String | CRC32 | String | CRC32 | String |
|---|---|---|---|---|---|
| `0xc622a2b1` | `acronisagent` | `0x5c6cd7ac` | `msexchangeum` | `0xe381a459` | `epredline` |
| `0xa8e4e8c2` | `backupexecagentaccelerator` | `0xab07d275` | `msexchangeumcr` | `0xe7a6b2c5` | `mozyproba` |
| `0x6d7d9112` | `backupexecdevicemediaservice` | `0xe3d46892` | `mssqlserver` | `0xdf73ec1c` | `masvc` |
| `0xfef41240` | `backupexecjobengine` | `0xf203a569` | `msdtsserver` | `0xcc5f5bf1` | `macmnsvc` |
| `0x6c99d156` | `backupexecmanagementservice` | `0x6d90a649` | `mysql57` | `0x467255e4` | `mfemms` |
| `0x8ff434f5` | `backupexecrpcservice` | `0x2181c15e` | `osearch15` | `0x0f2ae79c` | `psqlwge` |
| `0xc08e25a9` | `backupexecvssprovider` | `0xcdf97a8b` | `oracleclientcache80` | `0x7e26520a` | `swprv` |

| | | | | | |
|---|---|---|---|---|---|
| 0x634332ff | dfsr | 0xcaff10b3 | quickbooksdb25 | 0x656c0e35 | wsbexchan |
| 0xfd7e1ab0 | epintegrationservice | 0x00c7b7a9 | spadminv4 | 0xde2373de | winvnc4 |
| 0x69e7bca5 | epprotectedservice | 0x66a8eead | spsearchhostcontroller | | |
| 0x68507185 | epsecurityservice | 0x46b607c2 | sptracev4 | | |
| 0x5809f6f7 | epupdateservice | 0xdb1ac7bb | spusercodev4 | | |
| 0xc2b55fa6 | mb3service | 0xf3c045e4 | spwriterv4 | | |
| 0xeca1f89e | msexchangees | 0xd917e4cb | sqlbrowser | | |
| 0x44dda068 | msexchangemgmt | 0x23bc321e | sqlsafeolrservice | | |
| 0xbebe6687 | msexchangemta | 0x9626475b | sqlserveragent | | |
| 0x03803c01 | msexchangesa | 0xf76fde75 | sqltelemetry | | |
| 0x0de53e33 | msexchangesrs | 0x9626475b | sqlserveragent | | |
| 0x822dd426 | msexchangeadtopology | 0x25a92500 | sqlwriter | | |
| 0xacedcdb8 | msexchangedelivery | 0x243d4975 | syncoveryvssservice | | |
| 0x9060bcd4 | msexchangediagnostics | 0xc2a56207 | veeambackupsvc | | |
| 0x50f0d551 | msexchangeedgesync | 0x8dbf54db | veeamcatalogsvc | | |
| 0xa300bbb0 | msexchangehm | 0x82d1c632 | veeamcloudsvc | | |
| 0x3040bb72 | msexchangehmrecovery | 0xb97407ef | veeamendpointbackupsvc | | |
| 0x4014b792 | msexchangeis | 0x0aabacba | veeamenterprisemanagersvc | | |
| 0x7e7e47bc | msexchangemailboxreplication | 0x43d71e6c | veeammountsvc | | |
| 0x23a626e2 | msexchangerpc | 0x0c6574ad | veeamnfssvc | | |
| 0xa323c785 | msexchangerepl | 0x2491fd1c | veeamrestsvc | | |
| 0xbfec4da3 | msexchangeservicehost | 0xe076d4a9 | veeamtransportsvc | | |
| 0xbe3d66d5 | msexchangetransport | 0xd67d1e60 | epag | | |

**Table 7. DoppelPaymer Email Server, Backup, and Database Software CRC32 Blacklist**

| CRC32 | String | CRC32 | String | CRC32 | String |
|---|---|---|---|---|---|
| 0xae5a22b4 | dropbox.exe | 0xdc40adba | onenote.exe | 0x306d51a0 | sidebar.exe |
| 0x6274fa64 | cis.exe | 0x4107aa76 | oracle.exe | | |
| 0xf62526b9 | cistray.exe | 0xbfdf529e | postgres.exe | | |

**Table 8. DoppelPaymer Antivirus, Backup, Database, and Windows Tool CRC32 Blacklist**

| CRC32 | String | CRC32 | String | CRC32 | String |
|---|---|---|---|---|---|
| 0x45a1c197 | windefend | 0x0b4fa6cf | msmpsvc | 0xe067db30 | mcafeeframework |
| 0x987163e9 | wdnissvc | 0x360b9799 | sentinelagent | 0xfc95ba9d | mcafeeframeworkmcafeeframewo |
| 0x34220c33 | cylancesvc | 0xde3dabc7 | ekrn | 0x360b9799 | sentinelagent |

| | | | | | |
|---|---|---|---|---|---|
| 0x59d2dbbf | mbamservice | 0xb78f9b4e | wrsvc | 0x3b9f1b3e | sentinelhelperservice |
| 0x27462fff | mbendpointagent | 0x23b07ca0 | vipre business service | 0xa6772c96 | sentinelstaticengine |
| 0x93a7f221 | sbamsvc | 0x9a4f7f43 | mcafeeengineservice | | |

**Table 9. DoppelPaymer Endpoint Security Software CRC32 Blacklist**

| CRC32 | String | CRC32 | String |
|---|---|---|---|
| 0xf26f12c8 | zonealarm.exe | 0xcff1c71e | fortiwf.exe |
| 0x993f5471 | a2guard.exe | 0x64760001 | nortonsecurity.exe |
| 0xd5345e50 | a2service.exe | 0x43c3c112 | bullguard.exe |
| 0xc459d010 | a2start.exe | 0x0d71efa0 | bullguardbhvscanner.exe |
| 0x0b02ef94 | avastsvc.exe | 0xa7dd5f59 | bullguardscanner.exe |
| 0x21579df3 | avshadow.exe | 0x77a2fba9 | bullguardtray.exe |
| 0x6b68c4c6 | avastui.exe | 0x50dbcbda | bullguardupdate.exe |
| 0x0108a03e | fortiesnac.exe | 0x6e7d6782 | avira.servicehost.exe |
| 0x830b705a | fortiproxy.exe | 0xb8894b22 | avira.systray.exe |
| 0xca2d58f0 | fortisslvpndaemon.exe | 0x40cb21d3 | avp.exe |
| 0xe2c0fe91 | fortitray.exe | 0xb018d47e | mbcloudea.exe |

**Table 10. DoppelPaymer Security Software CRC32 Blacklist 1**

| CRC32 | String | CRC32 | String |
|---|---|---|---|
| 0x1a2124c0 | msascuil.exe | 0x895abd73 | nod32.exe |
| 0x456b109f | wrsa.exe | 0x2fba3706 | mcshield.exe |

**Table 11. DoppelPaymer Security Software CRC32 Blacklist 2**

---

Source: https://www.crowdstrike.com/blog/doppelpaymer-ransomware-and-dridex-2/