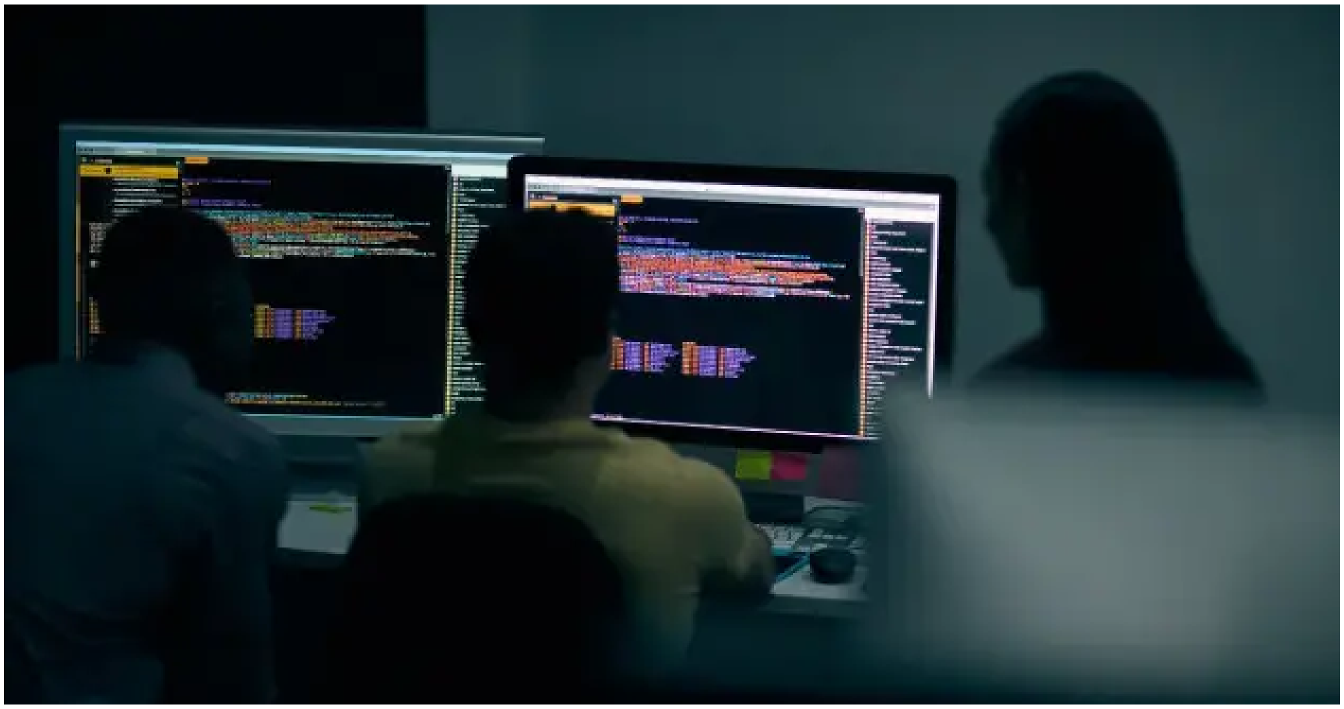
The Trickbot/Conti crypters: Where are they now?

securityintelligence.com/posts/trickbot-conti-crypters-where-are-they-now/



Threat Intelligence June 27, 2023

By Charlotte Hammond Ole Villadsen 23 min read

Despite Conti shutdown, operators remain active and collaborative in new factions

In IBM Security X-Force, we have been following the crypters used by the Trickbot/Conti syndicate, who we refer to as ITG23, since 2021 and demonstrated the intelligence that can be revealed through tracking their use in a blog we published last May. One year on, ITG23 has experienced many organizational changes, splintering into factions and forging new relationships. Despite these events, ITG23 crypters remain fundamental to tracking post-ITG23 factions and their activity; so much so that we believe identifying and tracking the crypters is just as important, if not even more so, than tracking the malware itself. Our research indicates that while ITG23 may have fractured apart after shutting down Conti, many of its various members continue to be very active — still communicating amongst themselves and using shared infrastructure:

- ITG23-related factions, including Quantum, Royal, Zeon, and BlackBasta, continue to use many of the same crypters — plus a few new ones — with their tools and malware, highlighting the ongoing cooperation between former members of the syndicate and their continued access to wider resources available to the post-ITG23 collective.
- Our research into the crypters uncovered several new malware families in use by former ITG23 members and their factions, reflecting the new relationships established with other criminal gangs over the past year.

In this blog, we revisit the crypters to see where they are being deployed and examine how their presence in recent campaigns can provide insight into the current workings of the syndicate.

Background and assessment

Last year we published an article analyzing 13 Trickbot/Conti (aka ITG23) crypters and how they were used by the syndicate's "friends and family," providing us with a means to identify and track their malware and activity. According to our original analysis, the presence of these crypters on a file sample meant that its developer, distributor, or operator was either a part of ITG23 or had a partnership with the syndicate. Much has happened to ITG23 over the past year, begging the question “Where are the crypters now?”

Before we answer that question, it is important to review some background on ITG23 and its journey, especially over the past year. ITG23 had always been more of a decentralized syndicate — a “group of groups” — rather than a single cohesive organization. The syndicate is known primarily for developing the Trickbot banking trojan in 2016 to facilitate online banking fraud. It later developed and operated new malware such as BazarLoader, Anchor, and Bumblebee, which were used to gain a foothold in victim environments for ransomware attacks after developing and operating the Ryuk, Conti, and Diavol ransomware operations. ITG23 also had close partnerships with other criminal gangs, including the developers and distribution groups behind Emotet, Qakbot, Gozi, and IcedID.

ITG23 suffered a series of disruptive events at the start of 2022 — including the Russian invasion of Ukraine and two high-profile leaks on ITG23 activity and members, referred to as the ContiLeaks and TrickLeaks — which culminated in the group shutting down their Conti ransomware operation in May 2022. Following these events, the syndicate reportedly fractured even further into multiple groups we call “factions” such as Quantum, Royal, Zeon, BlackBasta, and Silent Ransom. These new factions established their own operations, deploying ransomware or engaging in data theft and extortion. Connected to this, we have seen them experimenting with new malware and tactics, and forging new relationships. Some factions began developing or operating new malware strains, including commodity malware, and some affiliates are likely using other ransomware as a service operations, such as Nokoyawa.

Over the past year, the relationships and boundaries between the factions have continued to shift, with Quantum disappearing and likely becoming part of Royal. Other groups previously affiliated with ITG23, such as the data extortion group Karakurt, also continue to operate.

So where are they now?

Over the past year, the syndicate’s crypters discussed in our original blog are still being used in attacks by many ITG23-related factions and their partners, affiliates, and initial access brokers, and they continue to provide a critical window into their activities. Our key findings include:

Old Ties Still Going Strong — Former ITG23 members and affiliates behind Quantum, Royal, Zeon, BlackBasta, and Karakurt attacks have maintained access to these crypters. The use of the crypters by members of multiple former ITG23 factions signals a high level of communication and cooperation between these actors, challenging the assumption that the new factions are all separate or distinct groups.

- **Shared Resources** — Post-ITG23 factions and their partners continue to have access to the crypters for use with their initial access and post-compromise tools and malware, which makes them incredibly useful for identifying and tracking activity associated with these factions. Over the past year, these crypters have been used on the same initial access malware previously used during ITG23 attacks, including Emotet, IcedID, Qakbot, Bumblebee, and Gozi. They have also been used on tools and malware employed during attacks by former ITG23 members and affiliates, including CobaltStrike, NetSupport, and Ligolo-NG.
- **The Eight “MIA”** — Eight of the crypters have not been seen since the first half of 2022 suggesting that the events of early 2022 may have significantly disrupted aspects of the group’s crypting operation. The remainder of the crypters continue to be used and updated, and we have also seen the emergence of new crypters we have dubbed Snow and Forest.
- **New Uses** — The crypters have appeared on new malware over the past year that have been used for initial access or information stealing, including SVCReady, CargoBay, Matanbuchus, Pikabot, Aresloader, Vidar, Minodo, and LummaC2 Stealer. We have separately tied much of this malware to Quantum, Royal, Zeon, and BlackBasta attacks. They have also been used with the ransomware itself, including Quantum, Royal, BlackBasta, and Nokoyawa.

The crypters

As described in our previous report, crypters, which are also referred to as loaders or packers, are applications designed to encrypt and obfuscate malware to evade detection by antivirus (AV) scanners and hinder analysis. Crypters generally operate by encrypting the pre-compiled malware payload and embedding it within a secondary binary, which we refer to as a loader. The loader contains code to decrypt and execute the malicious payload, and may also include additional sandbox-evasion or anti-analysis functions. The loaders are often designed to evade AV and signature-based detection, and will often make use of obfuscation and code-morphing techniques that render each compiled loader different from a code perspective, increasing the challenge of writing effective signatures. The use of crypters allows malware developers to easily experiment with different methods of evading antivirus detection without having to make changes to the malware itself.

As of May last year, we were tracking 13 crypters that we had attributed to ITG23, and that had been used with malware built or operated by ITG23 and their "friends and family." Eight of these crypters have not been seen since the first half of 2022, and their retirement may be linked to the disruption experienced by the group during that time period. We also identified two new crypters which we were able to attribute to former ITG23 developers. The first is Forest, which was introduced in March 2022 and is sometimes known as the Bumblebee Loader, due to its prolific usage with the Bumblebee malware. The second is Snow, which was first observed in December 2022. Its introduction coincided with the retirement of Hexa, and code overlap between the two indicates that Snow is likely Hexa's successor.

Over the past year, we also identified some noteworthy trends regarding crypter use on malware, including their use with several new malware families. Previously, the crypters were used predominately with the core malware families associated with ITG23 and their close partners; this included Trickbot, Emotet, BazarLoader, IcedID, CobaltStrike, and the Ryuk, Conti, and Quantum ransomware strains. However, the fracturing of ITG23 and emergence of new factions, relationships, and methods, have affected how the crypters are used.

In early 2022, the syndicate began to develop stronger ties with Qakbot, including the first use in February 2022 of ITG23 crypters with Qakbot payloads. Qakbot previously had been using its own set of crypters, including the crypter-as-a-service CryptOne and two others that we call Quartz and Quixotic. Over the course of 2022, Qakbot continued to use ITG23-related crypters alongside their own; Hexa was used predominately through May 2022 followed by Forest until the introduction of Snow in December 2022. In 2023, Qakbot's use of ITG23-related crypters, in particular the new Snow crypter, increased steadily to the point that Qakbot now uses Snow almost exclusively. Early 2022 also saw the introduction of the Bumblebee malware, indicating a potential relationship between ITG23 and the developers of the Ramnit malware. Bumblebee was released alongside the new Forest crypter, which it uses almost exclusively to the present day.

In April 2022, we observed the first use of an ITG23 crypter with the Gozi banking trojan, which we linked back to a campaign operated by Hive0106 (TA551) with whom ITG23 had an established relationship. Like Qakbot, crypted Gozi payloads have increased steadily throughout the past year, during which we have observed crypters such as Hexa, Forest, Snow, Lore and Dave used on Gozi malware, most often with LDR4 and Cutwail botnet distributions. Interestingly, in 2023 we also observed the Dave and Forest crypters used with Pushdo, a downloader tied to the Cutwail botnet.

From June 2022 onwards, we began to see an uptick in new malware families being used with the crypters, likely a consequence of the shutdown of the Conti ransomware strain and the emergence of the new factions. Some of these factions likely forged new relationships with other criminal gangs, in turn leading to the testing and use of new malware, such as SVCReady, CargoBay, and Matanbuchus, on which we have observed crypters such as Hexa and Dave deployed.

In 2023, this trend continued and we observed ITG23-related crypters deployed on a range of new malware, reflecting the continued focus on building new relationships with other threat actors to purchase and use new malware strains. These new families included information stealers (Lumma C2, Vidar), backdoors/downloaders (Aresloader, Canyon), and malware acquired from FIN7 developers such as Minodo and Diceloader.

More information on the crypters and associated malware can be found in the below sections. The following table also provides an overview of the current status of the core ITG23-related crypters that we track.

Status	Name	Activity	Malware Families from 2022 Report	Malware Families from Past Year
New/Active	Forest	High use. First introduced with Bumblebee.	N/A	Ligolo-NG, IcedID, Qakbot, Bumblebee, CobaltStrike, Gozi, PikaBot, Pushdo, BlackBasta Ransomware
New/Active	Snow	High use. Code overlap indicates this is likely the successor of Hexa.	N/A	IcedID, Qakbot, Gozi, Pikabot
Active	Dave	High use. Under regular development with multiple variants.	CobaltStrike, Colibri, Emotet, Trickbot, BazarLoader, Conti, Ryuk	Emotet, IcedID, CobaltStrike, SVCReady, CargoBay, Pushdo, Minodo, DiceLoader, AresLoader, LummaC2, Vidar, Gozi, Canyon, Nokoyawa Ransomware, BlackBasta Ransomware.
Active	Tron	Occasional use. Has received a few updates.	CobaltStrike, Conti, Emotet, Trickbot, BazarLoader, IcedID	CobaltStrike/Metasploit stagers primarily. Also, BlackBasta Ransomware and occasionally IcedID and NetSupport
Active	Lore	Occasional use. No significant updates to functionality.	CobaltStrike, Conti, IcedID, Emotet, Trickbot, BazarLoader	IcedID primarily (especially Forked variants). Occasionally, Qakbot, Gozi, Royal Ransomware, CobaltStrike
Limited activity	Mirror	Presumed retired in May 2022. Seen rarely during 2023.	BazarLoader, IcedID, CobaltStrike	IcedID
Retired	Hexa	Received update in August 2022. Last seen December 2022. Likely superseded by Snow.	IcedID, Qakbot, BazarLoader, CobaltStrike, Conti, Gozi	Payloads from May 2022 → Dec 2022. IcedID, CobaltStrike, SVCReady, Qakbot, Matanbuchus, Quantum Ransomware, Gozi.
Retired	Pear	Last seen Q1 2022	IcedID, BazarLoader, Colibri, Trickbot	N/A
Retired	Galore	Last seen Q1 2022	BazarLoader, IcedID, CobaltStrike	N/A
Retired	Rustic	Last seen Q1 2022	Sliver, CobaltStrike, BazarLoader, IcedID, Quantum	N/A
Retired	Stub	Last seen Q1 2022	BazarLoader, Quantum, CobaltStrike, IcedID, Conti	N/A
Retired	Error	Last seen Q1 2022	Emotet, BazarLoader, CobaltStrike, IcedID	N/A

Status	Name	Activity	Malware Families from 2022 Report	Malware Families from Past Year
Retired	Skeleton	Last seen Q2 2022	Trickbot, IcedID, CobaltStrike, Quantum	N/A
Retired	Charm	Last seen Q1 2022	BazarLoader, CobaltStrike, Conti, Quantum, Trickbot	N/A
Retired	Graven	Last seen Q1 2022	BazarLoader, Emotet, IcedID	N/A

Scroll to view full table

Forest

Forest was first seen in March 2022 when it was used predominantly with the newly released Bumblebee malware. To this day, Forest is one of the only ITG23 crypters configured for use with the Bumblebee malware, and it is likely that it was specifically designed for that purpose; however, its capabilities were soon expanded to handle additional payload types.

Over the past year, Forest has been used with a variety of malware and tools routinely associated with ITG23-related attacks, including Bumblebee, IcedID, CobaltStrike, Qakbot, and Pikabot. It has also been used with Pushdo malware as well as Gozi loaders distributed by the Cutwail botnet — signaling a close relationship between these actors. Of interest, we found a Ligolo-NG tunneling tool using the Forest crypter; Ligolo-NG has been used by actors affiliated with the Karakurt data extortion group, which in turn has been tied to ITG23.

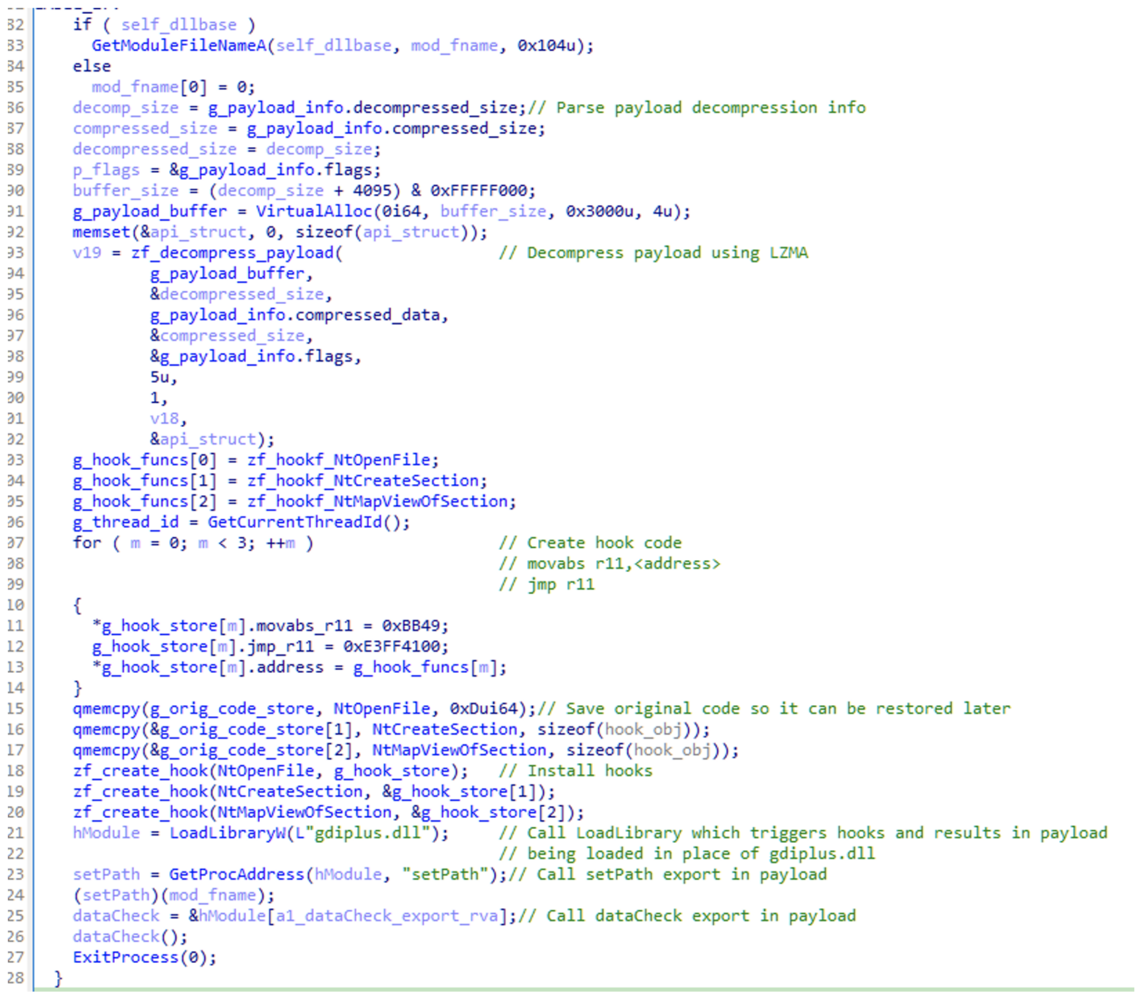
Like other crypters, Forest's use extends across different factions and partners affiliated with ITG23. A Forest-crypted CobaltStrike sample contacted a command and control (C2) domain used during a Nokoyawa ransomware attack in October 2022. This attack commenced with a Hexa-crypted IcedID loader likely distributed by the Cutwail botnet based on our analysis of the distribution emails. Other reporting indicates that Nokoyawa has been used by former Conti affiliates.

In April 2023, we identified a Forest-crypted CobaltStrike sample contacting C2 infrastructure that has been used exclusively in Qakbot attacks leading to BlackBasta ransomware. We also identified a Forest-crypted BlackBasta ransomware sample from late May 2023.

From a technical perspective, Forest stores its payload in multiple chunks across the data sections of the binary. Each chunk of data is decrypted/unpacked using a custom algorithm and then the outputs combined together and further decrypted using a XOR based algorithm. Some variants also include an additional layer of compression using the LZMA algorithm.

The loader is capable of executing both shellcode and PE payloads, with shellcode payloads executed directly and PE payloads executed indirectly via a hooking technique. The loader installs hooks within the library functions NtOpenFile, NtCreateSection and NtMapViewOfSection, such that when these APIs are called the loader's own functions will be executed instead. The loader will then attempt to load the library 'gdiplus.dll' using the LoadLibrary API, which in turn calls the above-mentioned NT APIs and triggers the hooks. The hook functions copy the unpacked payload into a newly created section and return the base address to LoadLibrary, which proceeds to load the malicious payload in place of the legitimate gdiplus.dll.

In newer versions of Forest, the hooking code has all been moved into a secondary shellcode stager, which is stored encrypted along with the final payload. Forest decrypts the payload, and then transfers execution to the shellcode which performs the rest of the loading process. A second shellcode loader has also been observed, which consists of just a basic PE loader and is used primarily with non-Bumblebee payloads, whereas the hooking shellcode seems to be more commonly used with Bumblebee.



```
  if ( self_dllbase )
    GetModuleFileNameA(self_dllbase, mod_fname, 0x104u);
  else
    mod_fname[0] = 0;
  decomp_size = g_payload_info.decompressed_size;// Parse payload decompression info
  compressed_size = g_payload_info.compressed_size;
  decompressed_size = decomp_size;
  p_flags = &g_payload_info.flags;
  buffer_size = (decomp_size + 4095) & 0xFFFFF000;
  g_payload_buffer = VirtualAlloc(0i64, buffer_size, 0x3000u, 4u);
  memset(&api_struct, 0, sizeof(api_struct));
  v19 = zf_decompress_payload(                // Decompress payload using LZMA
          g_payload_buffer,
          &decompressed_size,
          g_payload_info.compressed_data,
          &compressed_size,
          &g_payload_info.flags,
          5u,
          1,
          v18,
          &api_struct);
  g_hook_funcs[0] = zf_hookf_NtOpenFile;
  g_hook_funcs[1] = zf_hookf_NtCreateSection;
  g_hook_funcs[2] = zf_hookf_NtMapViewOfSection;
  g_thread_id = GetCurrentThreadId();
  for ( m = 0; m < 3; ++m )                   // Create hook code
                                              // movabs r11,<address>
                                              // jmp r11
  {
    *g_hook_store[m].movabs_r11 = 0xBB49;
    g_hook_store[m].jmp_r11 = 0xE3FF4100;
    *g_hook_store[m].address = g_hook_funcs[m];
  }
  qmemcpy(g_orig_code_store, NtOpenFile, 0xDui64);// Save original code so it can be restored later
  qmemcpy(&g_orig_code_store[1], NtCreateSection, sizeof(hook_obj));
  qmemcpy(&g_orig_code_store[2], NtMapViewOfSection, sizeof(hook_obj));
  zf_create_hook(NtOpenFile, g_hook_store);   // Install hooks
  zf_create_hook(NtCreateSection, &g_hook_store[1]);
  zf_create_hook(NtMapViewOfSection, &g_hook_store[2]);
  hModule = LoadLibraryW(L"gdiplus.dll");     // Call LoadLibrary which triggers hooks and results in payload
                                              // being loaded in place of gdiplus.dll
  setPath = GetProcAddress(hModule, "setPath");// Call setPath export in payload
  (setPath)(mod_fname);
  dataCheck = &hModule[a1_dataCheck_export_rva];// Call dataCheck export in payload
  dataCheck();
  ExitProcess(0);
}
```

Fig. 1 — Forest Loader code which decompresses a payload using LZMA, installs hooks within API functions, and then calls LoadLibrary to trigger the hooks and load the payload

Example hashes

Date	Payload	Hash
May 2023	Gozi Loader	ea2d71af9790b0a058d0d166c52c2609a1a106053189c515b6059b5f18e9e48b
May 2023	Pushdo	a6807d559eedefff6ff1d9d7e90e5765d1a0a1843139ec8eb03527b60e0630e4
May 2023	IcedID	c12d0d30e6b1b5567ceafab35f60f0ce7893f75c29bcaf8021a32035131b9d05
May 2023	Bumblebee	f5cbfffa43e8280cd9b68bea2c612adb5aa47fe802d28db48dfd1d9291f4ad71
May 2023	CobaltStrike	67c4c38819efe5855a77fda662d392588c3f38305bcda0fb7fcd78784f4a10c6

Date	Payload	Hash
May 2023	Pikabot	92153e88db63016334625514802d0d1019363989d7b3f6863947ce0e490c1006
May 2023	BlackBasta Ransomware	4e4129bb225622380646f9c25deea6403536bc9705e00d511adaae2715398923
April 2023	Qakbot	7e5725f8f67d9f8c622eea620a1bb0bc330701411530b20d952b22debeea9357
May 2022	Ligolo-NG	f58aae183e6953f202ecf6df908fc68c77fd72d6a69cc9f2132e8817ceae7f62

Scroll to view full table

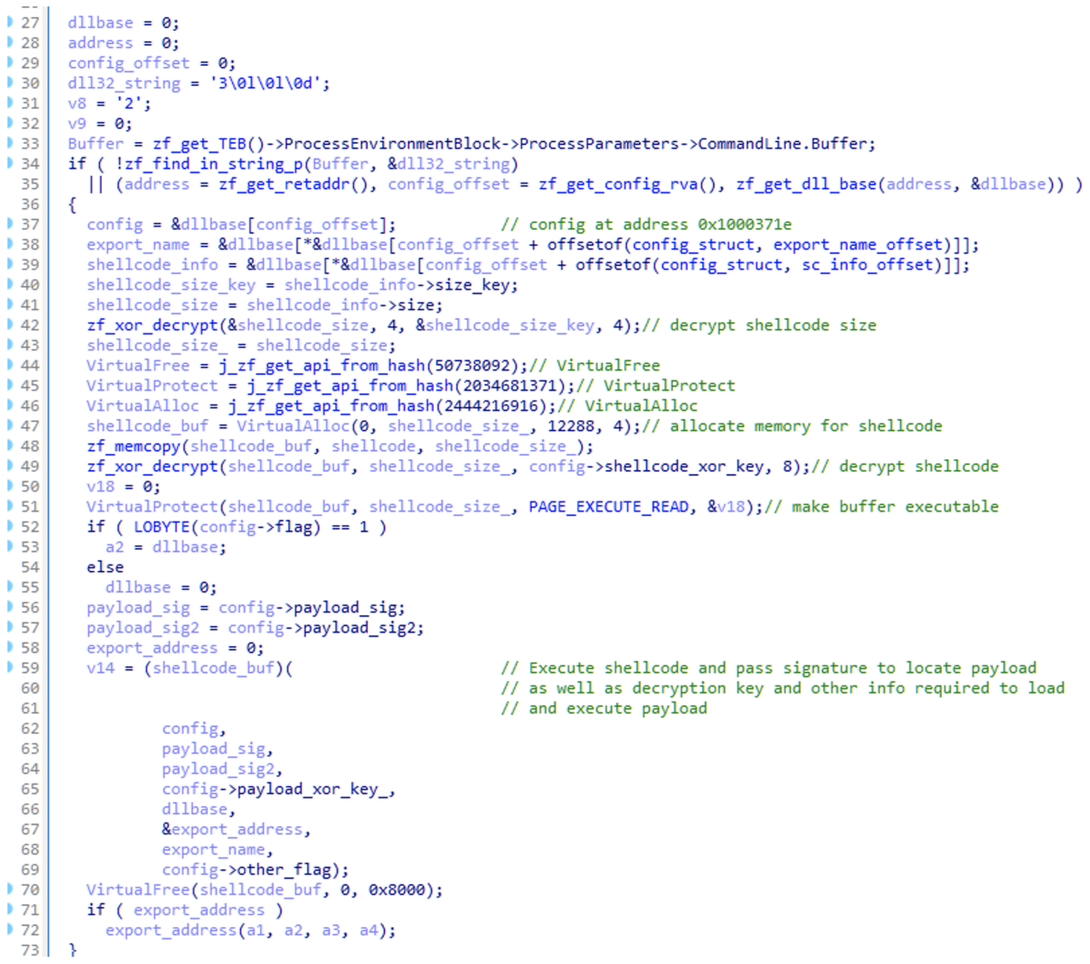
Snow

Snow crypter was first observed in December 2022 when it was used with IcedID, which prior to this time had been relying primarily on the Hexa crypter. Snow is thought to be the direct successor of Hexa, as they share some code overlap and the appearance of Snow coincided with Hexa's retirement.

The Snow crypter was used almost exclusively with IcedID and Qakbot from its debut in December 2022 to May 2023, when we observed Snow being used for the first time to crypt Gozi LDR4 and Pikabot malware. We also identified a DLL in March 2023 which we determined to be the Snow loader converted from a payload loader into a downloader. The sample was attached to an email providing instructions about an extortion attempt and was described as an Active Directory listing of stolen information. We assess this modified Snow loader was used during a fake extortion campaign affiliated with Royal ransomware.

Snow loader samples often use a legitimate executable as a base, and several of the exported functions of the executable are overwritten with the malicious Snow loader code. The majority of Snow's functionality is contained within a secondary shellcode loader, which shares code overlap with that used by later Hexa samples.

A configuration block within the Snow loader contains offsets for the XOR-encrypted shellcode, a signature for locating the final payload, and decryption keys. The initial Snow loader code uses these values to decrypt the shellcode, which it then executes. The shellcode then locates the final payload by searching for the signature value, decrypts it using a XOR and SUB-based algorithm, and decompresses it using QuickLZ. Finally, the PE-based payload is loaded into memory and executed.



```
dllbase = 0;
address = 0;
config_offset = 0;
dll32_string = '3\01\01\0d';
v8 = '2';
v9 = 0;
Buffer = zf_get_TEB()->ProcessEnvironmentBlock->ProcessParameters->CommandLine.Buffer;
if ( !zf_find_in_string_p(Buffer, &dll32_string)
  || (address = zf_get_retaddr(), config_offset = zf_get_config_rva(), zf_get_dll_base(address, &dllbase)) )
{
  config = &dllbase[config_offset];           // config at address 0x1000371e
  export_name = &dllbase[*&dllbase[config_offset + offsetof(config_struct, export_name_offset)]];
  shellcode_info = &dllbase[*&dllbase[config_offset + offsetof(config_struct, sc_info_offset)]];
  shellcode_size_key = shellcode_info->size_key;
  shellcode_size = shellcode_info->size;
  zf_xor_decrypt(&shellcode_size, 4, &shellcode_size_key, 4);// decrypt shellcode size
  shellcode_size_ = shellcode_size;
  VirtualFree = j_zf_get_api_from_hash(50738092);// VirtualFree
  VirtualProtect = j_zf_get_api_from_hash(2034681371);// VirtualProtect
  VirtualAlloc = j_zf_get_api_from_hash(2444216916);// VirtualAlloc
  shellcode_buf = VirtualAlloc(0, shellcode_size_, 12288, 4);// allocate memory for shellcode
  zf_memcopy(shellcode_buf, shellcode, shellcode_size_);
  zf_xor_decrypt(shellcode_buf, shellcode_size_, config->shellcode_xor_key, 8);// decrypt shellcode
  v18 = 0;
  VirtualProtect(shellcode_buf, shellcode_size_, PAGE_EXECUTE_READ, &v18);// make buffer executable
  if ( LOBYTE(config->flag) == 1 )
    a2 = dllbase;
  else
    dllbase = 0;
  payload_sig = config->payload_sig;
  payload_sig2 = config->payload_sig2;
  export_address = 0;
  v14 = (shellcode_buf)(                      // Execute shellcode and pass signature to locate payload
                                              // as well as decryption key and other info required to load
                                              // and execute payload
          config,
          payload_sig,
          payload_sig2,
          config->payload_xor_key_,
          dllbase,
          &export_address,
          export_name,
          config->other_flag);
  VirtualFree(shellcode_buf, 0, 0x8000);
  if ( export_address )
    export_address(a1, a2, a3, a4);
}
```

Fig. 2 — Snow Loader code which decrypts and executes shellcode, passing information required to identify the final payload. The shellcode then performs the rest of the loading functions.

In March 2023, a variant of Snow was discovered which had been converted into a downloader. The encrypted payload had been replaced with an encrypted data block containing a download URL, and the shellcode had been updated to download and load a payload from the URL. The Snow code had also been updated to require a password, passed via the command line /k parameter, in order to run correctly.

Whilst the download URL was no longer available, analysis of the downloader shellcode revealed that the downloaded data would have been formatted as text and contain a string '**/object/**' followed by base64 encoded data. The base64 data is decoded and then decrypted using XOR.

The decrypted data contains the final payload and its execution parameters delimited by named tags. The following tags were referenced in the code:

- **<content>:** A binary payload to be executed.
- **<export>:** The name of an exported function within the payload to be executed, if applicable.
- **<params>:** Parameters for running the payload if executed via the command line.
- **<name>:** The filename to use for the payload, if it is dropped to disk.
- **<drop_disk>:** A flag indicating whether the payload should be dropped to disk.
- **<frd_dll>:** A flag related to loading the payload DLL into memory.

If the **drop_disk** flag is set within the downloaded data, then the malware proceeds to write the payload binary to the **%Temp%** directory with the provided filename. It then creates a new process to execute the payload using the command line parameters given in the **params** field.

Alternatively, if the **drop_disk** flag is not set, then the malware treats the payload as a DLL, and proceeds to load it into the memory of the current process. Once loaded, the payload's entrypoint function is executed, followed by an exported function if specified.

In April 2023, a second variant of Snow was identified. This version of the loader included the same command-line password functionality seen in the downloader, and also included a sophisticated second stage shellcode designed to inject the third stage shellcode, which is the standard Snow PE loader shellcode from previous versions, into another running process such as svchost.exe.

The injector shellcode uses direct syscalls for most of its significant API calls and makes heavy use of the NtQueueApcThread API in order to run code within the context of pre-existing legitimate threads within the target process.

This variant of Snow is rarely seen, having been observed in use with only a handful of IcedID samples. The vast majority of malware activity utilising the Snow crypter uses the original version of the loader.

Example hashes

Date	Payload	Hash
May 2023	Qakbot	09ed2cf56af8385c87f297c2a4f168efdfc78434b8a42a9122328e775f5f0400
May 2023	Pikabot	e2723661efa1115c81bb13238b5925422ef3abf89909e005f7da6c4671d67930
May 2023	Gozi Loader	e33a713b96b45e2b2e0da350c0fdaaf865139607066aadff3b67b0ced82ca8bc
May 2023	IcedID	ea011bab24b88e3beac7eae49de818ddc4024f4c0de8f37cda0c26e10a72cc5e
March 2023	Snow Downloader	78bb0fd18def2602188ca0004ac5428ed039b8abef4926c7e9e9b908a1efa5b8

Scroll to view full table

Dave

Dave is one of the oldest crypters still in use by ITG23-related actors and dates back to at least 2019 when it was used primarily with Emotet and Trickbot. Over the past few years, Dave has continued to be the preferred loader for Emotet malware, though it is common to see it used with a wide range of ITG23 related payloads, especially during time periods when Emotet is offline.

In addition to Emotet, Dave was used in 2022 with CobaltStrike samples contacting C2 domains used in attacks leading to Quantum and Royal ransomware. It was also used to crypt SVCReady, a loader that has been observed in Quantum ransomware attacks, as well as the CargoBay loader, which we have linked to Zeon and Royal ransomware attacks.

In 2023, Dave's use expanded to several additional malware families, including Nokoyawa and BlackBasta ransomware; malware obtained or purchased from FIN7 developers such as Minodo and Diceloader; a new malware family we have dubbed Canyon; Aresloader; and the information stealers Vidar and LummaC2.

Dave Loader is under continuous development and appears to cycle through different variants, though certain features of the loader remain the same and it is generally possible to observe a clear evolution from one version to the next, allowing all the variants to be linked. Depending on which variant of Dave is currently active, its payload is generally stored either as a resource or within one of the data sections of the loader, and is decrypted using XOR or a custom RC4 algorithm. PE-based payloads are usually wrapped in a second-stage shellcode loader, which is based on code from the sRDI project, and the ASCII signature 'dave' may be used to mark the end of the payload.

In May 2022, Dave was updated to a new variant, which we named Fountain until its relationship with Dave was fully established. This variant combined the payload decryption and PE loading code into a single shellcode block, which it stored encrypted in a data section of the loader binary. Upon execution, the loader decrypts the shellcode using XOR and executes it. The payload itself was stored as a HTML, FONTDIR, or RCDATA type resource, and information including the resource ID and type would be passed by the main loader to the shellcode as parameters. The shellcode would then load the payload from the binary’s resources, decrypt it using XOR and the same key used prior, load it into memory and execute it.

Further updates included moving the encrypted shellcode from the data section to instead be stored as a stack-string within the code, and another variant also saw the encrypted payload being stored as multiple base64 encoded segments within the data sections, rather than as a resource.

In January 2023, after a period of downtime over the holidays, Dave returned in a more recognizable form with the XOR-encrypted payload and shellcode moved back to the resources (albeit as two separate resources). Then in April 2023, they returned to the original technique of wrapping the payload within the shellcode along with the ASCII ‘dave’ signature (which had disappeared in the previous variants) and moved the whole XOR-encrypted block back to the data section. At this point, we also witnessed some brand new additions to the Dave loader, as code from the SysWhispers2 library was incorporated as a means of obfuscating the loader’s API calls.

Example hashes

Date	Payload	Hash
May 2023	BlackBasta	d982401b64ae312363fafadcfdedabdd7c13ad89651767c5c6bc0fef03f63fb4
May 2023	CobaltStrike	16fc69a311b88c9133c0e5a66244f80a89a13a78422e3142b2a1b297c46511d6
May 2023	IcedID	6df2ece892c9192c90d4d9fdec768beb17aecfb17d44adc69a11cb50721fa68e
May 2023	LummaC2	e7bffb1369b4ed04cfabffa75d18cda1e78a4504fe0560f742bd8eabb6c02723
May 2023	Aresloader	9f6c4e242c82449e409e076e59f5d597698e2dd0e7fab86a718f815746336bcc
May 2023	Gozi	d50570c1b4d064fb1f6e855d0c27ac1958a7a32c3cef5e6373094d82647f5bd4
May 2023	Nokoyawa	418c1421c2424f152d83aa6886c15c42dd9947b63fcd4544a679eb0477d40dab
April 2023	Qakbot	6d0797aa364d6fdc1aa3a6b4ba318a2dacea72239f381b038385013876c39815
April 2023	Canyon	074ad8fb33dc4da5ce3a58fdd470697973f7131d91d74603a75fdccc8d48c9f0
March 2023	Vidar	f31fb56f187f6e95bb11040e12a3b99234d5a9900c2382d8648d4a0f4fa84217
March 2023	Diceloader	6983afa1c76af7e5ab48ce1c1fe0733749435f31a42c8b8db0b81eed566cb5a9
March 2023	Minodo	ce99b4c0d75811ce70610d39b1007f99560e6dea887a451e08916a4f8cf33678
March 2023	Pushdo	6618359d4d19997728359453b0598be7562c293ef9d6ac51f2635586096a52bd
November 2022	CargoBay	d800e466a5457ca63d18fcaaf8afea7743c94e26c847ba55175d14b1f0da2408
November 2022	Emotet	2d5d18b44d162cc16b3cac6b8824774716f095551c6ddac674953e33e7bec7af
July 2022	SVCReady	f690071e5394aa76f14e2b5cb5cfb15de51d689ed5213e9cf8b931a6721a11c6

Scroll to view full table

Tron

Tron first appeared in the wild in September 2021 when it was used to crypt Trickbot binaries followed by a number of other payloads including Emotet, Trickbot, BazarLoader, IcedID, Conti and CobaltStrike. Since our last report, ITG23-related actors have continued to use Tron fairly consistently, although it has not been observed as frequently as some of the other crypters.

Over the past year, Tron has primarily been used for crypting CobaltStrike and Metasploit stagers which are often linked to attacks leading to ransomware. Like other crypters, its use has not been restricted to a particular ITG23-related faction, as Tron-crypted malware has been observed in incidents linked to BlackBasta, Quantum, and Royal ransomware. In addition to CobaltStrike payloads, Tron has been used to crypt the BlackBasta ransomware itself and has been observed loading IcedID, most notably during campaigns targeting Ukraine in April 2022. It was also observed on a NetSupport sample used during a Royal ransomware attack.

The Tron loader has had a few updates since our last report, but most of its core functionality remains the same. The payload has now been split up into multiple chunks, and several variations of this have been seen. For the two most common variants, the first contains the address of each chunk directly within the payload reconstruction function, and the second uses a linked-list mechanism, whereby the reconstruction function contains the address of the first node only, and then each node contains a pointer to the next one.

Each payload chunk is unpacked using a custom library, described in our previous report, and then the chunks are joined together. Finally, the reconstructed payload is either decompressed using the Zlib library or decrypted using XOR, depending on the variant.

Example hashes

Date	Payload	Hash
April 2023	CobaltStrike	0f1c31eb21e7c95d624eb2fe15978a4bbe08bfaa3256e9a664a80e486cda5b21
March 2023	BlackBasta	1c711ca465dace4d2a8d0542e75410c417375c4ee484294fcd959e99651fccb8
October 2022	NetSupport	abf937fb2f162d1dbbe76c7386c9892db5191e17de586f0a5c49819cd68b5e0f
April 2022	IcedID	ac1d19c5942946f9eee6bc748dee032b97eb3ec3e4bb64fead3e5ac101fb1bc8

Scroll to view full table

Lore

Lore was first observed around May 2021 and had been previously used with payloads including Emotet, Trickbot, BazarLoader, IcedID and CobaltStrike. The Lore loader stores its payload as a BITMAP type resource, with a 103-byte bitmap file header added to the start of the encrypted payload data. Upon execution, the loader code loads the resource, removes the bitmap header, and decrypts the remaining data using XOR and a hard-coded key.

Lore crypter has had low levels of use over the past year, with a large gap from June to December 2022. Since its reappearance, it has been used primarily with the forked variants of IcedID, and occasionally with Royal ransomware, Qakbot and Gozi.

Example hashes

Date	Payload	Hash
June 2023	Royal Ransomware	feb1e25f7d1e26a90af5d1a64f1a790280754ad566655c628abc81f87256ca7f
December 2022	Qakbot	95d2d427251bd10427f078255981bee74ed39b9fde78e0e7f1fc5c7c38ad4a10
February 2023	Gozi Loader	545599c9def124e8c251ecf68389b1a8758f7ab1360a6b371cf8b884776eaa69
April 2023	IcedID IntBot Loader (int-bot.dll)	1f6c62c696570bca30655072930e664ce4b01e344236d169ed41f668eeee3138
April 2023	IcedID Downloader (Loader.dll)	371c2cdb76692d1f4db02a946607bc69d768a8acad42c7e96014eaf18e51e599

Date	Payload	Hash
May 2022	CobaltStrike	65d05aaa7fb82cf6aaa317c50d92a28394442424d573561945e85e0847048535

Scroll to view full table

Mirror

Mirror was first observed in November 2021 and was originally used with BazarLoader, IcedID and Cobalt Strike payloads. Mirror loader shares some code overlap and obfuscation mechanisms with Lore, suggesting that the two may share the same developer or codebase. Mirror's encrypted payload is split into three parts which are stored across different sections of the loader. Two main variants of the Mirror loader were identified, one which decrypts the payload using AES-256 via the Windows CryptDecrypt API, and a second which decrypts the payload using XOR. In addition to encryption, the payload may also be compressed using a simple compression algorithm.

Mirror was presumed retired in May 2022, until February 2023 when a Mirror-crypted IcedID sample was found, which utilised AES encryption plus compression for the payload.

Example hashes

Date	Payload	Hash
February 2023	IcedID	a5fceb009ff1fc6b808407abcdec32741f520b1b86fb0e89d98f664978f3cffa

Scroll to view full table

Hexa

Hexa crypter was first observed towards the end of 2021, with payloads including BazarLoader, Cobalt Strike, and Conti. Over the course of 2022 its use expanded to include malware such as IcedID, Qakbot, Gozi, Quantum ransomware, SVCReady, and Matanbuchus. Hexa was used particularly often with IcedID until December 2022 when the latter switched to using the Snow crypter along with Qakbot.

The early versions of Hexa contained an RC4-encrypted payload which was also compressed using QuickLZ and then encoded as a hexadecimal ascii string to reduce entropy. The encoded payload was then stored in the data sections of the loader binary, with some variants splitting the payload across two or three different sections. The string '|SPL|' was used to mark the payload location, and as such the loader may also be referred to as SPLCrypt by some vendors. Upon execution the payload is reconstructed, decompressed and decrypted and then copied to a newly created memory section and execution transferred to the payload.

In mid-2022 Hexa was updated, and half of the loader code was moved into an encrypted shellcode block. The payload was also split into numerous small chunks, which each chunk preceded by a signature value. At the start of its execution, the loader would calculate a list of signatures, and these would then be used to identify each chunk of the payload and allow for its reconstruction.

As of December 2022, Hexa is considered retired and superseded by Snow crypter.

Example hashes

Date	Payload	Hash
November 2022	IcedID	112f268339b6811b60fd64a3f8f0f9e7fec990510739be6f5438a72ceb1e5f38
August 2022	CobaltStrike	fe0f34664407da15d6a9fe11368484607e75450108525dd9626409c1fd3ecf94
August 2022	SVCReady	f766d2ea0d8124120d712caad5f00ac51114076fa3354fb760ae64aae39147f1

Date	Payload	Hash
May 2022	Qakbot	1e3da157c01f8d3bafe42deed66bdd5cf1f12312f550576b2a32f02da7410b9b
June 2022	Quantum ransomware	4b6b2ba910783cdc587f4339678f9a072fa60bb01cf4297eb09c0b3cfc89a199
June 2022	Matanbuchus	68e20d668c488ab3f988640f942ff519681e5c004ca9ce91c6e5fd2f6ec7ffa5

Scroll to view full table

Other crypter activity

In addition to the ITG23-related crypters above, we are also tracking several additional loaders related to former ITG23 partners or factions. These include three loaders linked to the Qakbot group, which have been used regularly to deploy Qakbot malware; one linked to BlackBasta; and several others which appear to have been used exclusively by a BlackBasta-affiliated team to crypt their CobaltStrike samples.

The Qakbot linked loaders we refer to as Quartz, Quixotic, and Quicksand. All three are written in C++ and appear to have originated from the same developer, based on code overlap and the presence of similar PDB strings. We began tracking Quartz and Quixotic in May 2022, and first observed Quicksand in March 2023.

- Quartz stores its payload in a data section of the loader PE and decrypts it using a sequence of XOR, SUB, and ADD operations with single byte keys. The payload is wrapped in a secondary shellcode loader, which performs the remainder of the PE loading operations. The loader contains large amounts of junk code to obscure the payload decryption functions.
- Quixotic also has its payload stored in a data section and decrypts it using XOR and a key constructed from multiple strings. Some samples include rudimentary execution checks, such as checking for the presence of rundll32 in the command line and won’t produce the correct decryption key string unless the conditions are met. The decrypted payload is wrapped in a secondary shellcode loader, which is the same type used by Quartz, and encoded as base64. The loader decodes the payload and executes the shellcode, which performs the remainder of the loading tasks.
- Quicksand is a more complex loader that splits the payload into small chunks spread across the code sections of the loader PE. Each chunk is immediately proceeded by a small function that copies its attached chunk to a buffer. The functions are called in a chain, and after a function has copied its chunk to the buffer it calls the next function in the sequence which then copies the next part of the payload. Execution jumps from one function to the next, until the entire payload has been constructed. The payload is then decrypted using a XOR- and ROR-based algorithm, and then decompressed using LZO1C. The Quicksand samples we analysed do not use the secondary shellcode loader, and instead include the PE loading code within the main loader. However, similarities were observed between the PE loading code found in Quicksand, and that of the shellcode loaders used by Quartz and Quixotic.

These three loaders were initially used almost exclusively with Qakbot, however, as the relationship between the former ITG23 members and Qakbot strengthened, we have observed them with additional malware payloads, in particular those related to the BlackBasta faction. For example, Quixotic was used to crypt a BlackBasta ransomware sample as recently as May 2023 and a CobaltStrike sample that was used in a BlackBasta attack in October 2022. Quicksand has also been used heavily with BlackBasta in 2023, and we have also observed it in use with SystemBC during attacks attributed to a team deploying BlackBasta.

In addition to these, we also track ITG23-related use of the CryptOne crypter, which has been used frequently with Qakbot. This is a Delphi-based crypter, which dates back to 2015, and is reportedly offered as a Crypter-As-A-Service. CryptOne has been used with a wide range of different malware families over the years including Gozi, Dridex, NetWalker, and WastedLocker. In 2019 it was observed with Emotet, and over the past couple of years it has been used heavily by Qakbot. CryptOne has also been recently used by former ITG23 actors to crypt malware other than Qakbot itself, including NetSupport used in an attack last fall as well as a Vidar infostealer from March 2023.

Example hashes

Date	Crypter	Payload	Hash
May 2023	Quixotic	Blackbasta Ransomware	462bbb8fd7be98129aa73efa91e2d88fa9cafc7b47431b8227d1957f5d0c8ba7
May 2023	Quicksand	Blackbasta Ransomware	723d1cf3d74fb3ce95a77ed9dff257a78c8af8e67a82963230dd073781074224
May 2023	CryptOne	Qakbot	380b8a70cef9604929177aa519ab7f02658648bde02892aa107e123764df8d54
March 2023	Quicksand	SystemBC	6f78256f20eb2b5594391095a341f8749395e7566fdd2ddd3a34a0db9bb9f871
March 2023	CryptOne	Vidar	bcda636524c83f25b8fe1508b9940c75af30a830c112b542600159991d56922c
March 2023	Quicksand	Qakbot	41da9355b1137296861187c51515f019cb358ce493136c54a60d1c1d8bf98ed9
December 2022	Quixotic	Qakbot	74784b81cf6d2c06903930a9a6d064d94cc1854e842dab5d45805f42bd766d98
September 2022	Quixotic	CobaltStrike	1751c378e2b14bd6238c3189e13501d191c117fdfe65e4e0ea1cb5829cce2bb9
October 2022	CryptOne	NetSupport	6767447d4a1ff6f14489fb7b316465c070d566c14e909c598ac94ee6b1dc6f76
June 2022	Quartz	Qakbot	80b90884283a2fec755d89591baf45cedf263dc46bae2996ccfc1ca99019f1d9

Scroll to view full table

In addition to those linked to Qakbot, we have also been tracking several additional loaders which are tied more closely to the BlackBasta faction. One of these is SharpDepositorCrypter, which was the primary loader used by the BlackBasta ransomware throughout most of 2022.

SharpDepositorCrypter appears to have originated as a loader for a .NET infostealer named SharpDepositor, which may explain the name found in PDB strings of early samples; however, it has evolved into a more complex loader and is no longer restricted to .NET payloads. The original version which loaded the .NET payload appears to have been based off a crypter named BrutoCrypt. It loaded a payload from the resources and decrypted it using AES.

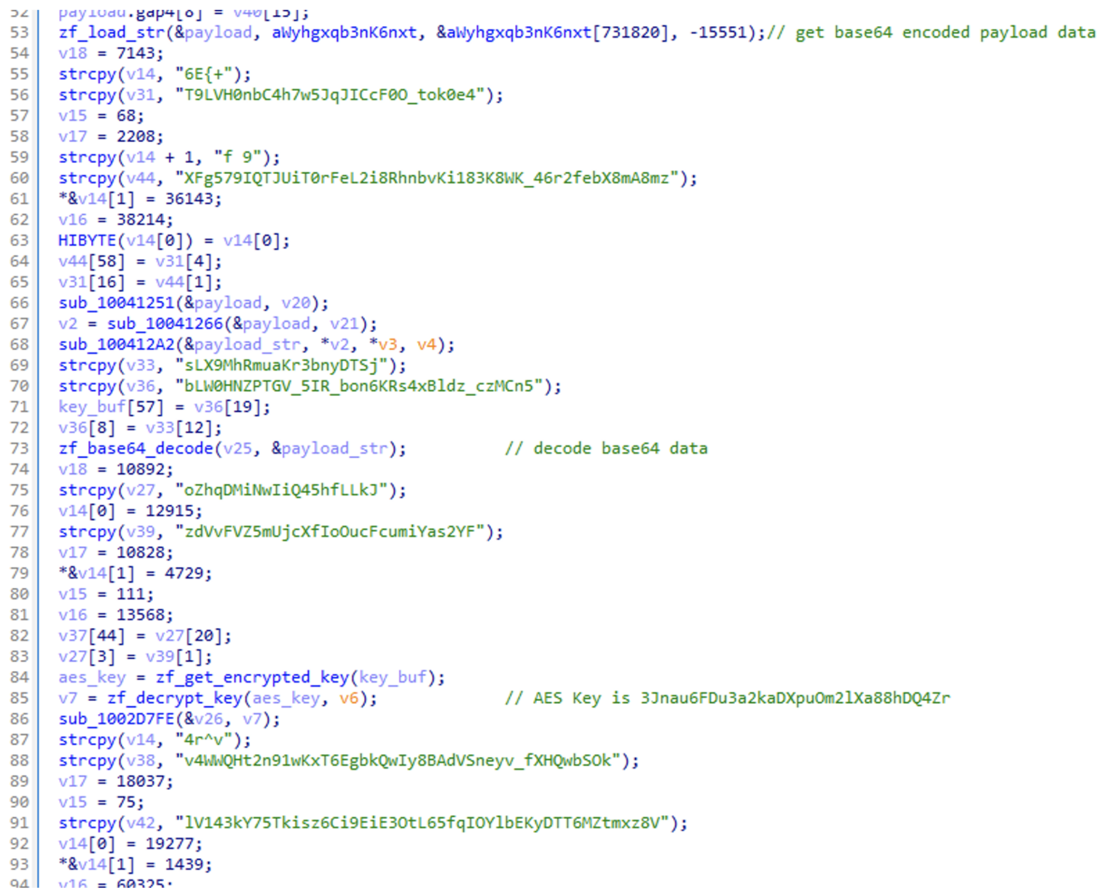
In the generic version of SharpDepositorCrypter, the payload was initially RC4 encrypted and base64 encoded; however, newer versions have switched to using AES encryption for the payload. A new instance of the loader process is created in suspended mode and the payload injected into it via process hollowing. The thread context in the target process is updated to execute the entrypoint of the payload, and the thread resumed.

The loader also includes a large quantity of anti-debugging and anti-analysis functions, many of which can be traced back to various open-source projects. In addition, newer variants also obfuscate API calls, encrypt strings, and contain a large amount of junk code in an effort to hide the malicious code and slow down analysis efforts.

SharpDepositorCrypter may also be known as OMCLoader. Newer samples have been identified that use this name, and early samples have also been found which include the string ‘OMC_BC’ in the command line during the process creation/payload injection stage. We are currently tracking the earlier RC4-based samples as SharpDepositorCrypter, and the AES-based versions under the name OMCLoader.

In addition to BlackBasta ransomware, these loaders have also been observed with CobaltStrike, exploit binaries used for privilege escalation, and rarely Gozi.

The use of SharpDepositorCrypter (SDC)/OMCLoader appears to have decreased during 2023, and the BlackBasta ransomware has been seen increasingly using other crypters including Quixotic, Quicksand, Dave and Tron.



```
payload.gap4[8] = v40[15];
zf_load_str(&payload, aWyhgxqb3nK6nxt, &aWyhgxqb3nK6nxt[731820], -15551);// get base64 encoded payload data
v18 = 7143;
strcpy(v14, "6E{+");
strcpy(v31, "T9LVH0nbC4h7w5JqJICcF0O_tok0e4");
v15 = 68;
v17 = 2208;
strcpy(v14 + 1, "f 9");
strcpy(v44, "XFg579IQTJUiT0rFeL2i8RhnbvKi183K8WK_46r2febX8mA8mz");
*&v14[1] = 36143;
v16 = 38214;
HIBYTE(v14[0]) = v14[0];
v44[58] = v31[4];
v31[16] = v44[1];
sub_10041251(&payload, v20);
v2 = sub_10041266(&payload, v21);
sub_100412A2(&payload_str, *v2, *v3, v4);
strcpy(v33, "sLX9MhRmuaKr3bnyDTSj");
strcpy(v36, "bLW0HNZPTGV_5IR_bon6KRs4xBldz_czMCn5");
key_buf[57] = v36[19];
v36[8] = v33[12];
zf_base64_decode(v25, &payload_str);          // decode base64 data
v18 = 10892;
strcpy(v27, "oZhqDMiNwIiQ45hfLLkJ");
v14[0] = 12915;
strcpy(v39, "zdVvFVZ5mUjcXfIoOucFcumiYas2YF");
v17 = 10828;
*&v14[1] = 4729;
v15 = 111;
v16 = 13568;
v37[44] = v27[20];
v27[3] = v39[1];
aes_key = zf_get_encrypted_key(key_buf);
v7 = zf_decrypt_key(aes_key, v6);             // AES Key is 3Jnau6FDu3a2kaDXpuOm2lXa88hDQ4Zr
sub_1002D7FE(&v26, v7);
strcpy(v14, "4r^v");
strcpy(v38, "v4WWQHt2n91wKxT6EgbkQwIy8BAdVSneyv_fXHQwbSOk");
v17 = 18037;
v15 = 75;
strcpy(v42, "lV143kY75Tkisz6Ci9EiE3OtL65fqIOYlbEKyDTT6MZtmxz8V");
v14[0] = 19277;
*&v14[1] = 1439;
v16 = 60325;
```

Fig. 3 — OMCLoader code which base64 decodes and AES decrypts a payload. This sample contains a large amount of junk code.

In February 2023 a new malware named PikaBot emerged, which we noted had code overlap with the later variants of OMCLoader. Both OMCLoader and the PikaBot loader have a similar structure and also share code similarities with regard to some of the anti-debug checks and process injection code. Like OMCLoader, the Pikabot Loader also uses AES to decrypt its payload, and the AES key used by PikaBot has a similar format to that used by OMCLoader. It is possible that there is a link between the developers of SDC/OMCLoader and PikaBot, and that the decline in SDC/OMCLoader activity in 2023 could be due to efforts being instead focused on PikaBot.

Finally, a number of CobaltStrike loaders have been identified in use by a sub-team we have associated with the BlackBasta faction. These are generally based on either the Cobalt Strike Artifact Kit loader templates or third-party User Defined Reflective Loaders (UDRL) and also borrow heavily from other open-source projects.

Several of these we track under the name Relic Loader. Whilst much of the core functionality of Relic matches that of standard ArtifactKit loader binaries, Relic also incorporates additional techniques to evade detection, such as function hooking and making API calls via direct syscalls. Three variants of Relic Loader have been identified so far. The simplest variant decrypts its payload using XOR and a 4 byte key, as per the standard ArtifactKit template, and executes it using a direct system call to the API ZwCreateThreadEx. The other two variants expand on this functionality by including additional hooking and anti-detection code.

Relic Loader borrows code from several publicly available projects. It is built using the GitHub project NotMedic/ArtifactKit, which is a simple project which allows ArtifactKit to be used from VisualStudio. The loader also uses the MinHook library for its hooking functions and also uses SysWhispers2 as the basis for its direct system calls.

In May 2023, we identified the same faction using an additional loader which we are tracking under the name Crius Loader. Crius is based heavily on a derivative of the TitanLdr CobaltStrike UDRL, possibly the TitanLdr-ng project. It decrypts its payload using RC4 and injects it into msiexec.exe.

Example hashes

Crypter	Payload	Hash
SharpDepositorCrypter	BlackBasta	3eb22320da23748f76f2ce56f6f627e4255bc81d09ffb3a011ab067924d8013b
OMCLoader	BlackBasta	9f948af3a30f125dcd24d8a628b3a18c66b3d72baede8496ee735cbdfd9cf0c7
Relic	CobaltStrike	d662f84331fad63aa8a15638278506e240977b64c062c31424760a423832bbe2
Relic	CobaltStrike	62de3d3bf07c3574fe6c9497526a7c097e2d4384d345f18c8b07edd11716333a
Relic	CobaltStrike	90474155b212711890e4f99cdaeab013cbfdb6936a61c75c8a5c90cbbe813c2f
Crius	CobaltStrike	8c5d48dfdb6c4ccaf83632883409198faa0dde2d71b3325bb79128b66a8d82e3

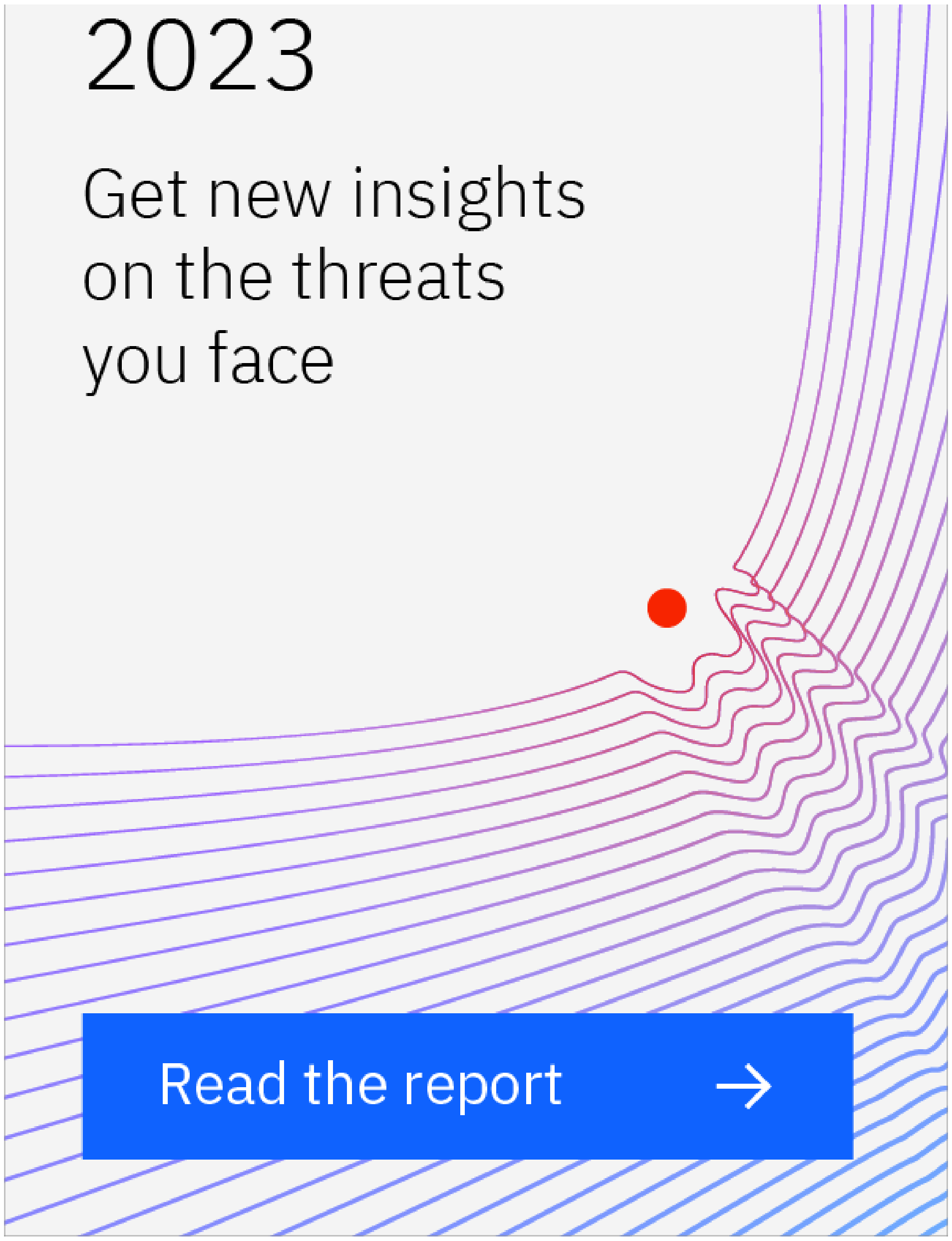
Scroll to view full table

The more things change…

About a year ago, ITG23 ceased using its highly successful Conti ransomware strain after which its members and affiliates created or joined new ransomware operations such as Quantum, Royal, Zeon, and BlackBasta. Through the next year, these actors and their partners continued to use many of the same crypters initially developed by ITG23, providing evidence of their continued close relationships.

Other intelligence paints a similar picture. The new ransomware operations rely on much of the same malware for initial access (IcedID, Emotet, Bumblebee, Qakbot, Gozi) distributed by many of the same initial access brokers (TA544, TA570, TA577, TA551, TA579, TA580). Other researchers have also demonstrated how Bitcoin wallets tied to Royal, Quantum, and Karakurt trace back to former ITG23 head Stern.

To be sure, there are differences in the current landscape from the original ITG23; its descendants have tested and adopted new malware strains — such as SVCReady, CargoBay, and Minodo — and forged relationships with new actors, e.g. FIN7 and DEV-0569. That said, the similarities with ITG23's activities prior to Conti's sunset indicate many of the same actors behind these new operations continue to collaborate closely behind the scenes.


2023
Get new insights on the threats you face
Read the report →