#ShortAndMalicious: StrelaStealer aims for mail credentials

medium.com/@DCSO_CyTec/shortandmalicious-strelastealer-aims-for-mail-credentials-a4c3e78c8abc

DCSO CyTec Blog November 21, 2022

DCSO CyTec Blog

Nov 8

.

4 min read

Strela surface-to-air missile launcher (Source: Wikipedia)
In our newest category **#ShortAndMalicious** *DCSO CyTec* aims to briefly highlight new and interesting samples we come across in our daily hunt for malware.

For the first entry in the series, we take a brief look at an undocumented custom malware we have been analysing under the moniker “StrelaStealer” (“Стрела” == arrow) which appears to be purpose-built to steal mail login data.

PDB path contained in StrelaStealer samples
DCSO CyTec first discovered StrelaStealer early November 2022 distributed via ISO files with what appears to be Spanish targets based on used lure documents. It is unclear at this point in time if StrelaStealer is part of a targeted attack.

Blog authored by and .

Execution via polyglot

StrelaStealer samples are distributed in ISO files with varying content. In one instance, StrelaStealer uses a renamed msinfo32.exe to sideload StrelaStealer as slc.dll. Another, more interesting variant distributes StrelaStealer as a DLL/HTML polyglot.

Polyglots files are files that are valid as two or more different file formats. In this case, StrelaStealer uses a file that is both valid as a DLL as well as an HTML page.

Execution of StrelaStealer via polyglot
The ISO file contains two files, one `Factura.lnk` and the polyglot `x.html` file. The LNK file then executes `x.html` twice, once as a DLL and a second time as an HTML file.

Parsed LNK file — command to execute the polyglot
Inspecting `x.html` then shows that it simply contains HTML code appended to the DLL file:

Appended HTML code
Double-clicking it opens the browser and displays the lure document:

Lure document rendered by Firefox

Malware analysis

StrelaStealer samples are DLL files, with the main functionality triggered by calling its main export function named `Strela` or `s`. While its code is not obfuscated, strings are encrypted with a cyclic xor with a hardcoded key:

Hardcoded xor key
Once executed, StrelaStealer attempts to locate and steal mail login data from Thunderbird and Outlook.

For Thunderbird, StrelaStealer searches for `logins.json` and `key4.db` in the `%APPDATA%\Thunderbird\Profiles\` directory and sends the file contents to its C2.

For Outlook, StrelaStealer enumerates the registry key `HKCU\SOFTWARE\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\` in order to find the values `IMAP User`, `IMAP Server` and `IMAP Password`. StrelaStealer then decrypts the `IMAP Password` using `CryptUnprotectData` before sending the triple to its C2.

Communication

Communication is done using plain HTTP POSTs, with the payload encrypted using the same xor key as for the strings. C2 server and resource name are hardcoded and so far all samples were configured for the same one:

```
hxxp://193.106.191[.]166/server.php
```

The IP address is hosted on known Russian bulletproof hosting “Kanzas LLC” with the /24 likely being hosted in Moscow.

Stolen files from Thunderbird are sent home in the following format:

```
[prefix "FF"][DWORD size logins.json][contents of logins.json][contents of key4.db]
```

Outlook data uses the following format:

```
[prefix "OL"][Server1,User1,Password1][Server2,User2,Password2]...
```

When sending home data, StrelaStealer checks for the last two bytes of the response to be `KH` which appears to signal a successful transfer and causes StrelaStealer to quit, otherwise it retries to send the data again after a 1 second sleep.

IoCs

As usual, you can find below the IoCs. We share a MISP event on our GitHub.

```
fa1295c746e268a3520485e94d1cecc77e98655a6f85d42879a3aeb401e5cf15c8eb6efc2cd0bd10d9fdd4f644ebbebdebaff376ece
```

MITRE ATT&CK

```
T1003     - Credential Dumping T1041     - Exfiltration Over C2 ChannelT1041     - Exfiltration Over
Command and Control ChannelT1059.003 - Windows Command ShellT1071     - Standard Application Layer
ProtocolT1566.001 - Spearphishing AttachmentT1574.002 - DLL Side-Loading
```