StrelaStealer, Software S1183 | MITRE ATT&CK®

Archived: 2026-04-05 15:03:38 UTC

Enterprise T1071 .001 Application Layer Protocol: Web Protocols

StrelaStealer communicates externally via HTTP POST with encrypted content.[1][3][4]

Enterprise T1119 Automated Collection

StrelaStealer attempts to identify and collect mail login data from Thunderbird and Outlook following execution.[1][2][3][4]

Enterprise T1020 Automated Exfiltration

StrelaStealer automatically sends gathered email credentials following collection to command and control servers via HTTP POST.[1][4]

Enterprise T1059 .001 Command and Scripting Interpreter: PowerShell

StrelaStealer variants have used PowerShell scripts to download or drop payloads, including obfuscated variants to connect to a WebDAV server to download and executed an encrypted DLL for installation.[4]

.003 Command and Scripting Interpreter: Windows Command Shell

StrelaStealer has included BAT files in some instances for installation.[3][4]

.007 Command and Scripting Interpreter: JavaScript

StrelaStealer has been distributed as a malicious JavaScript object.[2][3][4]

Enterprise T1132 .001 Data Encoding: Standard Encoding

StrelaStealer utilizes a hard-coded XOR key to encrypt the content of HTTP POST requests to command and control infrastructure.[4]

Enterprise T1001 Data Obfuscation

StrelaStealer encrypts the payload of HTTP POST communications using the same XOR key used for the malware's DLL payload.[1]

Enterprise T1622 Debugger Evasion

StrelaStealer variants include functionality to identify and evade debuggers.[3]

Enterprise T1140 Deobfuscate/Decode Files or Information

StrelaStealer payloads have included strings encrypted via XOR.[1] StrelaStealer JavaScript payloads utilize Base64-encoded payloads that are decoded via certutil to create a malicious DLL file.[2][3]

Enterprise T1480 Execution Guardrails

StrelaStealer variants only execute if the keyboard layout or language matches a set list of variables.[3][4]

.002 Mutual Exclusion

StrelaStealer variants include the use of mutex values based on the victim system name to prevent reinfection.[3]

Enterprise T1041 Exfiltration Over C2 Channel

StrelaStealer exfiltrates collected email credentials via HTTP POST to command and control servers.[1][2][3][4]

Enterprise T1574 .001 Hijack Execution Flow: DLL

StrelaStealer has sideloaded a DLL payload using a renamed, legitimate `msinfo32.exe` executable.[1]

Enterprise T1105 Ingress Tool Transfer

StrelaStealer installers have used obfuscated PowerShell scripts to retrieve follow-on payloads from WebDAV servers.[4]

Enterprise T1036 Masquerading

StrelaStealer PE executable payloads have used uncommon but legitimate extensions such as `.com` instead of `.exe`.[4]

.003 Rename Legitimate Utilities

StrelaStealer has used a renamed, legitimate `msinfo32.exe` executable to sideload the StrelaStealer payload during initial installation.[1]

.005 Match Legitimate Resource Name or Location

StrelaStealer payloads have tailored filenames to include names identical to the name of the targeted organization or company.[4]

.008 Masquerade File Type

StrelaStealer has been distributed as a DLL/HTML polyglot file.[1][4]

Enterprise T1027 Obfuscated Files or Information

StrelaStealer has been distributed in ISO archives.[1] StrelaStealer has been delivered in encrypted, password-protected ZIP archives.[4]

.002 Software Packing

StrelaStealer variants have used packers to obfuscate payloads and make analysis more difficult.[2]

.013 Encrypted/Encoded File

StrelaStealer uses XOR-encoded strings to obfuscate items.[1]

.015 Compression

StrelaStealer has been delivered via JScript files in a ZIP archive.[2][3]

.016 Junk Code Insertion

StrelaStealer variants have included excessive mathematical functions padding the binary and slowing execution for anti-analysis and sandbox evasion purposes.[3]

Enterprise T1566 .001 Phishing: Spearphishing Attachment

StrelaStealer has been distributed as a spearphishing attachment.[1]

Enterprise T1518 Software Discovery

StrelaStealer variants use COM objects to enumerate installed applications from the "AppsFolder" on victim machines.[4]

Enterprise T1553 .002 Subvert Trust Controls: Code Signing

StrelaStealer variants have used valid code signing certificates.[4]

Enterprise T1218 .011 System Binary Proxy Execution: Rundll32

StrelaStealer DLL payloads have been executed via `rundll32.exe`.[2][4]

Enterprise T1082 System Information Discovery

StrelaStealer variants collect victim system information for exfiltration.[4]

Enterprise T1614 .001 System Location Discovery: System Language Discovery

StrelaStealer variants check system language settings via keyboard layout or similar mechanisms.[3][4]

Enterprise T1552 .001 Unsecured Credentials: Credentials In Files

StrelaStealer searches for and if found collects the contents of files such as `logins.json` and `key4.db` in the `$APPDATA%\Thunderbird\Profiles\` directory, associated with the Thunderbird email application.[1][3]

.002 Unsecured Credentials: Credentials in Registry

StrelaStealer enumerates the registry key `HKCU\SOFTWARE\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\` to

identify the values for "IMAP User," "IMAP Server," and "IMAP Password" associated with the Outlook email application.[1][3][4]

Enterprise T1204 .002 User Execution: Malicious File

StrelaStealer relies on user execution of a malicious file for installation.[1]

Enterprise T1497 Virtualization/Sandbox Evasion

StrelaStealer payloads have used control flow obfuscation techniques such as excessively long code blocks of mathematical instructions to defeat sandboxing and related analysis methods.[2][3]
