LightSpy, Software S1185 | MITRE ATT&CK®

Archived: 2026-04-05 14:38:12 UTC

Enterprise T1071 .001 Application Layer Protocol: Web Protocols

LightSpy's C2 communication is performed over WebSockets using the open source library SocketRocket with functionality such as, heartbeat, receiving commands, and updating command status.[2]

Enterprise T1123 Audio Capture

LightSpy uses Apple's built-in AVFoundation Framework library to capture and manage audio recordings then transform them to JSON blobs for exfiltration.[2]

Enterprise T1217 Browser Information Discovery

To collect data on the host's Wi-Fi connection history, LightSpy reads the `/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist` file. It also utilizes Apple's `CWWiFiClient` API to scan for nearby Wi-Fi networks and obtain data on the SSID, security type, and RSSI (signal strength) values.[2]

Enterprise T1555 .001 Credentials from Password Stores: Keychain

LightSpy performs an in-memory keychain query via `SecItemCopyMatching()` then formats the retrieved data as a JSON blob for exfiltration.[2]

Enterprise T1480 Execution Guardrails

On macOS, LightSpy checks the existence of a process identification number (PID) file, `/Users/Shared/irc.pid`, to verify if LightSpy is currently running.[2]

Enterprise T1041 Exfiltration Over C2 Channel

To exfiltrate data, LightSpy configures each module to send an obfuscated JSON blob to hardcoded URL endpoints or paths aligned to the module name.[2]

Enterprise T1083 File and Directory Discovery

LightSpy uses the `NSFileManager` to move, create and delete files. LightSpy can also use the assembly `bt` instruction to determine a file's executable permissions.[2]

Enterprise T1105 Ingress Tool Transfer

On macOS, LightSpy downloads a `.json` file from the C2 server. The `.json` file contains metadata about the plugins to be downloaded, including their URL, name, version, and MD5 hash. LightSpy retrieves the plugins

specified in the `.json` file, which are compiled `.dylib` files. These `.dylib` files provide task and platform specific functionality. LightSpy also imports open-source libraries to manage socket connections.[2]

Enterprise T1046 Network Service Discovery

To collect data on the host's Wi-Fi connection history, LightSpy reads the `/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist file`.It also utilizes Apple's CWWiFiClient API to scan for nearby Wi-Fi networks and obtain data on the SSID, security type, and RSSI (signal strength) values.[2]

Enterprise T1027 .001 Obfuscated Files or Information: Binary Padding

LightSpy's configuration file is appended to the end of the binary. For example, the last `0x1d0` bytes of one sample is an AES encrypted configuration file with a static key of `3e2717e8b3873b29`.[2]

.013 Obfuscated Files or Information: Encrypted/Encoded File

LightSpy encrypts the C2 configuration file using AES with a static key, while the module `.dylib` files use a rolling one-byte encoding for obfuscation.[2]

Enterprise T1057 Process Discovery

If sent the command `16002`, LightSpy uses the `NSWorkspace runningApplications()` method to collect the process ID, path to the executable, bundle information, and the filename of the executable for all running applications.[2]

Enterprise T1113 Screen Capture

LightSpy uses Apple's built-in AVFoundation Framework library to access the user's camera and screen. It uses the `AVCaptureStillImage` to take a picture using the user's camera and the `AVCaptureScreen` to take a screenshot or record the user's screen for a specified period of time.[2]

Enterprise T1129 Shared Modules

LightSpy's main executable and module `.dylib` binaries are loaded using a combination of `dlopen()` to load the library, `_objc_getClass()` to retrieve the class definition, and `_objec_msgSend()` to invoke/execute the specified method in the loaded class.[2]

Enterprise T1518 Software Discovery

If sent the command `16001`, LightSpy uses the `NSFileManger contentsOfDirectoryAtPath()` to enumerate the Applications folder to collect the bundle name, bundle identifier, and version information from each application's `info.plist` file. The results are then converted into a JSON blob for exfiltration.[2]

Enterprise T1082 System Information Discovery

LightSpy's second stage implant uses the `DeviceInformation` class to collect system information, including CPU usage, battery statistics, memory allocations, screen size, etc.[2]

Mobile T1437 .001 Application Layer Protocol: Web Protocols

LightSpy has used both HTTPS and Websockets to communicate with the C2.[3][4][5]

Mobile T1532 Archive Collected Data

LightSpy collects and compresses data to be exfiltrated using SSZipArchive.[5][4]

Mobile T1429 Audio Capture

LightSpy has captured environment audio, phone calls and Voice over IP (VoIP) calls.[6][1][3][4][5]

Mobile T1398 Boot or Logon Initialization Scripts

LightSpy has established auto-start execution during the system boot process.[4]

Mobile T1623 Command and Scripting Interpreter

LightSpy has plugins for executing shell commands either from the C2 server or a library file called `zt.dylib`.[1][4][5]

Mobile T1634 .001 Credentials from Password Store: Keychain

LightSpy has accessed the device's KeyChain data.[1][4][7][5]

Mobile T1662 Data Destruction

LightSpy has deleted media files and messenger-related files on the device.[4] Additionally, LightSpy has used the AppDelete plugin to remove multiple messaging applications, such as WeChat, QQ, Telegram, Line and Whatsapp.[5]

Mobile T1533 Data from Local System

LightSpy has collected and exfiltrated files from messaging applications, such as Telegram, QQ, WeChat, and Whatsapp, and browser history from Chrome and Safari.[1][3][4][7][5]

Mobile T1456 Drive-By Compromise

LightSpy gains initial execution when a victim visits a compromised or adversary-controlled website, including those mimicking legitimate sources such as a Hong Kong newspaper. Upon loading `index.html`, a Safari WebKit exploit is triggered, leading to the download of a Mach-O binary disguised with a `.png` extension.[6][7][5][4]

Mobile T1642 Endpoint Denial of Service

LightSpy has used the DeleteSpring plugin to render the device's user interface inoperable by disabling SpringBoard, which is iOS's home screen manager.[5] LightSpy has used the BootDestroy plugin to prevent the victim device from booting by modifying the NVRAM parameter `auto-boot` to `false`.[5] Additionally, LightSpy has renamed the Wi-Fi daemon to disable wireless connectivity.[5]

Mobile T1646 Exfiltration Over C2 Channel

LightSpy has exfiltrated collected data to the C2.[5]

Mobile T1658 Exploitation for Client Execution

LightSpy has compromised iPhones running iOS 12.1 and 12.2 without any user interaction.[7]

Mobile T1404 Exploitation for Privilege Escalation

LightSpy uses the embedded `time_waste` function to bypass standard iOS API restrictions and enable unauthorized audio/video recording. This exploit injects a `.dylib` into the `SpringBoard` process, allowing persistent access to audio and video capture.[5][4]

Mobile T1544 Ingress Tool Transfer

LightSpy has retrieved files from the C2 server.[1][4] Examples of files from the C2 are `amfidebilitate` (jailbreak component), `jbexec` (executable to verify jailbreak), `bb` (FrameworkLoader), `cc` (launchctl binary for persistence), `b.plist` (configuration for auto-start), and `resources.zip`, which contains additional jailbreak-related components.[5]

Mobile T1430 Location Tracking

LightSpy has accessed the device's GPS location.[1][3][7][5]

Mobile T1655 Masquerading

LightSpy has masqueraded a Mach-O executable as a png file.[4][5]

Mobile T1575 Native API

LightSpy's main executable and modules use native libraries to execute targeted functionality.[3][1][5][4]

Mobile T1423 Network Service Scanning

LightSpy uses the `landevices` module to enumerate devices on the same WiFi network through active scanning.[4][5][7]

Mobile T1509 Non-Standard Port

LightSpy has communicated with the C2 using ports 52202, 51200, 43201, 43202, 43203, and 21202.[3]

Mobile T1406 Obfuscated Files or Information

Using an XOR-chain algorithm, LightSpy decrypts an embedded configuration blob containing URLs for jailbreak components and next-stage payloads. It also decrypts modules in memory and on disk using AES-ECB with the hardcoded key `3e2717e8b3873b29`.[3][1][4][5] Additionally, LightSpy's plugins have been encrypted during transmission.[5]

Mobile T1660 Phishing

LightSpy has delivered malicious links through Telegram channels and Instagram posts.[6][7]

Mobile T1424 Process Discovery

LightSpy has collected a list of running processes.[4][5]

Mobile T1631 Process Injection

LightSpy injects libcynject.dylib into the SpringBoard process to enable audio/video recording.[5]

Mobile T1636 .002 Protected User Data: Call Log

LightSpy has accessed the device's call log.[1][3][4][7][5]

.003 Protected User Data: Contact List

LightSpy has accessed the device's contact list.[1][3][4][7][5]

.004 Protected User Data: SMS Messages

LightSpy has accessed SMS messages.[1][3][4][5]

Mobile T1513 Screen Capture

LightSpy has a plugin that can take screenshots.[4][5]

Mobile T1582 SMS Control

LightSpy has sent and deleted SMS messages.[3][4][5]

Mobile T1418 Software Discovery

LightSpy has accessed a list of installed applications.[1][3][4][5]

Mobile T1409 Stored Application Data

LightSpy has collected payment history from WeChat Pay.[1][3][5]

Mobile T1426 System Information Discovery

LightSpy collects device information, including the phone number, IMEI, CPU details, screen specifications, and memory information.[5][4][3][1]

Mobile T1422 System Network Configuration Discovery

LightSpy has collected device information such as IMEI, phone number, MAC address and IP address.[5]

.002 Wi-Fi Discovery

LightSpy uses the WifiList (or `libWifiList`) plugin to gather Wi-Fi network information, such as the SSID, BSSID, signal strength (RSSI), channel, security type, and previously saved networks.[1][5][4][3]

Mobile T1421 System Network Connections Discovery

LightSpy has collected a list of cellular networks and connected Wi-Fi history using a LAN scanner based on MMLanScan.[6][1][3][4][7]

Mobile T1512 Video Capture

LightSpy has the ability to take one picture, continuous pictures or event-related pictures using the device's camera.[6][1][3][4][5] For iOS devices, the default file type for pictures is in High Efficiency Image Format (HEIC); for Android devices, the default file type for pictures is in JPEG format.
