BlueNoroff strikes again with new macOS malware

By Jamf Threat Labs

Home

- Why Jamf

  Why Jamf

  Meet Jamf: The most complete Apple device management and security solution.

  Learn More

- Products

  Products

  Empower your employees with our best-in-class products.

  Learn More

- Pricing

  Pricing

Find the best set of Jamf tools for your budget.

- Resources

  Resources

  Whether you're looking for education or inspiration, Jamf has you covered with the latest industry and product-specific resources.

  View Resources

- Partners

  Partners

  Current partner, future partner or purchasing from a partner - you're in the right place.

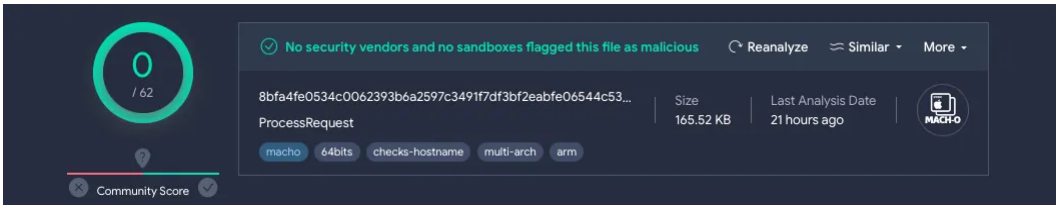
  Learn More

- Start Trial Contact

Research led by Ferdous Saljooki.

Background

Jamf Threat Labs has identified a new malware variant attributed to the BlueNoroff APT group. BlueNoroff's campaigns are financially motivated, frequently targeting cryptocurrency exchanges, venture capital firms and banks. During our routine threat hunting, we discovered a Mach-O universal binary communicating with a domain that Jamf has previously classified as malicious. This executable was undetected on VirusTotal at the time of our analysis, piquing our interest.



The standalone binary, labeled `ProcessRequest`, is ad-hoc signed and has been observed communicating with the domain `swissborg[.]blog`. This raised suspicions, especially since a legitimate cryptocurrency exchange exists operating under the domain swissborg.com, where they host a legitimate blog at the URL swissborg.com/blog. The malware splits the command and control (C2) URL into two separate strings that get concatenated together. This is likely an attempt to evade static-based detection.

The usage of this domain greatly aligns with the activity we've seen from BlueNoroff in what Jamf Threat Labs tracks as the Rustbucket campaign. In this campaign, the actor reaches out to a target claiming to be interested in partnering with or offering them something beneficial under the guise of an investor or head hunter. BlueNoroff often creates a domain that looks like it belongs to a legitimate crypto company in order to blend in with network activity.

The malicious domain `swissborg[.]blog` was registered on May 31, 2023, and resolves to the IP address `104.168.214[.]151`. Pivoting from this domain revealed several URLs used for the malware's communication. However, at the time of our analysis, the C2 server did not respond to any of these URLs and went offline shortly after our attempts to communicate.

The IP address `104.168.214[.]151` has been associated with malware previously used by this attacker.

We have observed submissions to VirusTotal from countries such as Japan and the US in September and October.

Analysis

The malware is written in Objective-C and operates as a very simple remote shell that executes shell commands sent from the attacker server. Although it is not entirely clear how initial access was achieved, this malware is likely being used as a later stage to manually run commands after compromising a system. This malware at a glance is very different from the previously mentioned RustBucket malware seen used in other attacks, but the attacker's focus in both cases seems to be providing simple remote shell capability.

Upon execution, the malware calls a function titled `sendRequest` to send a POST message to the hardcoded URL `hXXp://swissborg.blog/zxcv/bnm`. The malware then uses the Objective-C `NSProcessInfo` functionality which allows them to gain information about the malware process itself. It then retrieves the `operatingSystemVersionString` to determine the macOS version. An `NSMutableURLRequest` object is created using the hardcoded URL and the HTTP method and header fields are set accordingly.

This POST request uses the `NSURLSession` class to generate the user-agent in the following format.

- `AppName` : The name of the app derived from the `CFBundleName` key in the app's `Info.plist` . In the case where the executable is not run as part of an app bundle (which we suspect to be the case), this value gets set to the name of the executable.
- `AppVersion` : The version of the app obtained from the `CFBundleShortVersionString` key in the app's `Info.plist` . In the absence of app-specific details it would be set to `unknown version` .
- `CFNetworkVersion` : The version of the CFNetwork framework used by the app.
- `DarwinVersion` : The version of Darwin or XNU kernel.

The HTTP POST data is constructed using the following JSON formatted string, `{"sdf":"wsx","info":"operatingSystemVersionString"}` , where `operatingSystemVersionString` will be replaced by the property value fetched from the `processInfo` object.

Below is an example of the POST message being sent to the attacker server from the victim system.

The block callback `[ProcessRequest sendRequest]_block_invoke` serves as the command executor if a response is received from the C2.

The malware utilizes the `system()` function for command execution, inherently invoking `sh -c` . It logs the server response via `NSLog` for commands awaiting execution and records both successes and failures. The choice to log these activities is intriguing, as attackers crafting sophisticated malware typically omit any statements that might leave traces.

The main function of the program initializes an instance of the `ProcessRequest` class, then sets up a repeating timer using the `startTimer` method. This timer triggers the `sendRequest` method at regular intervals, facilitating periodic network requests. To ensure continuous operation, the `NSRunLoop` class is used, keeping the main thread active.

Conclusion

Although fairly simple, this malware is still very functional and will help attackers carry out their objectives. This seems to be a theme with the latest malware we've seen coming from this APT group. Based on previous attacks performed by BlueNoroff, we suspect that this malware was a late stage within a multi-stage malware delivered via social engineering. Jamf Threat Labs tracks this malware as ObjCShellz and as part of the RustBucket campaign.
