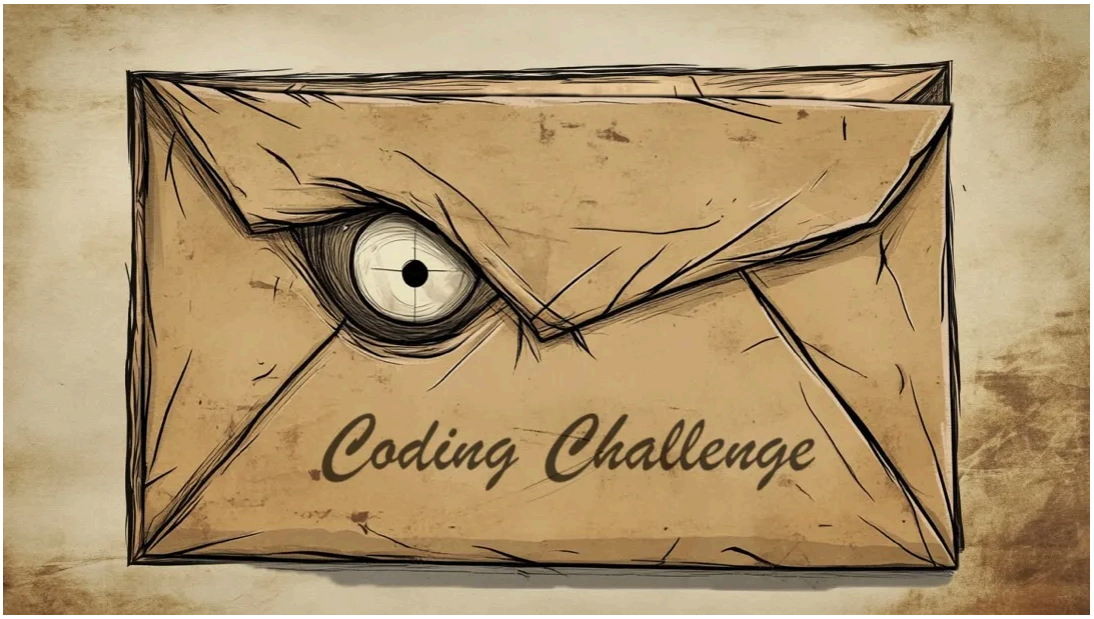
Jamf Threat Labs observes targeted attacks amid FBI Warnings

By Jamf Threat Labs

Archived: 2026-04-05 17:39:33 UTC

On September 3, 2024 the Federal Bureau of Investigations (FBI) released a public service announcement set to warn those in the Crypto Industry that the Democratic People's Republic of Korea ("DPRK" aka North Korea) has been targeting individuals by using clever social engineering techniques for the successful delivery of malware.



Authors: Jaron Bradley and Ferdous Saljooki

The DPRK has a long history of acquiring financial gains through creative and illicit means. Over the years, a significant portion of these financial gains has come from successful cyberattacks. As mentioned by the FBI’s public service announcement, specific individuals within crypto companies are being targeted.

As part of Jamf's ongoing research, Jamf Threat Labs had been proactively monitoring attacks that closely aligned with these warnings. Below, we provide detailed insights into the nature of these attacks in order to provide others with the knowledge needed to better identify and mitigate potential threats. The majority of attacks begin with an individual reaching out over a social media platform leading to the delivery of malware in some manner.

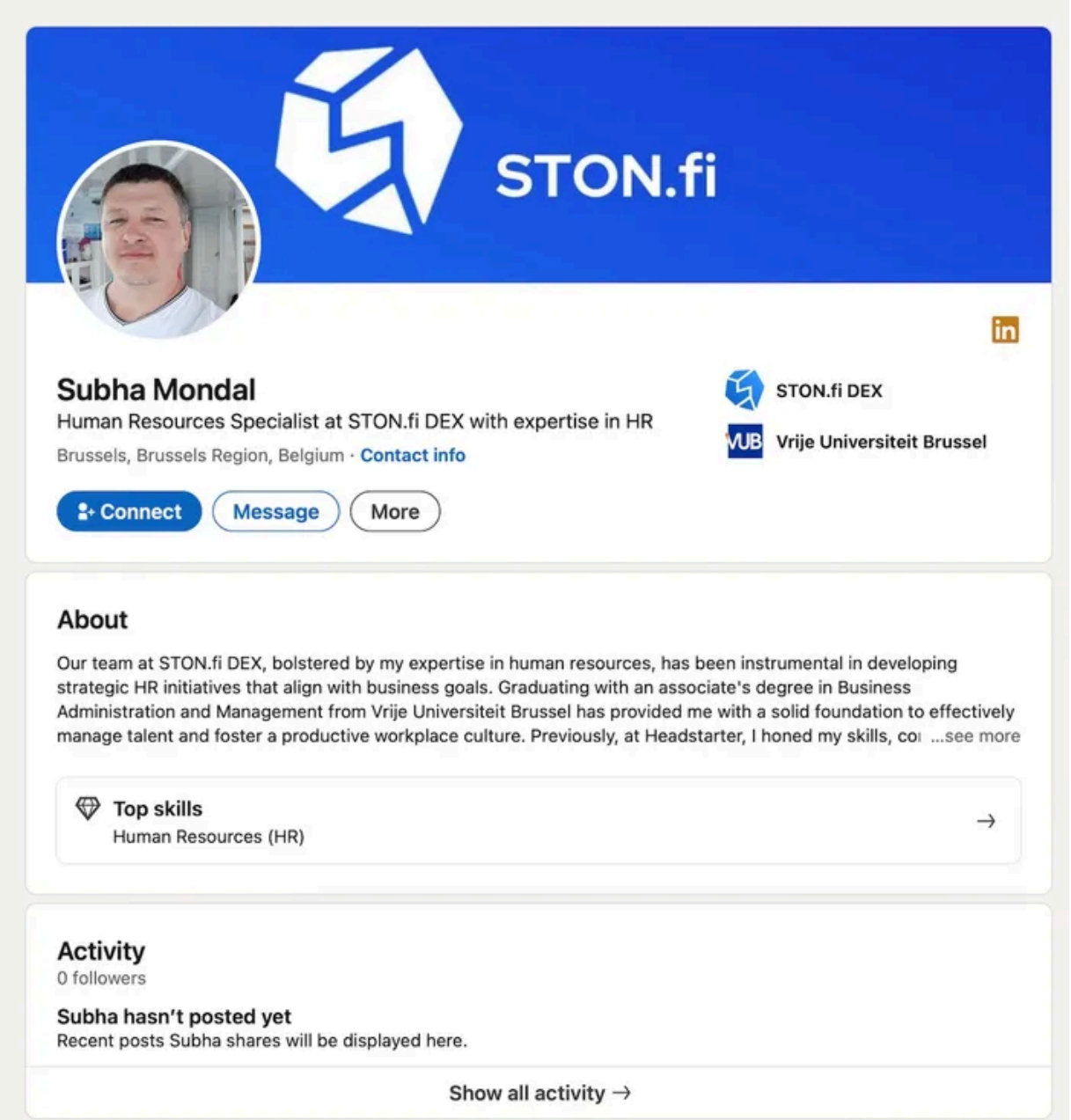
Social engineering

Humans have long been considered the weakest link in the cybersecurity chain, and attackers continue to exploit this vulnerability through increasingly sophisticated social engineering tactics. Social engineering schemes often target individuals through professional networking platforms, making users the first line of defense but also the most vulnerable.

Per the FBI announcement:

> *Before initiating contact, the actors scout prospective victims by reviewing social media activity, particularly on professional networking or employment-related platforms.*

Jamf Threat Labs noted an attack attempt in which a user was contacted on LinkedIn by an individual claiming to be a recruiter on the HR team at a tech company that specializes in decentralized finance.



LinkedIn profile impersonating an HR professional and used to contact potential victims

Note at the bottom of the image that this profile has 0 followers which can be a good indicator that this account was created recently. Much of this profile and the techniques used align with further documentation within the FBI announcement.

"*The actors may also impersonate recruiting firms or technology companies backed by professional websites designed to make the fake entities appear legitimate.*"

Although we are unfamiliar with the website ston.fi and can’t speak to its legitimacy, the recruiter claiming to work there is clearly meant to capture the target's interest.

Code execution attempts

The FBI announcement goes on to document a number of ways in which the fraud recruiter might convince a user to install malware. An attack scenario observed by Jamf Threat Labs was closest to that of bullet point two from the writeup.

- Requests to conduct a "pre-employment test" or debugging exercise that involves executing non-standard or unknown Node.js packages, PyPI packages, scripts, or GitHub repositories.

In the observed scenario, the recruiter sent a zipped coding challenge to the target (51a88646f9770e09b3505bd5cbadc587abb952ba), which is considered to be a fairly common step in the screening processes of a modern day development role. This coding challenge came in the form of a Visual Studio project that has the developer focus on converting Slack messages to CSV format in C#. However, buried within two separate csproj files are malicious bash commands that both download a second stage payload. The two csproj files can be seen at the following locations:

The following bash commands will execute upon building the project:

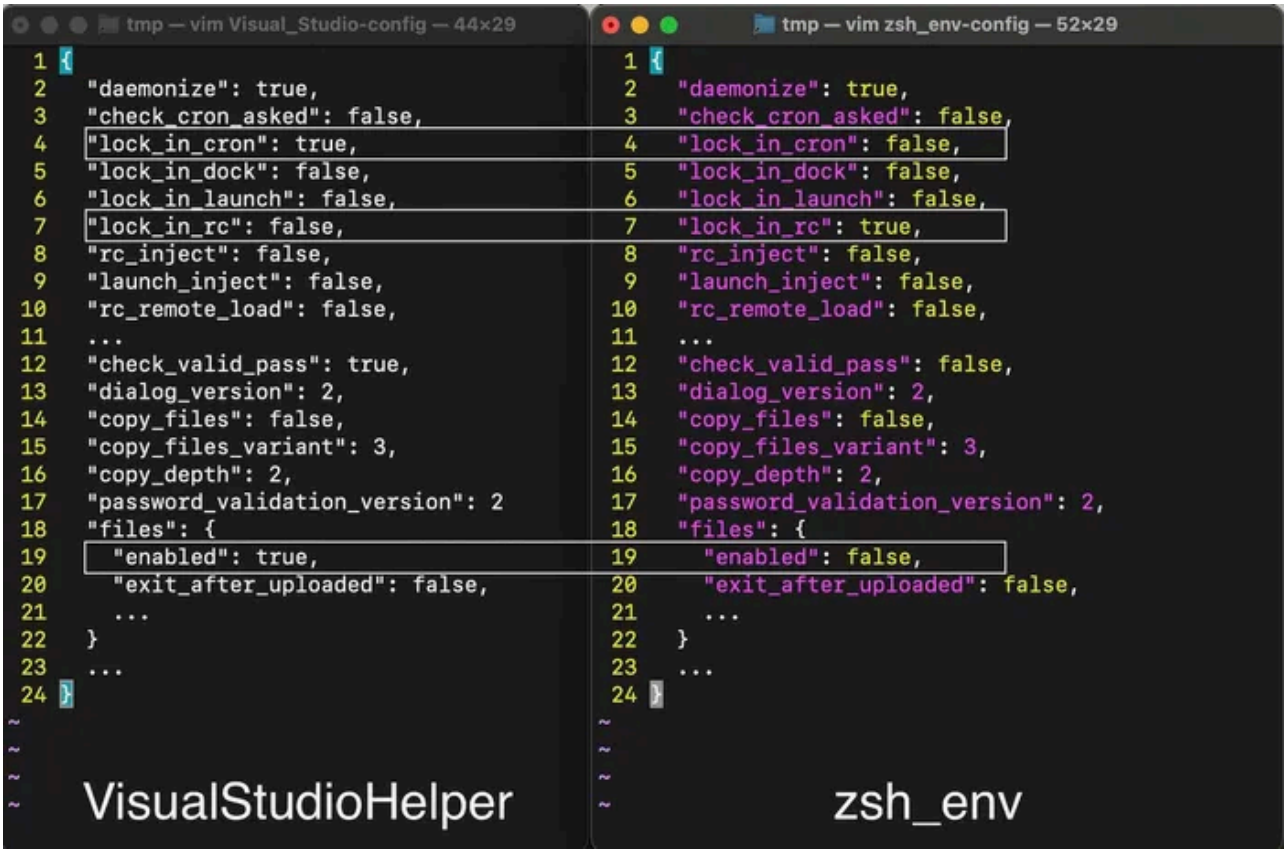
Both scripts change their root directory and then download a second stage payload via curl from taurihostmetrics[.]com.

Each payload is marked as executable and then hidden before being run. These two executables are both stage two malware. VisualStudioHelper communicates with wiresapplication[.]com while zsh_env communicates with juchesoviet48[.]com.

The stage two malware that is dropped by the coding challenge is tracked by Jamf Threat Labs under the name Thiefbucket but is known to some as “Rustdoor.” Jamf Threat Labs has always attributed this malware to the ongoing DPRK activity due to the stage one techniques and the manner in which they are delivered to their targets.

Stage two: comparison of configuration

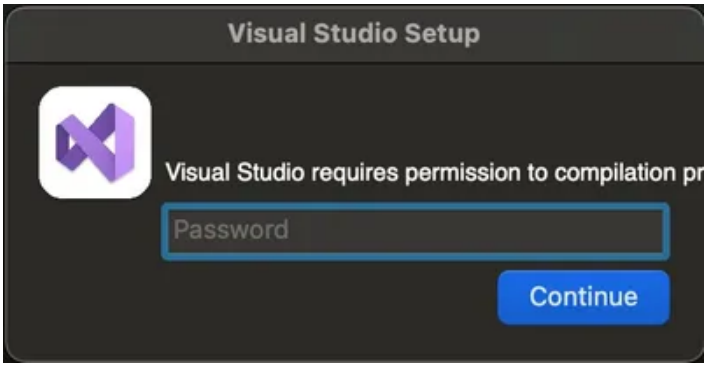
As mentioned in the above section, two executables were downloaded and executed by the fake coding challenge. These two executables are nearly identical in functionality. What primarily sets them apart is their embedded configurations.



The config files embedded within the two separate malware samples shows that the VisualStudioHelper will persist via cron while zsh_env will persist via the zshrc file.

Further down, line 19 shows that VisualStudioHelper has a configuration setting called “files” set to true. This setting will cause the malware to act as an infostealer by grabbing a number of different files specified further down in the config. In order to acquire some of the most valuable files, infostealers often require further permissions. These permissions are obtained by the malware via a popup window. This prompt is also defined within the config file.

The above excerpt shows a portion of the config within the VisualStudioHelper payload that will cause the malware to prompt the user for their password using a prompt window that is tailored to look as though it originated from Visual Studio. Given that this prompt is displayed at the same time the project is built, the user may be more likely to think nothing of it and enter their password.



The other stage two malware that is downloaded (zsh_env) simply sets up persistence via the .zshrc configuration. This ensures that any time the user opens a zsh shell moving forward, the malware will also be executed. This is a technique that likely ends up being reliable given the attacker knows they're targeting a developer who will likely use the Terminal, again causing the backdoor to be run in the background.

In summary: both payloads are highly similar. The difference between the two is:

1. VisualStudioHelper acts as an automated infostealer, can operate as a standard backdoor when invoked by cron and communicates with wiresapplication[.]com.
2. zsh_env operates as a backdoor, does not automate any of the infostealer functionality, persists via the zshrc config file, and uses a command and control server at juchesoviet48[.]com.

Stage two capabilities and updates

Since its original discovery, Thiefbucket has held the following capabilities:

- Automation of infostealer-like logic
- Download files
- Upload files
- Kill processes
- Delete files and directories
- Sleep
- Quickly search indexed files using Spotlight
- Ability to self delete
- Ability to run shell commands
- Ability to prompt the user with dialog boxes
- Ability to persist via LaunchAgent, cron, dock, and zshrc profiles.

The malware has a handful of differences from its first appearance, most notably the executable that was originally written in Rust seems to have been re-created in Objective-C.

Jamf Threat Labs continues to investigate the differences in features, but at a first glance they appear to be minor. The help page for the malware has been updated with a handful of new arguments. Most of these appear to be ways to run or test the embedded config features. We've marked the new available arguments below with asterisks.

It's worth noting that the VisualStudioHelper payload makes use of the `--dialog` argument when it prompts the user for their password.

When testing the `--bin` argument, we observed that Thiefbucket will add the supplied binary path to the zshrc file before deleting itself.

Conclusion

Threat actors continue to remain vigilant in finding new ways to pursue those in the crypto industry. Jamf Threat Labs has seen attacks in line with the FBI warning that went out this month. It's important to train your

employees, including your developers, to be hesitant to trust those who connect on social media and ask users to run software of any type. These social engineering schemes performed by the DPRK come from those who are well-versed in English and enter the conversation having well researched their target. We recommend reading the public service announcement for a list of mitigations and best practices.
