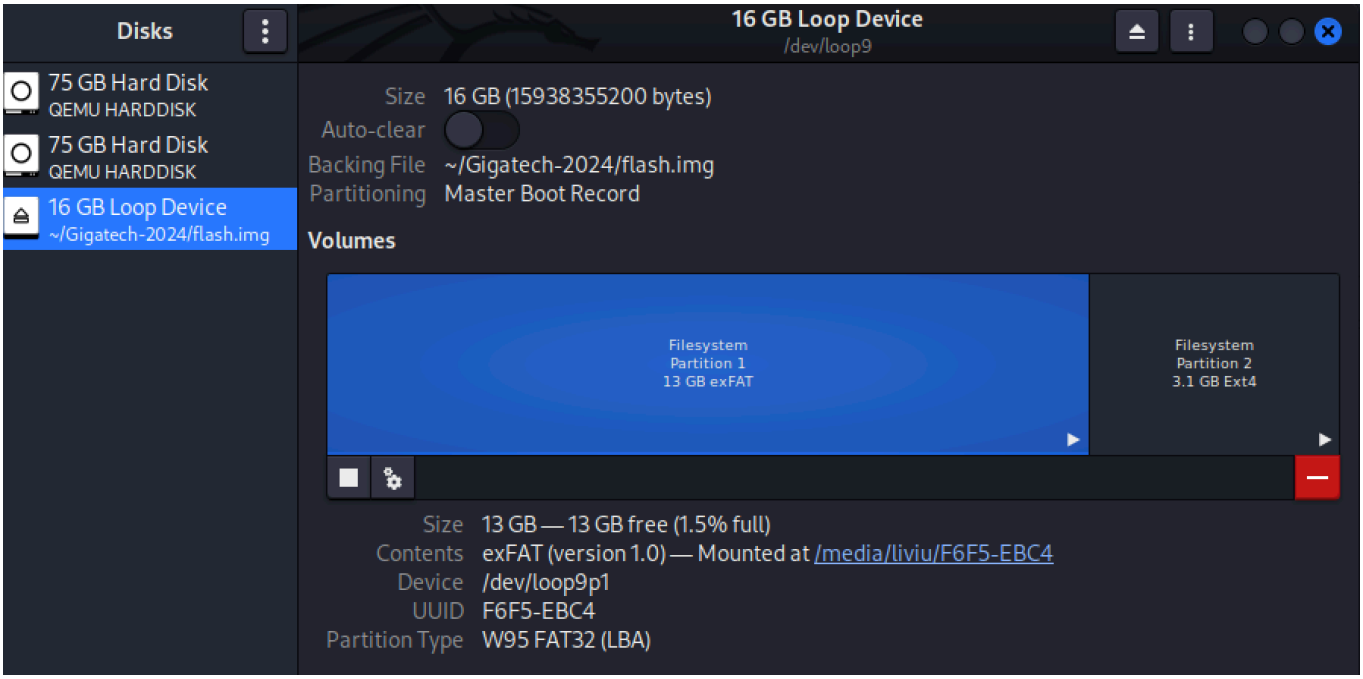
Team members:

Mereacre Liviu
Mocrenco Artiom
Brinzila Calin
Nicu Iurie
Gidilica Nichita

We were given a flash drive that had 2 partitions each with their filesystem:



First had a file with photos (Source) and a binary (Validator), the other partition had 2 folders (Keepass and Remote) with 2 ssh keys.

After decompiling the Validator binary we notice that it has a ROT13 function and a weird string that if we apply ROT we get "Lovelace" and the variable NQN would turn into "ADA"

```
// Validator.Program

using System;

using System.IO;

using System.Security.Cryptography;
```

```
internal class Program

{

private static void Main(string[] args)

{

if (1 > args.Length)

{

Console.WriteLine("You forgot something!");

Environment.Exit(42);

}

string pwd = EBG13(Environment.GetEnvironmentVariable(EBG13("NQN")));

//if (!string.IsNullOrEmpty(pwd) && EBG13(EBG13(pwd)) == "Ybirynpr")

//{

byte[] tmp = File.ReadAllBytes(args[0]);

using (SHA256 sha = SHA256.Create())

{

if (BitConverter.ToString(sha.ComputeHash(tmp)).Replace("-", "").ToLower()
== Path.GetFileNameWithoutExtension(args[0])!.ToLower())

{

Console.WriteLine("Valid");

}
```

```
else

{

Console.WriteLine("Invalid");

}

}

Environment.Exit(0);

return;

//}

//while (true)

//{

// Console.Write("Gotcha! You are dead!");

//}

}



private static string EBG13(string input)

{

if (string.IsNullOrEmpty(input))

{

return null;

}
```

```
char[] buffer = new char[input.Length];

for (int i = 0; i < input.Length; i++)

{

char c = input[i];

if (c >= 'a' && c <= 'z')

{

int k = c + 13;

if (k > 122)

{

k -= 26;

}

buffer[i] = (char)k;

}

else if (c >= 'A' && c <= 'Z')

{

int j = c + 13;

if (j > 90)

{

j -= 26;

}
```

```
buffer[i] = (char)j;

}

else

{

buffer[i] = c;

}

}

return new string(buffer);

}

}
```

Ada Lovelace(maybe a coincidence but very unlikely)

Ada Lovelace

107 limbi

Articol Discuție Lectură **Modificare** Modificare sursă Istoric Unelte

De la Wikipedia, enciclopedia liberă

Augusta Ada King, Contesă de Lovelace (10 decembrie 1815 - 27 noiembrie 1852), născută **Augusta Ada Byron**, iar acum cunoscută sub numele de **Ada Lovelace**, a fost o matematiciană engleză și o scriitoare cunoscută în principal pentru munca ei la calculatorul mecanic al lui Charles Babbage, motorul analitic. Consemnările ei privind motorul includ ceea ce este recunoscut ca fiind primul algoritm care urmează să fie procesat de către o mașină. Din acest motiv, ea este adesea considerată primul programator de calculator din lume.[5][6][7]

Ada Lovelace

Ada King, Contesă de Lovelace, 1840

Biografie [modificare | modificare sursă]

Augusta Ada Byron s-a născut la 10 decembrie 1815, fiind singurul copil legitim al poetului Lord Byron. Mama ei a fost Anne Isabella (Annabella) Milbanke. Toți ceilalți copii ai poetului au fost ilegitimi.[8] Anne Isabella Byron s-a despărțit de poet la doar la o lună după nașterea Adei, iar poetul a părăsit Anglia pentru totdeauna patru luni mai târziu. Byron a decedat din cauza unei boli în timpul Războiului de independență al Greciei, când Ada avea doar opt ani. Mama Adei a rămas supărată pe Lord Byron și i-a susținut

We notices that the image names in Source folder were suspicious and found that they are the sha256sum of the file:

```
┌──(liviu㉿kali)-[/media/liviu/F6F5-EBC4/Source]
└─$ sha256sum 0a7c51f409c0392f790141f4fa6267bdff9d234fd10ad7c426064188d9126ccb.jpg
0a7c51f409c0392f790141f4fa6267bdff9d234fd10ad7c426064188d9126ccb  0a7c51f409c0392f790141f4fa6267bdff9d234fd10ad7c426064188d9126ccb.jpg
```

Next, these images are also in the Keepass and Remote folders but the names do not match their checksums, after creating a spreadsheet we noticed a pattern:

	A	B	C	D	E	F	G
1	Partition	Directory	Image ID	Hash	Filename	Filename matches Hash	Size
2	First	Source	Mountain B	0a7c51f409c0392f790141f4fa6267bdff9d234fd10ad7c426064188d9126ccb	0a7c51f409c0392f790141f4fa6267bdff9d234fd10ad7c426064188d9126ccb.jpg	Yes	21819546
3	First	Source	Space E	1092b8b4e3fe6e97917c6cf40d89b1507f9350f0de9d58ff6a0a0a28311ce22a	1092b8b4e3fe6e97917c6cf40d89b1507f9350f0de9d58ff6a0a0a28311ce22a.jpg	Yes	4870588
4	First	Source	Space C	1401a1bd86d55d54e1b67ed1657d087738490476828 38a19f99a9ba0fb43cec1	1401a1bd86d55d54e1b67ed1657d087738490476828 38a19f99a9ba0fb43cec1.jpg	Yes	6045007
5	First	Source	Beach C	1dc84a28284240c1d85268fb1c2baaa462caf761f0969da17a0c93ea2e8b29d1	1dc84a28284240c1d85268fb1c2baaa462caf761f0969da17a0c93ea2e8b29d1.jpg	Yes	8048954
6	First	Source	Field A	346dc927140d7c11b8c7304c1aee071d8185134a006ddab42e0dddd2ae9abdd2	346dc927140d7c11b8c7304c1aee071d8185134a006ddab42e0dddd2ae9abdd2.jpg	Yes	1921813
7	First	Source	Road A	397996988beed0a27e1d48ff176054dd23005292093bb1c8e64974008855b9c2	397996988beed0a27e1d48ff176054dd23005292093bb1c8e64974008855b9c2.jpg	Yes	8110483
8	First	Source	Lake A	39ab0cd6708328ebf581388dc7dcf936bbb6ce88169ebff1591c6d479104390f	39ab0cd6708328ebf581388dc7dcf936bbb6ce88169ebff1591c6d479104390f.jpg	Yes	4732060
9	First	Source	Mountain A	3ae8382383161c699525f3c95f5737597419b0d3844be13171e90dfcea130ed1	3ae8382383161c699525f3c95f5737597419b0d3844be13171e90dfcea130ed1.jpg	Yes	2717389
10	First	Source	Space A	3b72aa36b2c4a42f261890ba97b602a5c40ba1941b2f40ca19c532a470bd1f39	3b72aa36b2c4a42f261890ba97b602a5c40ba1941b2f40ca19c532a470bd1f39.jpg	Yes	3271594
11	First	Source	Lake B	544fb10f4084b6c716acca64a9d83415afbb1180dc2416d1c1265110883273fd	544fb10f4084b6c716acca64a9d83415afbb1180dc2416d1c1265110883273fd.jpg	Yes	4380557
12	First	Source	Beach B	5876887a860d2634b48a0a766128261cc69f63cb3d00f5793e399e3b2d2111fb	5876887a860d2634b48a0a766128261cc69f63cb3d00f5793e399e3b2d2111fb.jpg	Yes	290138
13	First	Source	Space D	6a7e3a24bc0556cec9499192866c0b597aa0427599c8d3280b1f65ecefab35a6	6a7e3a24bc0556cec9499192866c0b597aa0427599c8d3280b1f65ecefab35a6.jpg	Yes	7982047
14	First	Source	Field D	75ee0d0170bcf182f0ef6d766723a8a553d7d529f8b949eaadc53bb90dacf8b6	75ee0d0170bcf182f0ef6d766723a8a553d7d529f8b949eaadc53bb90dacf8b6.jpg	Yes	12566660
15	First	Source	Waterfall B	82e2e85d557ddc3163e2943a2b3e9741006cf43a46812408f056ab45afaff5cd	82e2e85d557ddc3163e2943a2b3e9741006cf43a46812408f056ab45afaff5cd.jpg	Yes	8447436
16	First	Source	Tree B	970c6c3ff2bec951907c5c0bcd2252d18e60dd9745adf0d3955534dc89e2f500	970c6c3ff2bec951907c5c0bcd2252d18e60dd9745adf0d3955534dc89e2f500.jpg	Yes	3655537
17	First	Source	Field C	97f952b96f9d267656cf482f8d584a3c0b0eaf8bf237eb01573d84b126fe2684	97f952b96f9d267656cf482f8d584a3c0b0eaf8bf237eb01573d84b126fe2684.jpg	Yes	1689092
18	First	Source	City A	9d954ed4c8a6ba40b993418b336ae4becce513ce31ac6edaffe97ae7d0ddaab5	9d954ed4c8a6ba40b993418b336ae4becce513ce31ac6edaffe97ae7d0ddaab5.jpg	Yes	2158884
19	First	Source	Beach D	b76688b88fae5879e52ffb6ac74b08d38e973e1a2f7fb2e467c8e77836ad22a1	b76688b88fae5879e52ffb6ac74b08d38e973e1a2f7fb2e467c8e77836ad22a1.jpg	Yes	4058373
20	First	Source	Space F	bb0236e303f2ed69eb49466bbd961d178948234114b68a6ca84c569fa5ffb219	bb0236e303f2ed69eb49466bbd961d178948234114b68a6ca84c569fa5ffb219.jpg	Yes	4190674
21	First	Source	Space G	c67b777c47ef1fbc69a6ea47a1dc9095742f8f3c8d771040e301799bb2fb4ecb	c67b777c47ef1fbc69a6ea47a1dc9095742f8f3c8d771040e301799bb2fb4ecb.jpg	Yes	12830525
22	First	Source	Turtle A	ce5d0ed4f3b8641b8fc7e60d1c02de4241c63a119b7990477a6a85784451355f	ce5d0ed4f3b8641b8fc7e60d1c02de4241c63a119b7990477a6a85784451355f.jpg	Yes	7444941
23	First	Source	Space B	cfa686e6fb78330092b69ca8cc15df29bdb702c84e3dca2b40e3ff007f235a2d	cfa686e6fb78330092b69ca8cc15df29bdb702c84e3dca2b40e3ff007f235a2d.jpg	Yes	4232448
24	First	Source	Field B	d6b1ef3dfd2223b73b87f3b7c2a9d7bc270136440236915554 3d6f43fdd51431	d6b1ef3dfd2223b73b87f3b7c2a9d7bc270136440236915554 3d6f43fdd51431.jpg	Yes	4067765
25	First	Source	Beach A	e8c0801cc699e6b6582c4dc115fd045738820e6b21e916bd927f0d03fab533fd	e8c0801cc699e6b6582c4dc115fd045738820e6b21e916bd927f0d03fab533fd.jpg	Yes	6327159
26	First	Source	Tree A	eda921415a263025ff0456b582fd8710965639fc84f63d68c23349e780fcb4ec	eda921415a263025ff0456b582fd8710965639fc84f63d68c23349e780fcb4ec.jpg	Yes	5916835
27	First	Source	Tree C	f22cc5805b9834280d2465ec0784964e07cb39dff426c0257f067299943000 71	f22cc5805b9834280d2465ec0784964e07cb39dff426c0257f067299943000 71.jpg	Yes	14630633
28	First	Source	Waterfall A	f90fa2d93b4cd61b8321bff7e9d880f898379244fa3f90bddfe0c85b28bcdccb	f90fa2d93b4cd61b8321bff7e9d880f898379244fa3f90bddfe0c85b28bcdccb.jpg	Yes	20299747
29	Second	Remote	Space E	1092b8b4e3fe6e97917c6cf40d89b1507f9350f0de9d58ff6a0a0a28311ce22a	007db8b4e3fe6e97917c6cf40d89b1507f9350f0de9d58ff6a0a0a28311ce22a.jpg	No	4870588
30	Second	Remote	Beach C	1dc84a28284240c1d85268fb1c2baaa462caf761f0969da17a0c93ea2e8b29d1	01fb4a28284240c1d85268fb1c2baaa462caf761f0969da17a0c93ea2e8b29d1.jpg	No	8048954
31	Second	Remote	Tree A	eda921415a263025ff0456b582fd8710965639fc84f63d68c23349e780fcb4ec	02b421415a263025ff0456b582fd8710965639fc84f63d68c23349e780fcb4ec.jpg	No	5916835
32	Second	Remote	Tree C	f22cc5805b9834280d2465ec0784964e07cb39dff426c0257f067299943000 71	03c2c5805b9834280d2465ec0784964e07cb39dff426c0257f067299943000 71.jpg	No	14630633
33	Second	KeyPass	Beach B	5876887a860d2634b48a0a766128261cc69f63cb3d00f5793e399e3b2d2111fb	4376887a860d2634b48a0a766128261cc69f63cb3d00f5793e399e3b2d2111fb.jpg	No	290138
34	Second	KeyPass	Space D	6a7e3a24bc0556cec9499192866c0b597aa0427599c8d3280b1f65ecefab35a6	6a613a24bc0556cec9499192866c0b597aa0427599c8d3280b1f65ecefab35a6.jpg	No	7982047
35	Second	KeyPass	Field D	75ee0d0170bcf182f0ef6d766723a8a553d7d529f8b949eaadc53bb90dacf8b6	75ee720170bcf182f0ef6d766723a8a553d7d529f8b949eaadc53bb90dacf8b6.jpg	No	12566660
36	Second	KeyPass	Waterfall B	82e2e85d557ddc3163e2943a2b3e9741006cf43a46812408f056ab45afaff5cd	82e2e8645557ddc3163e2943a2b3e9741006cf43a46812408f056ab45afaff5cd.jpg	No	8447436
37	Second	KeyPass	Tree B	970c6c3ff2bec951907c5c0bcd2252d18e60dd9745adf0d3955534dc89e2f500	970c6c3f61bec951907c5c0bcd2252d18e60dd9745adf0d3955534dc89e2f500.jpg	No	3655537
38	Second	KeyPass	Field C	97f952b96f9d267656cf482f8d584a3c0b0eaf8bf237eb01573d84b126fe2684	97f952b96f6e267656cf482f8d584a3c0b0eaf8bf237eb01573d84b126fe2684.jpg	No	1689092
39	Second	KeyPass	City A	9d954ed4c8a6ba40b993418b336ae4becce513ce31ac6edaffe97ae7d0ddaab5	9d954ed4c8a66f40b993418b336ae4becce513ce31ac6edaffe97ae7d0ddaab5.jpg	No	2158884
40							

The images in Remote folder are different by 4 characters and the ones from Keepass are different by 2.

The differences from the Keepass images are very easily converted in ASCII like so:

- 43 = C
- 61 = a
- 72 = r
- 64 = d
- 61 = a
- 6e = n
- 6f = o

After some tinkering we noticed a pattern in the filename column, the numbers were ascending (`00`, `01`, `02` and `03`) and we guessed that the remaining 2 numbers are hex:

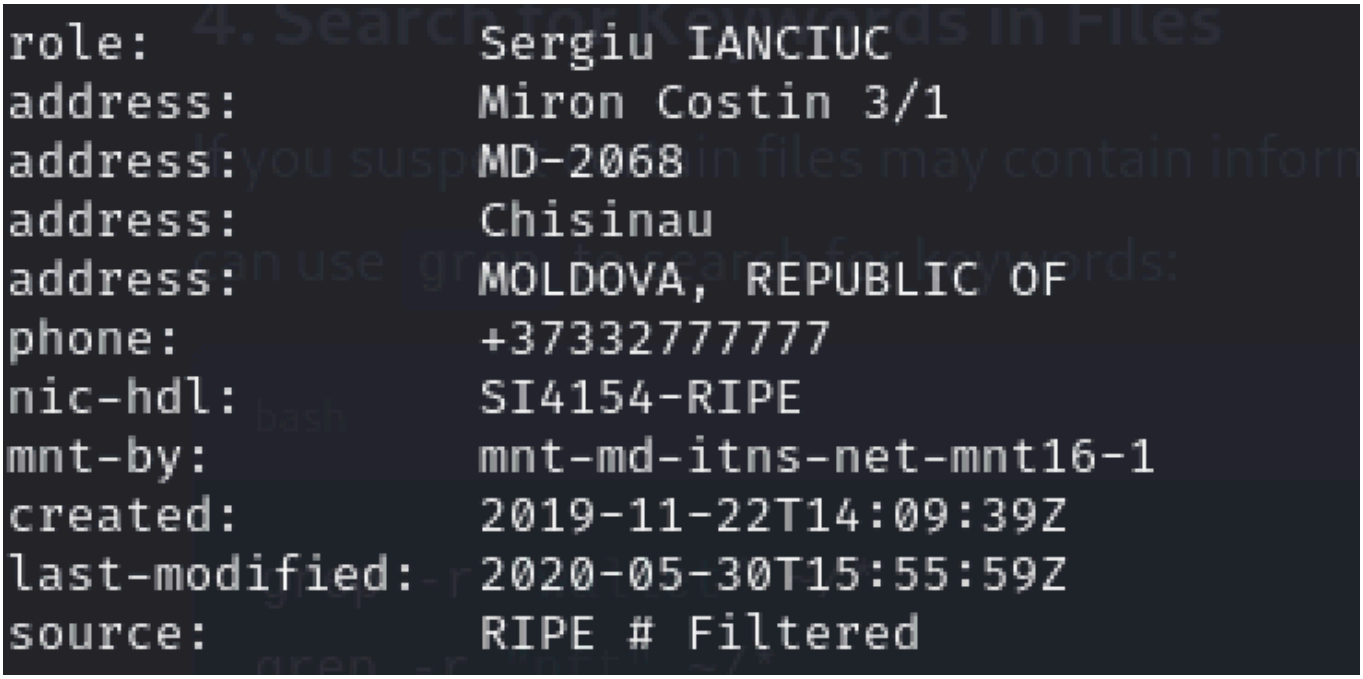
Each two hex digits can be converted into a decimal number:

- `00` = `0`
- 7d = 125
- `01` = 1

- fb = 251
- 02 = 2
- b4 = 180
- 03 = 3
- c2 = 194

This looks exactly like an IP but when we try it it doesn't work so we try to reverse it:

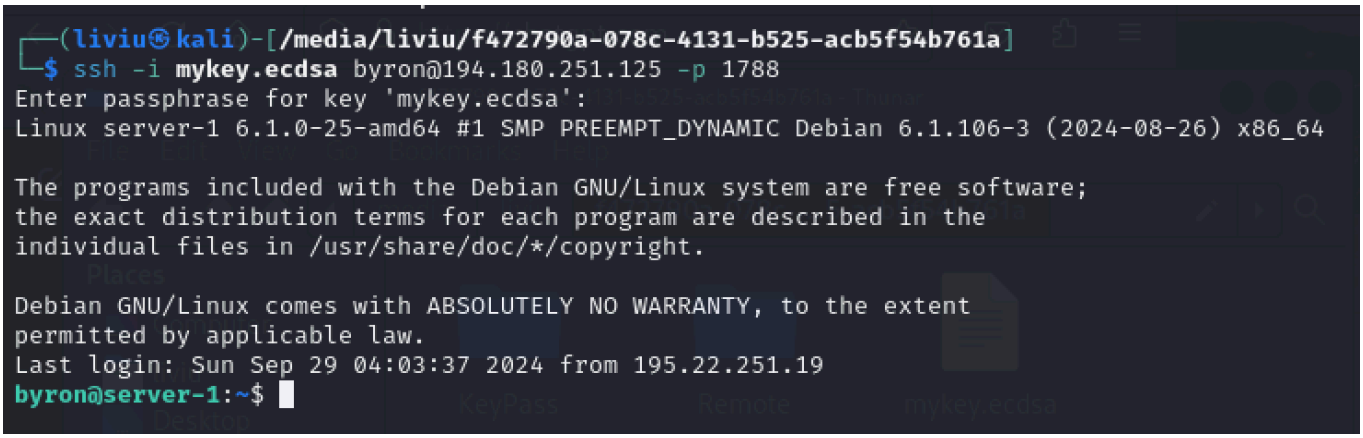
-instead of: 125.251.180.194
-we try: 194.180.251.125



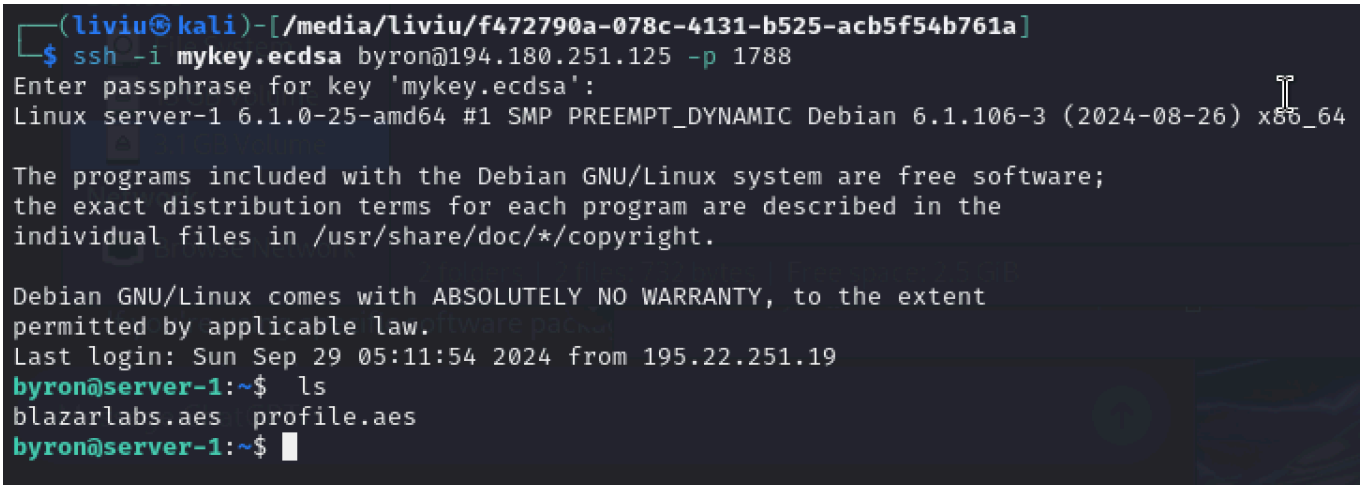
We see that this Ip works and also it's from Moldova!

After a nmap scan we see that port 1788 is open and supports ssh.

Next we try to connect to it via ssh, with the sshkeys from the usb drive using the crime leaders name "byron" and using the password "Cardano".



Inside we see 2 very interesting files, and these are the wallet key and the second is probably used to show us how it was made:



fileul blazarlabs.aes pare suspect deci cautam ce poate insemna, gasim aceasta companie cu lucratori din md:

BlazarLabs

Software Development · 6 followers · 2-10 employees

+ Follow | Message | ···

Home | About | Posts | Jobs | People | Insights

2 associated members

Search employees by title, keyword or school

Where they live

+ Add

1 | Colombia

1 | United Kingdom

1 | England, United Kingdom

1 | Bolívar, Colombia

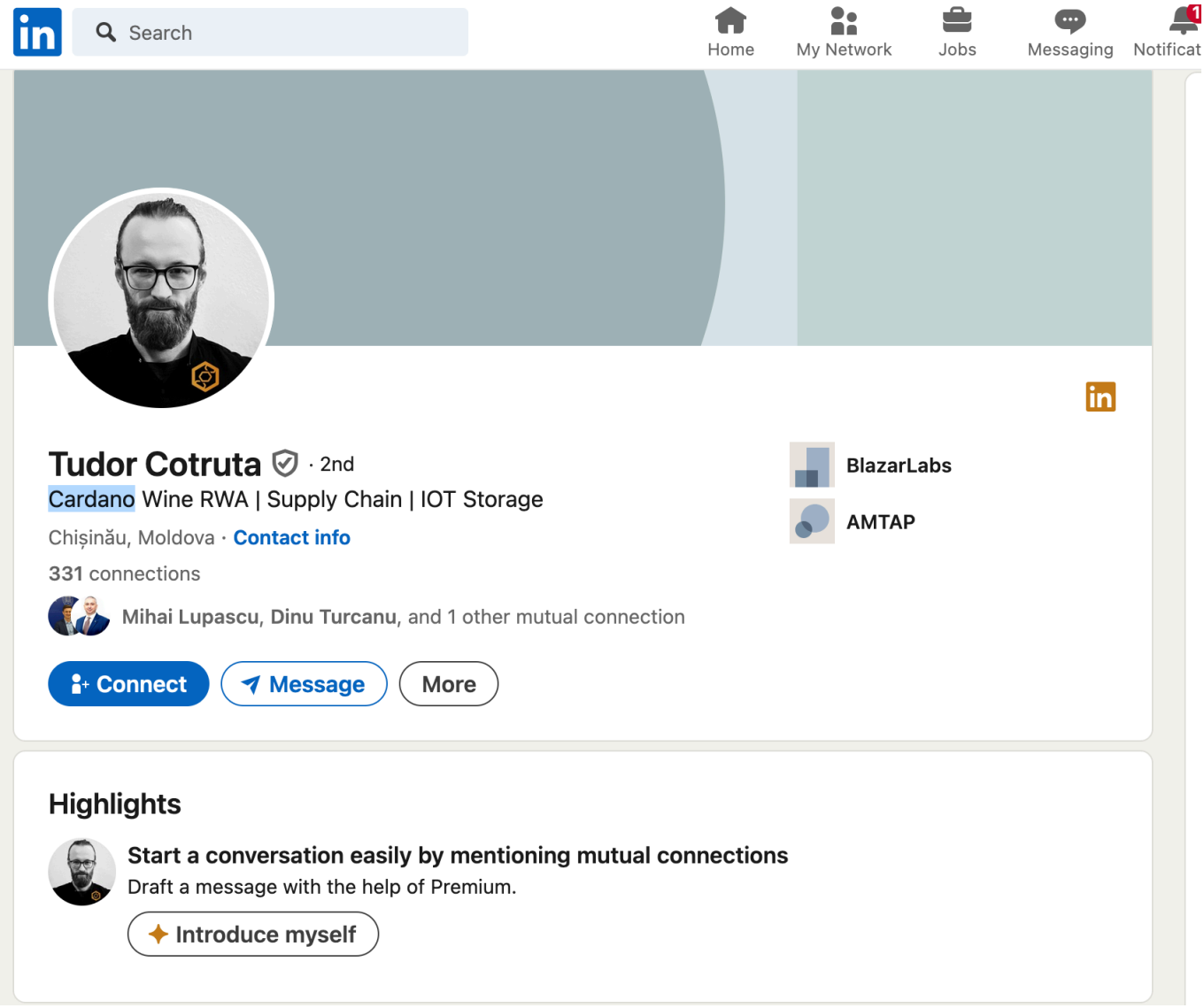
Where they studied

+ Add

1 | Universidad Jorge Tadeo Lozano

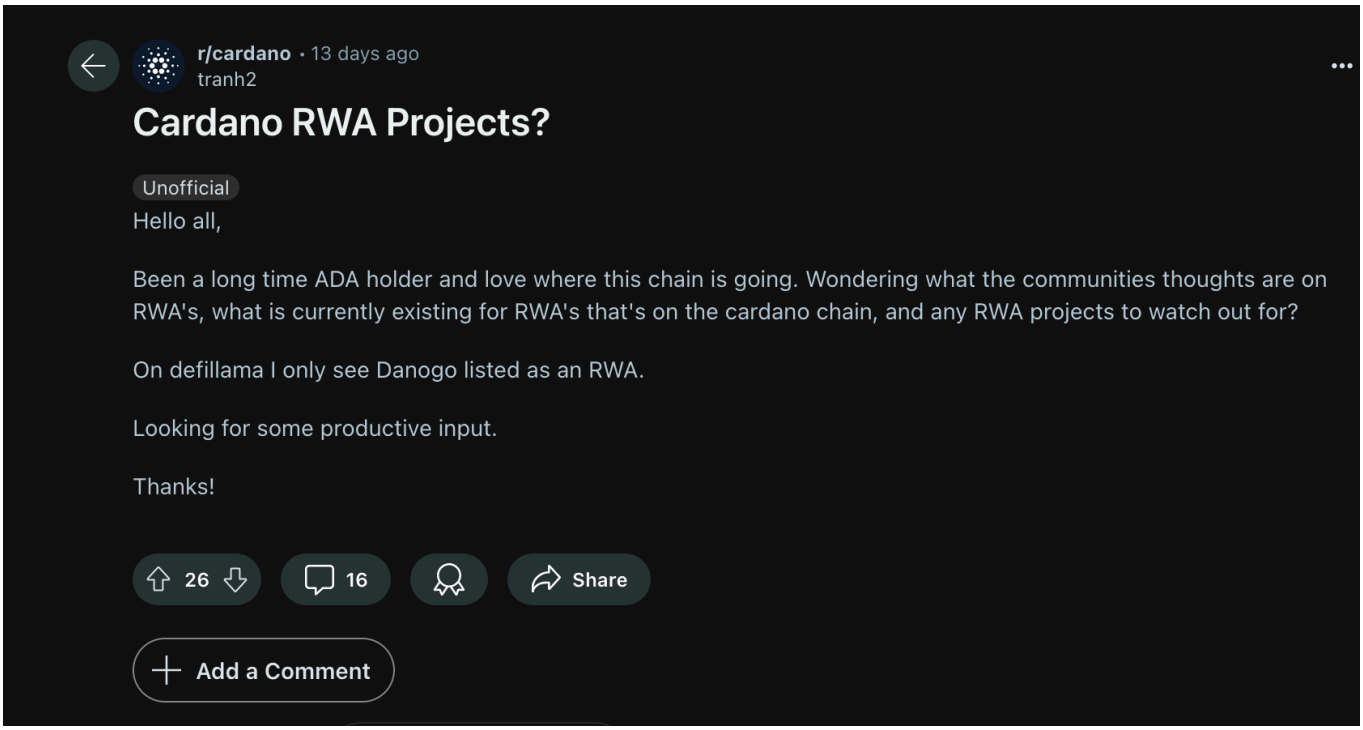
1 | Parsons School of Design - The New Sch...

Show more



Deasemenea are "Cardano" in bio ceia ce era parola la ssh.

Vedem ca este creatorul sau un dev la un blockchain "cardano Wine RWA":

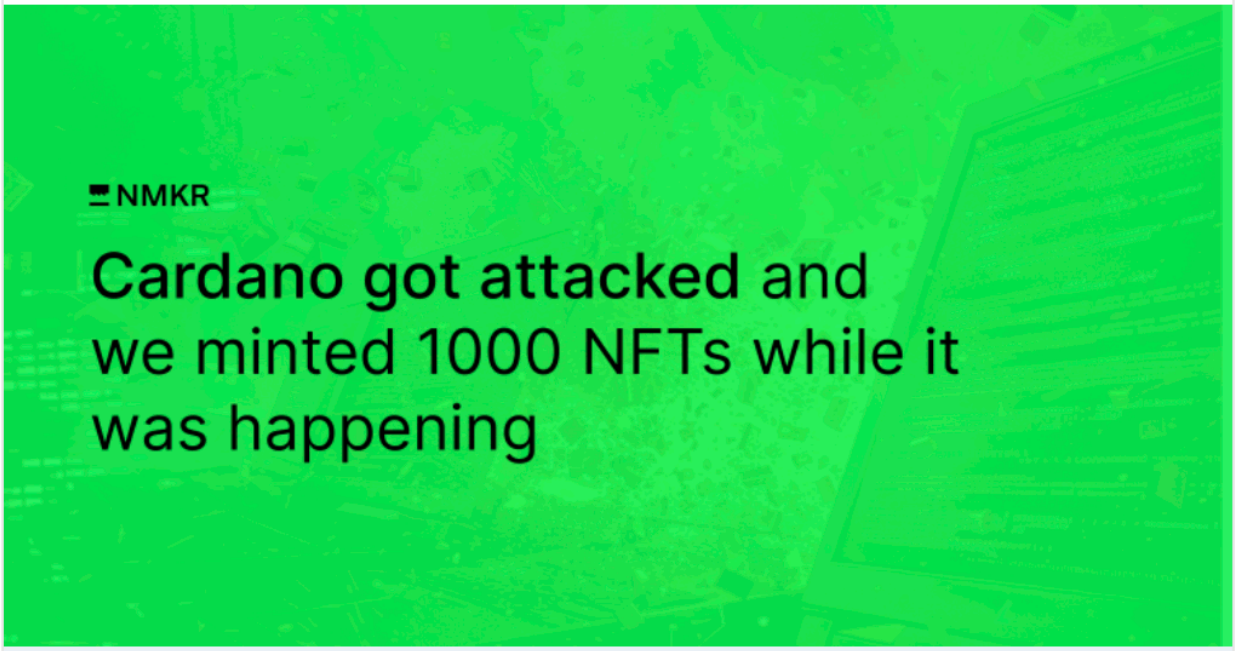

r/cardano • 13 days ago
tranh2
Cardano RWA Projects?
Unofficial
Hello all,
Been a long time ADA holder and love where this chain is going. Wondering what the communities thoughts are on RWA's, what is currently existing for RWA's that's on the cardano chain, and any RWA projects to watch out for?
On defillama I only see Danogo listed as an RWA.
Looking for some productive input.
Thanks!
26
16
Share
Add a Comment

Tudor Cotruta reposted this

Patrick Tobler • 2nd
CEO & Founder NMKR
3mo •

+ Follow

So this happened yesterday.



Cardano got attacked and we minted 1000 NFTs while it was happening

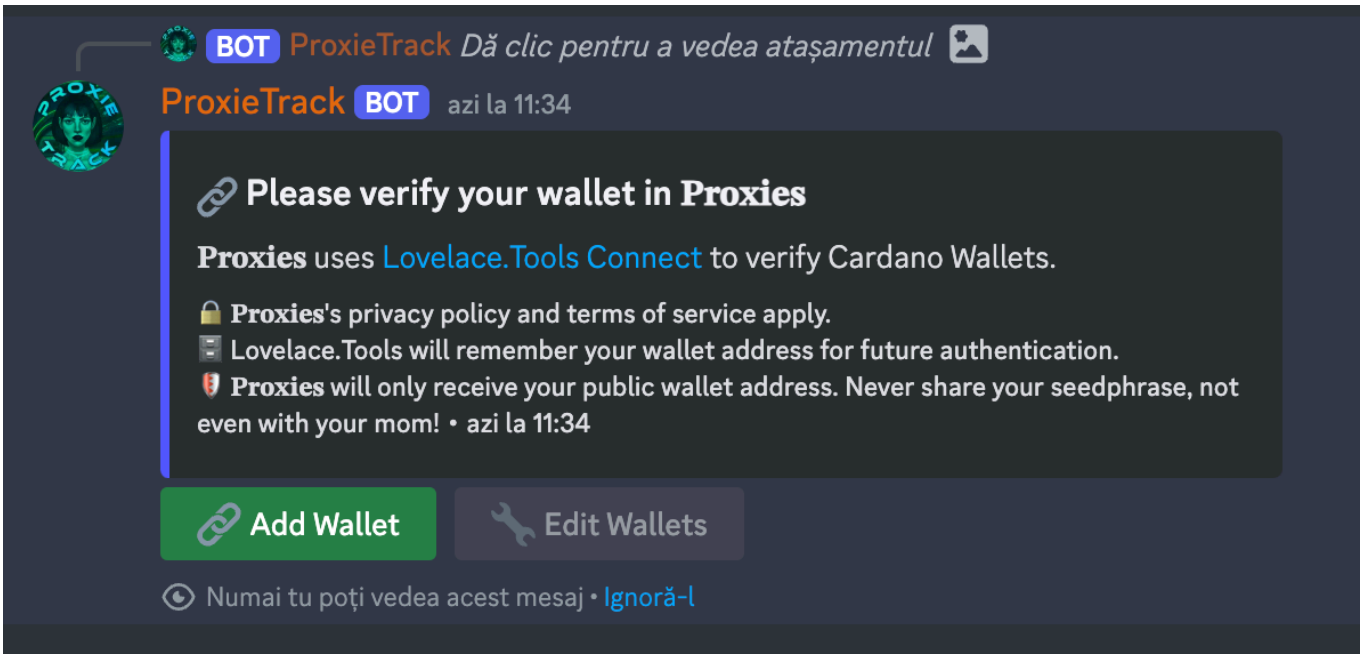
Patrick Tobler on LinkedIn • 2 min read

Yesterday, the Cardano blockchain faced a high-stakes DDOS attack, an attempt to ...

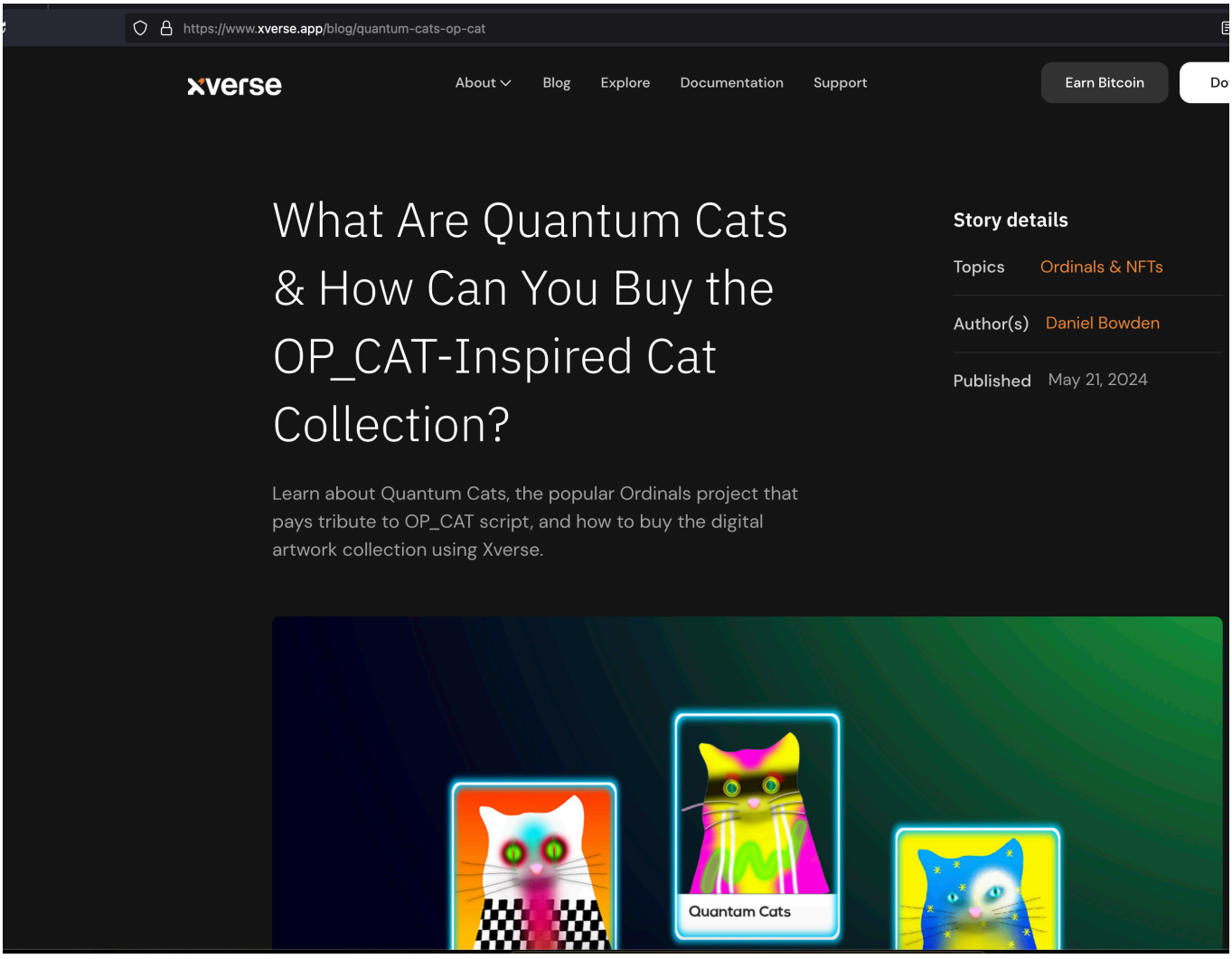
Tudor Cotruta and 97 others

7 comments · 5 reposts

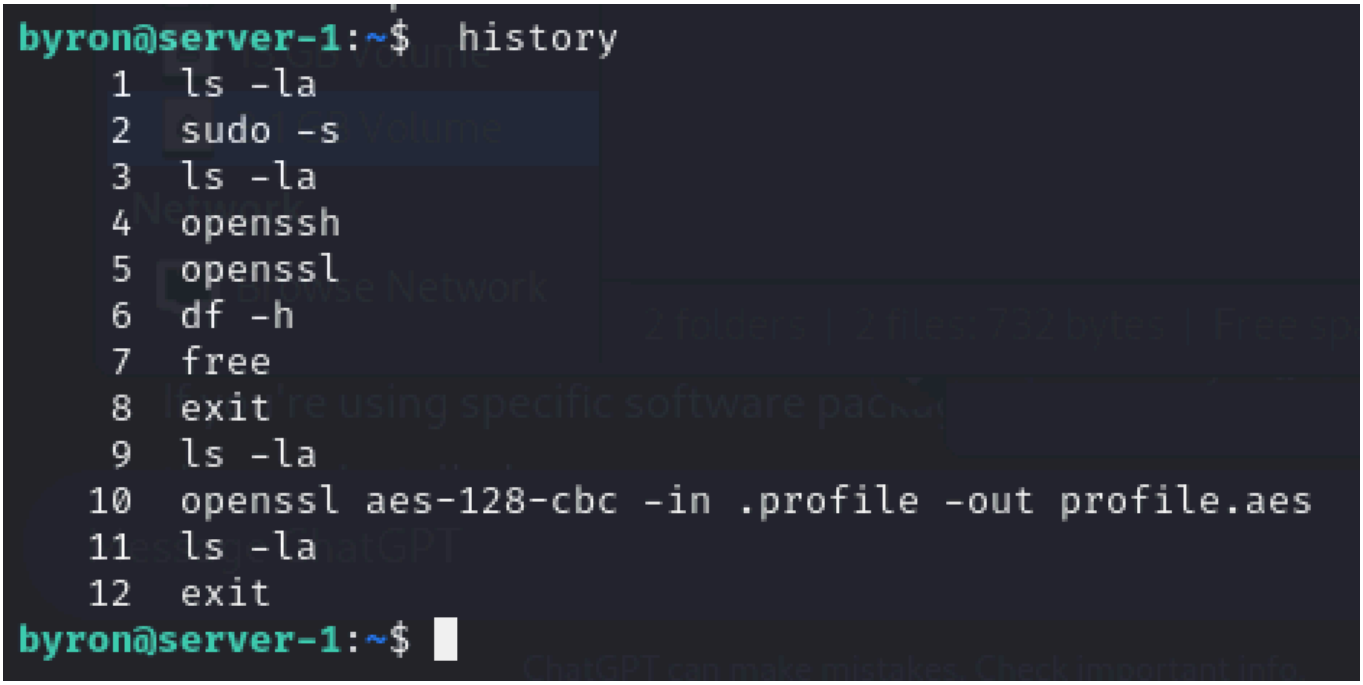
Like Comment Repost Send



https://cardanoproxies.tradingtent.io/connect-wallet



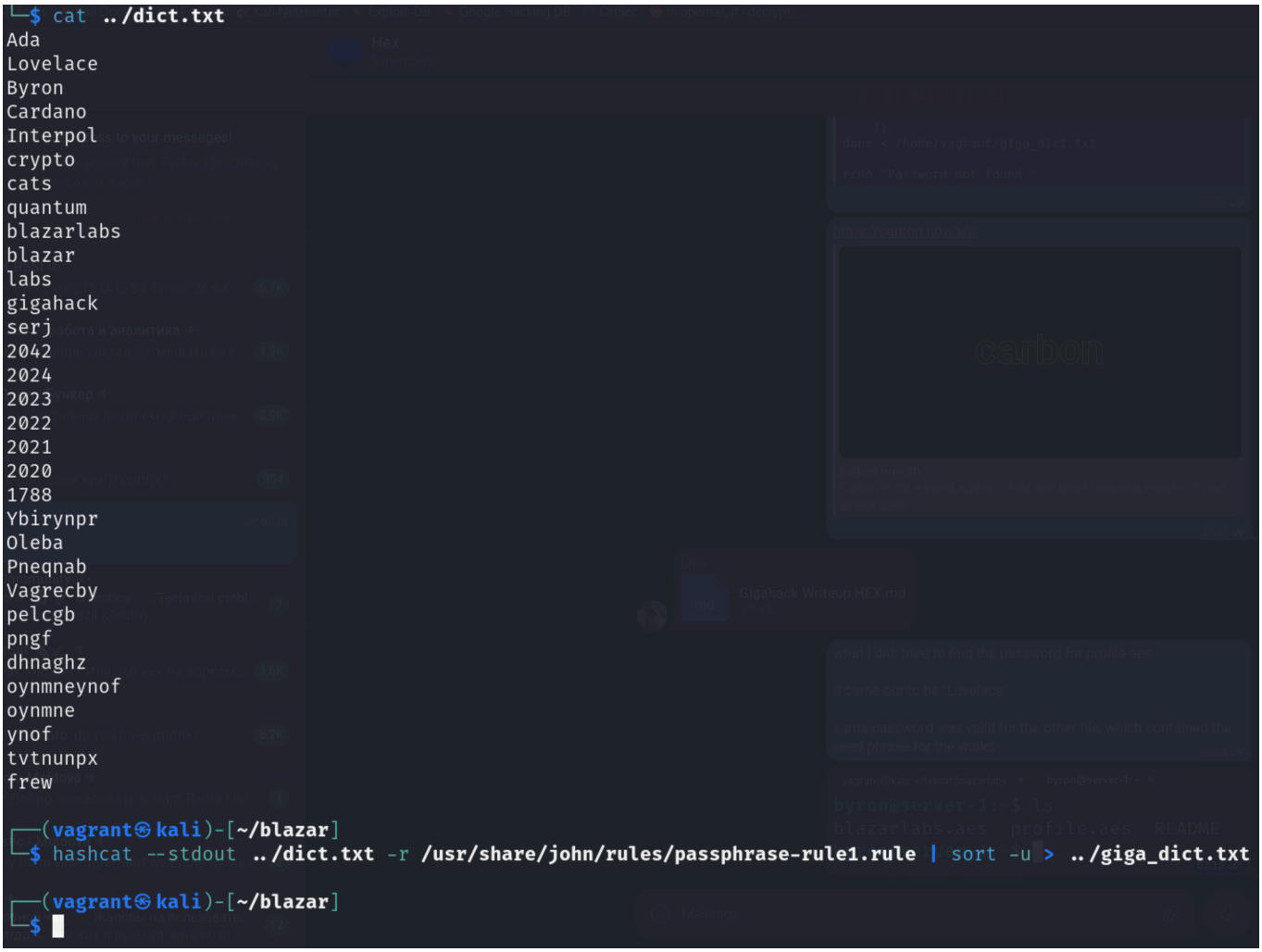
Here we can see how another file was encrypted:



What we did: tried to find the password for profile.aes

It came out to be "Lovelace"

Same password was valid for the other file, which contained the seed phrase for the wallet
Down bellow is the dictionary I used for brute forcing the passphrase that was used for encrypting profile.


└─$ cat ../dict.txt
Ada
Lovelace
Byron
Cardano
Interpol
crypto
cats
quantum
blazarlabs
blazar
labs
gigahack
serj
2042
2024
2023
2022
2021
2020
1788
Ybirynpr
Oleba
Pneqnab
Vagrecby
pelcgb
pngf
dhnaghz
oynmneynof
oynmne
ynof
tvtnunpx
frew
┌──(vagrant㉿kali)-[~/blazar]
└─$ hashcat --stdout ../dict.txt -r /usr/share/john/rules/passphrase-rule1.rule | sort -u > ../giga_dict.txt
┌──(vagrant㉿kali)-[~/blazar]
└─$

```
#!/bin/bash

# Define the target hash
TARGET_HASH="f4e81ade7d6f9fb342541152d08e7a97"

# Loop through each word in the dictionary
while IFS= read -r w; do
    # Decrypt the file using the current word as the password
    openssl aes-128-cbc -d -in /home/vagrant/profile.aes -out /home/vagrant/
profile_decrypted -k "$w" 2>/dev/null

    # Check if the decryption was successful by comparing the hash
    if [ -f /home/vagrant/profile_decrypted ]; then
        CURRENT_HASH=$(md5sum /home/vagrant/profile_decrypted | awk '{ print $1
}')
        if [ "$CURRENT_HASH" == "$TARGET_HASH" ]; then
            echo "Password found: $w"
            rm profile_decrypted  # Optional: clean up decrypted file
            exit 0  # Exit the script after finding the password
        fi
        rm profile_decrypted  # Clean up decrypted file for the next iteration
    fi
done < /home/vagrant/giga_dict.txt

echo "Password not found."
```

This is how we found the passphrase Lovelace that was the same passphrase used for encrypting blazarlabs.aes.

We decrypted the file using Lovelace and got an archive. After the extraction we got 2 screenshots that contained the crypto wallet app name and the seed phrase.

After getting into the account, we created another wallet and confiscated all the crypto currency and 2 NFTs.

New Transaction

1 Cardano Address

addr1q925x9kmpfjup44684lvcfg8c68w7dvlpqr3ag0w2q9a9vxqx63
l2kvf8d6pazrm4st0zjud40wk0rfqnne933mhw5cq7ey8p9

2 Tokens

198.64666 Ada

3 Collectibles

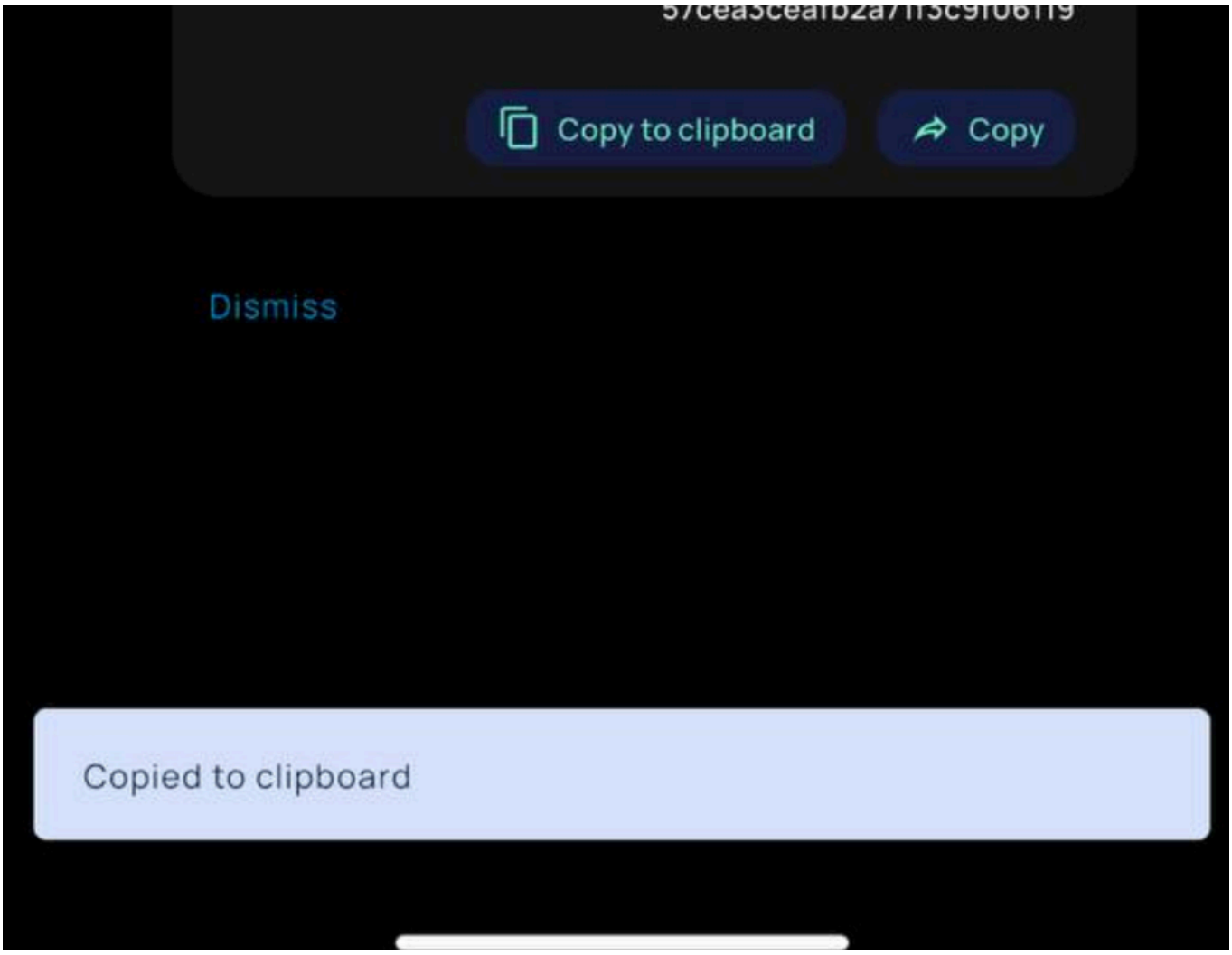
Wisdom#0257
Easelle#0127

4 Fees Overview

Total fees: 0.172233₳

5 ✓ Transaction Submitted

Tx Hash 6b359b1b5ac0b34b5b0d8a811fcae2329d3dce2


57cea3cea1b2a7113c9f06119
Copy to clipboard
Copy
Dismiss
Copied to clipboard

14:29

4G 46

HEX team

$78.22

+ Buy ADA

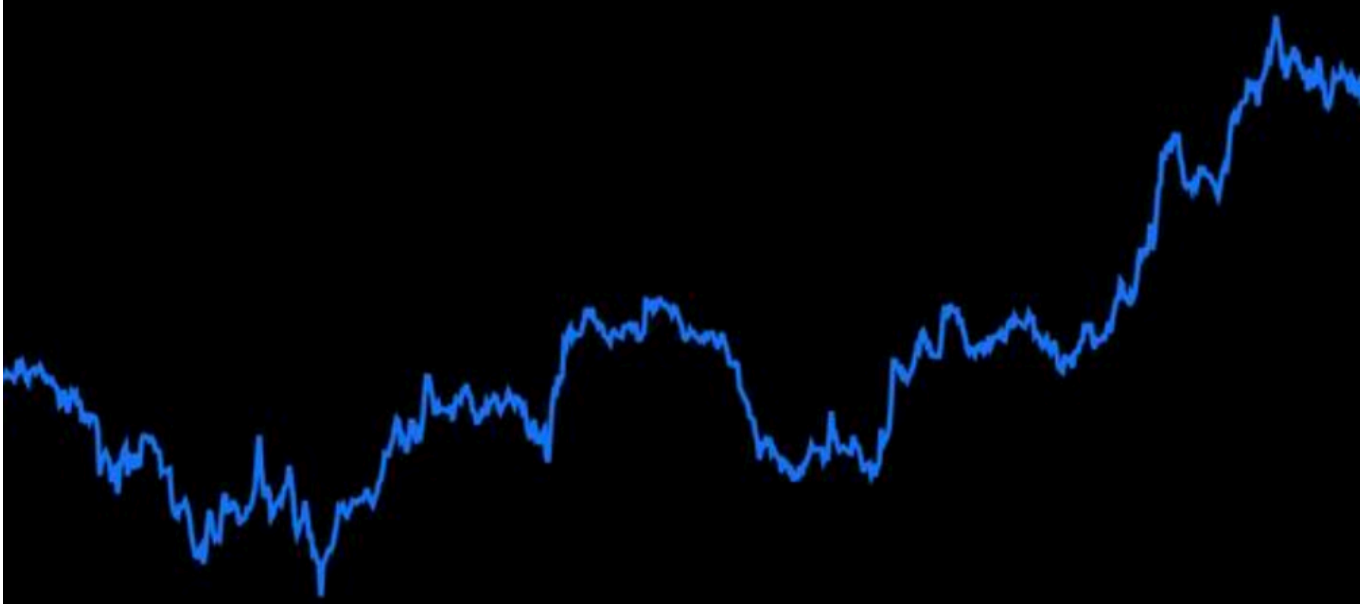
Portfolio Balance

Cardano

198.65 ADA

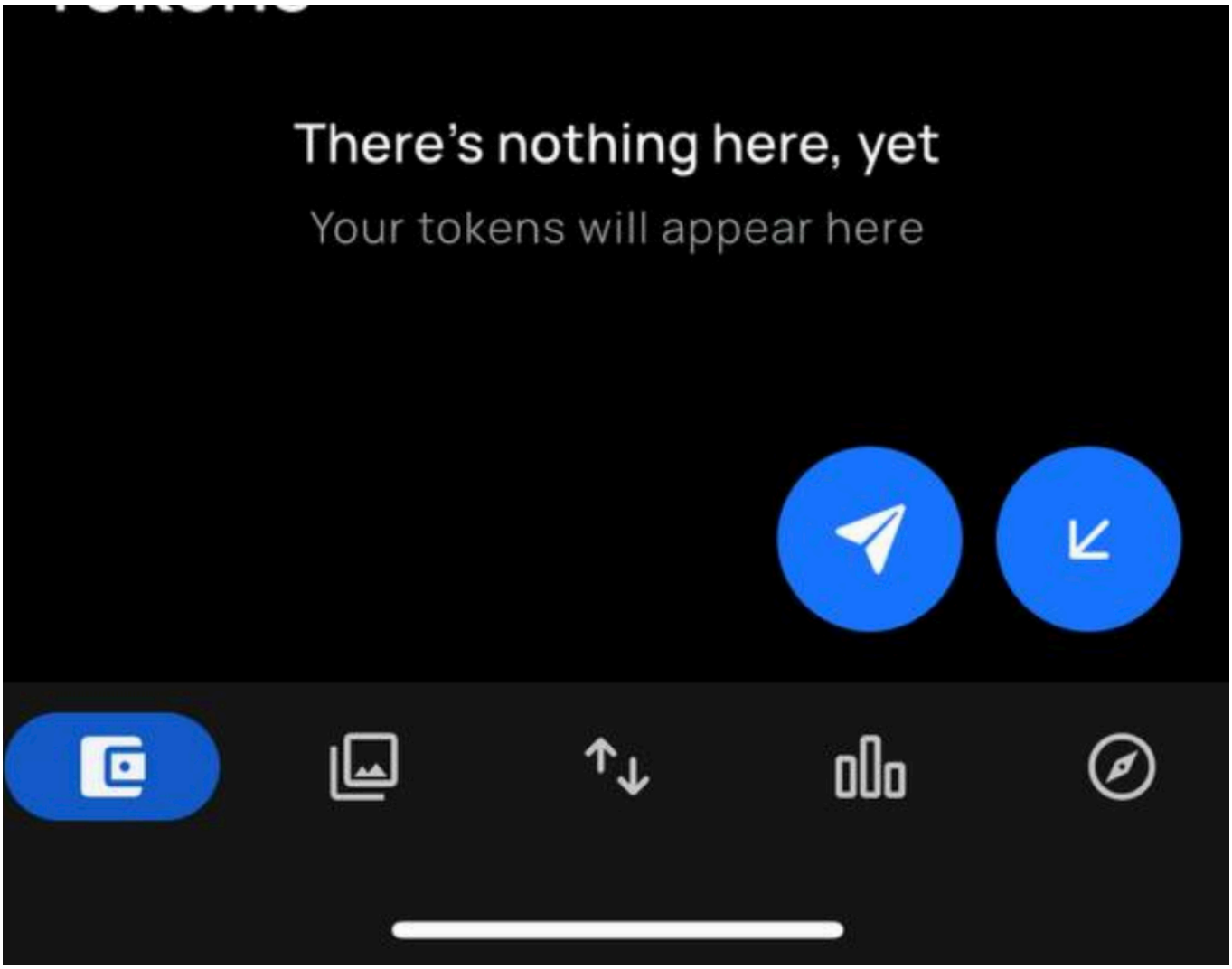
$0.39 +12.25%

$78.22



1 Y 3 M 1 M 1 W 24 H

Tokens


There’s nothing here, yet
Your tokens will appear here

14:29 4G 46

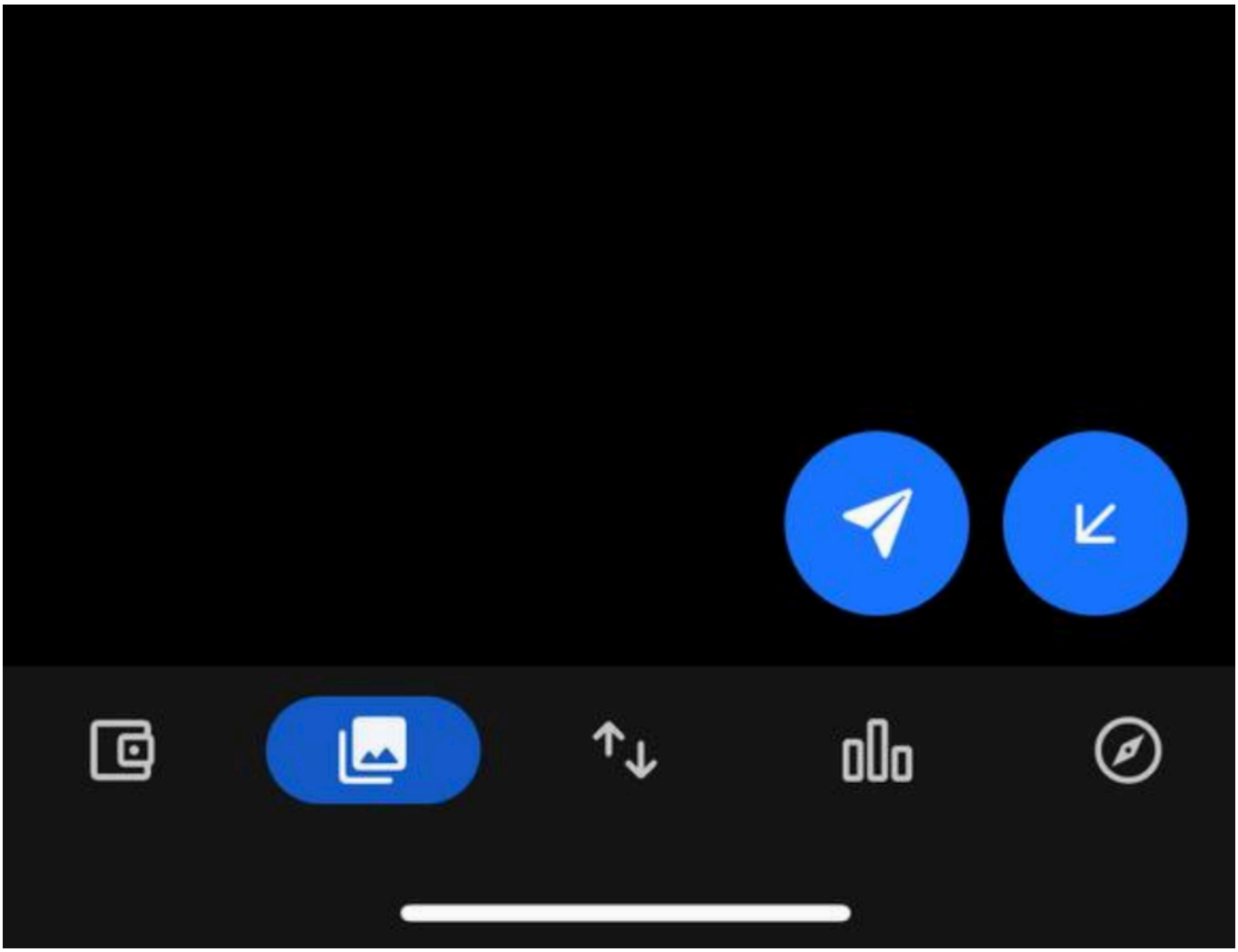
Collectibles

For Sale

Search collectibles

Wisdom Proxies
Floor price: 44 ADA
1 NFTs >

CardanoProxies
Floor price: 32 ADA
1 NFTs >



This is how we completed the challenge.

-HEX team