

vishwaCTF

CHALLENGE NAME : [Boot-Failed]

DEV : [Samarth Ghante]

CATEGORY : [Web]

LEVEL : [Hard]



2024

Challenge Name: Boot-Failed

Challenge Description: Fix The System ASAP!

Home Page

```
----- BOOTING OF -----  
Loading...  
[OK].  
  
Howdy!  
My Name is Samarth Ghante.  
You can reach me out on my [LinkedIn].  
----- BOOTUP LOGS -----  
  
[0.000000] Initializing cgroup subsys cpuset  
[0.000000] Initializing cgroup subsys cpuacct  
[0.000000] Linux version 4.15.0-29-generic (builddd@lgw01-amd64-039) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #31-Ubuntu SMP Tue Jul 17 15:39:52 UTC 2018 (Ubuntu 4.15.0-29.31-generic 4.15.18)  
[0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.15.0-29-generic root=UUID=0a3407de-e4c6-48e7-98d6-6158131746d0 ro quiet splash vt.handoff=1  
[0.000000] KERNEL supported cpus:  
[0.000000]   Intel GenuineIntel  
[0.000000]   AMD AuthenticAMD  
[0.000000]   Centaur CentaurHauls  
[0.000000] x86/fpu: Supporting XSAVE feature 0x01: 'x87 floating point registers'  
[0.000000] x86/fpu: Supporting XSAVE feature 0x02: 'SSE registers'  
[0.000000] x86/fpu: Supporting XSAVE feature 0x04: 'AVX registers'  
[0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256  
[0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.  
[0.000000] e820: BIOS-provided physical RAM map:  
[0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable  
[0.000000] BIOS-e820: [mem 0x000000000000009fc00-0x0000000000009ffff] reserved  
[0.000000] BIOS-e820: [mem 0x000000000000f0000-0x000000000000ffff] reserved  
[0.000000] BIOS-e820: [mem 0x00000000000100000-0x0000000007bf00000] usable  
[0.000000] BIOS-e820: [mem 0x0000000007c00000-0x0000000007ffff000] reserved  
[0.000000] Notice: System is Running out of RAM. Upgrade your memory to 8GB.  
[0.000000] Memory: 4096MB = 4096MB total  
[0.000000] Memory: 4083564k/4083564k available, 1231556k reserved, 0K highmem  
[0.000000] NX (Execute Disable) protection: active  
[0.000000] Kernel/User page tables isolation: enabled  
[0.000000] ftrace: allocating 34004 entries in 133 pages  
...  
...  
...  
  
EOF; -
```

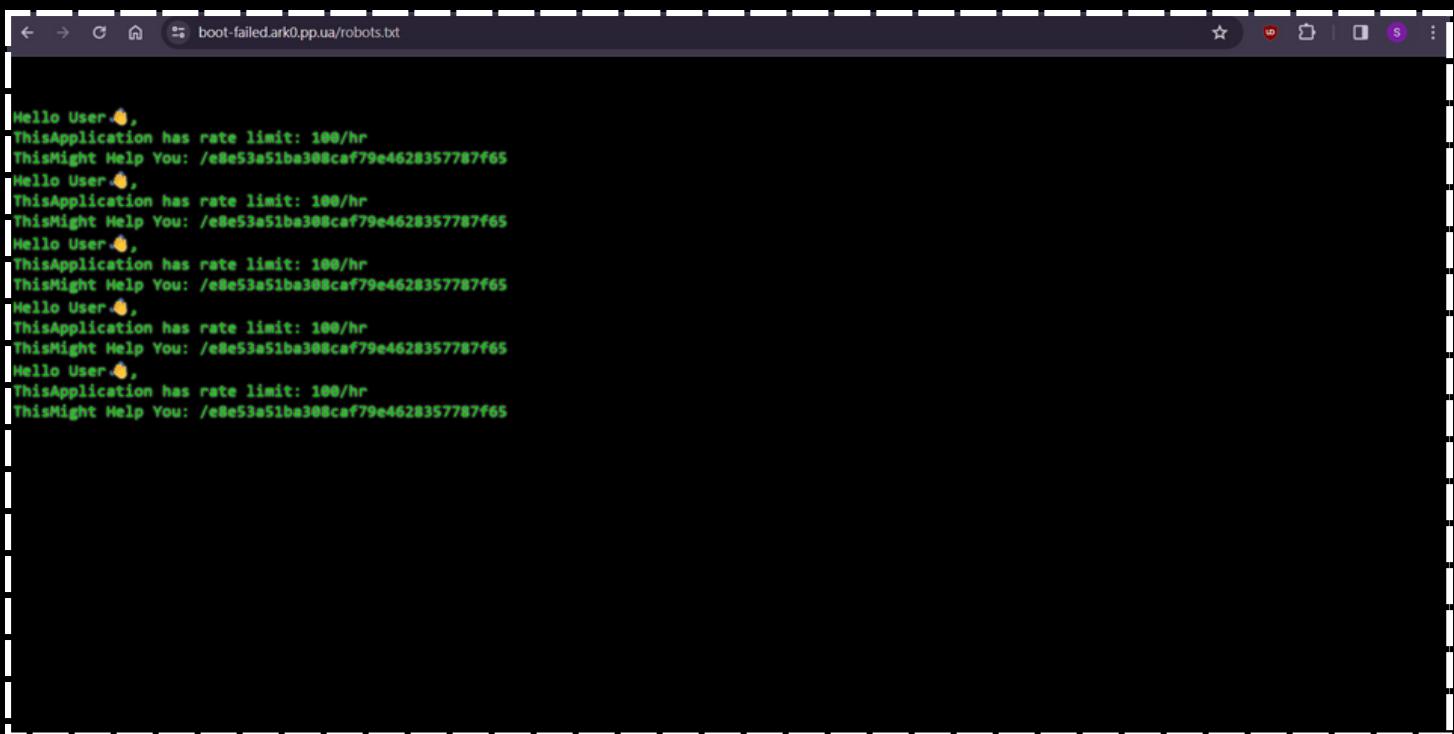
It shows the boot logs of the system!
& There is a error notice highlighted

[0.000000] Notice: System is Running out of RAM. Upgrade your memory to 8GB.

Remember, We have to upgrade the RAM to 8GB

Now, We know what to solve in this! Lets Begin!!!

Go To /Robots.txt



A screenshot of a web browser window displaying the contents of the robots.txt file from the URL boot-failed.ark0.pp.ua/robots.txt. The page title is "boot-failed.ark0.pp.ua/robots.txt". The content of the robots.txt file is as follows:

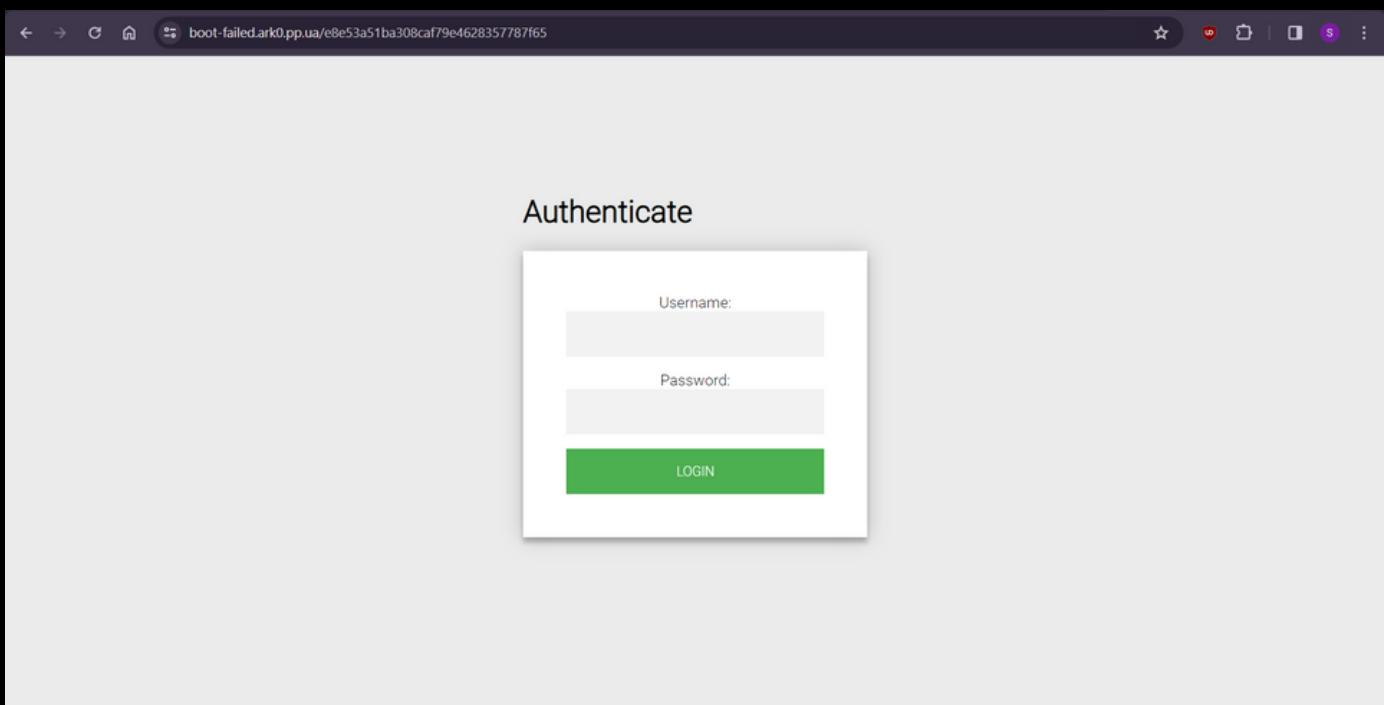
```
Hello User.❶,  
ThisApplication has rate limit: 100/hr  
ThisMight Help You: /e8e53a51be308caf79e4628357787f65  
Hello User.❷,  
ThisApplication has rate limit: 100/hr  
ThisMight Help You: /e8e53a51be308caf79e4628357787f65  
Hello User.❸,  
ThisApplication has rate limit: 100/hr  
ThisMight Help You: /e8e53a51be308caf79e4628357787f65  
Hello User.❹,  
ThisApplication has rate limit: 100/hr  
ThisMight Help You: /e8e53a51be308caf79e4628357787f65  
Hello User.❺,  
ThisApplication has rate limit: 100/hr  
ThisMight Help You: /e8e53a51be308caf79e4628357787f65
```

You have press key on keyboards to reveal the msg...

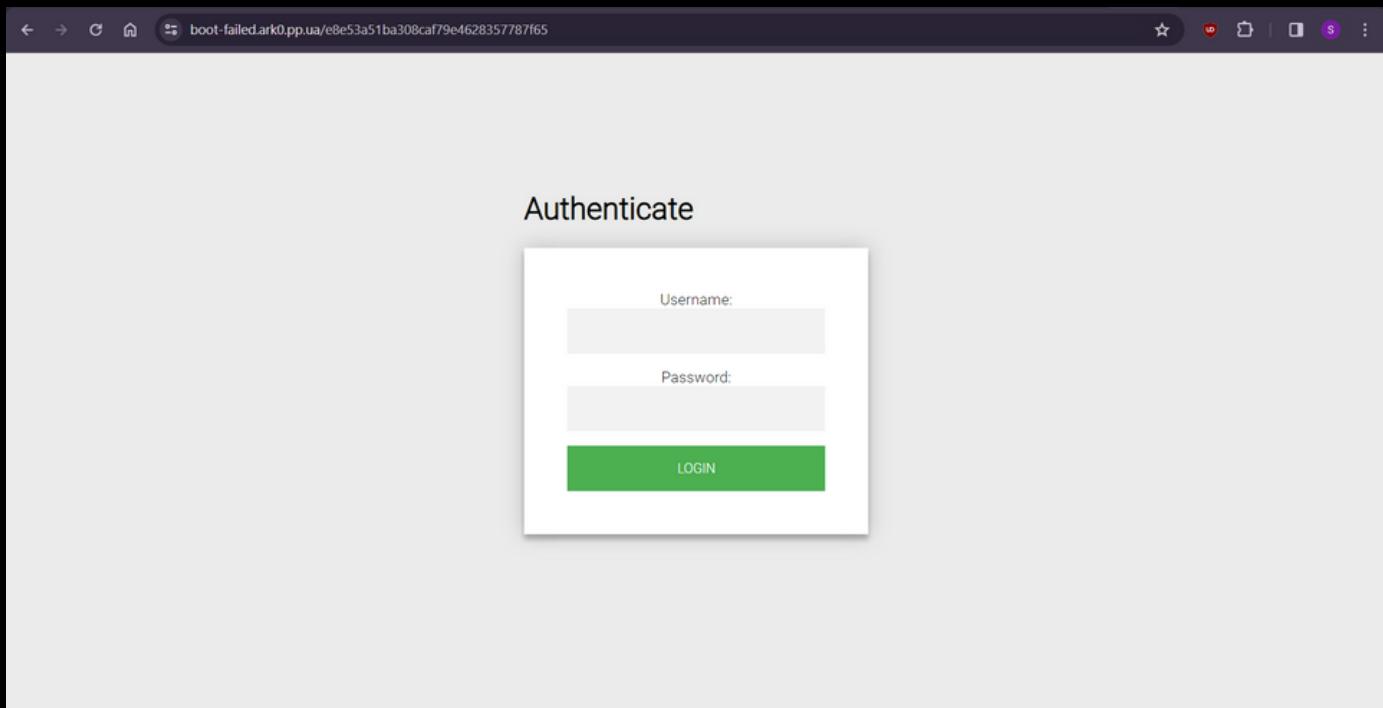
← → ⌛ ⌂ boot-failed.ark0.pp.ua/robots.txt

```
Hello User👋,  
ThisApplication has rate limit: 100/hr  
This Might Help You: /e8e53a51ba308caf79e4628357787f65  
Hello User👋,  
ThisApplication has rate limit: 100/hr  
This Might Help You: /e8e53a51ba308caf79e4628357787f65  
Hello User👋,  
ThisApplication has rate limit: 100/hr  
This Might Help You: /e8e53a51ba308caf79e4628357787f65  
Hello User👋,  
ThisApplication has rate limit: 100/hr  
This Might Help You: /e8e53a51ba308caf79e4628357787f65  
Hello User👋,  
ThisApplication has rate limit: 100/hr  
This Might Help You: /e8e53a51ba308caf79e4628357787f65
```

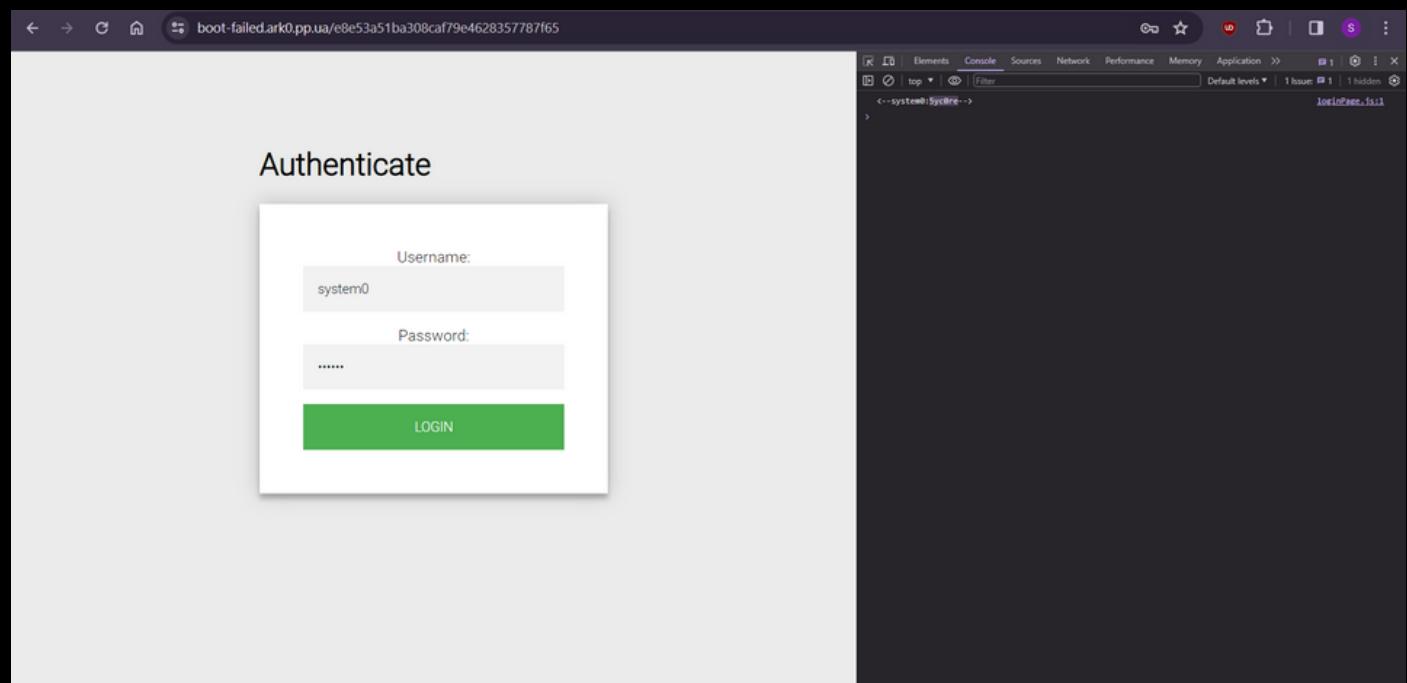
Go to /e8e53a51ba308caf79e4628357787f65



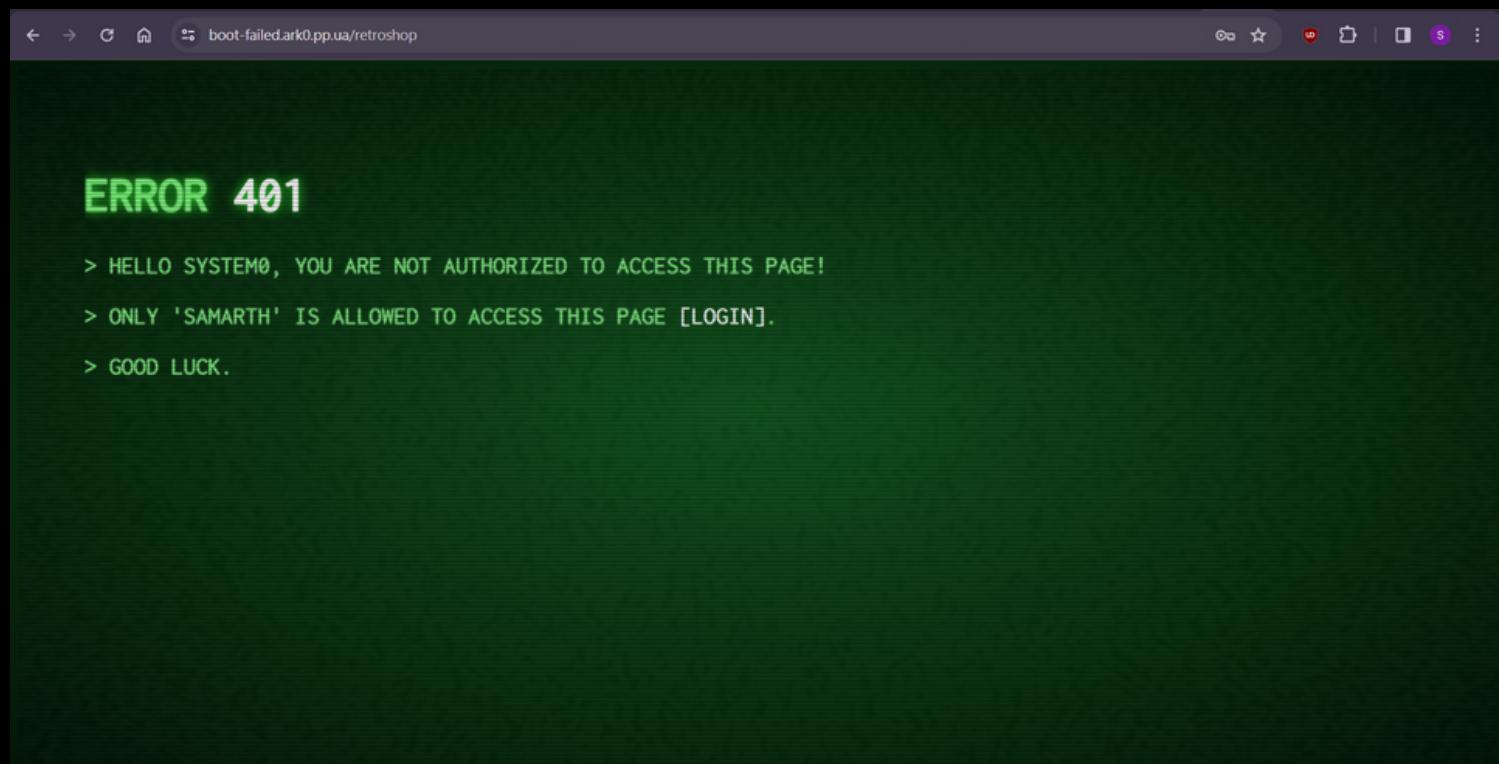
You have to find the login credentials!



login creds are in console log --> system0:5yc0re



After You Authenticate with the login creds



It shows error, “System0” is not authorized to access the page, only “samarth” is authorized to do so!

So We have to somehow login as the samarth. Let’s check if this site is using token system for user session. (if its JWT Token, then it could be cracked)

ERROR 401

> HELLO SYSTEM0, YOU ARE NOT AUTHORIZED TO ACCESS THIS PAGE!

> ONLY 'SAMARTH' IS ALLOWED TO ACCESS THIS PAGE [LOGIN].

> GOOD LUCK.

Name	Value	Domain	Path	Expires / ..	Size	Hit...	Sec...	Sa...	Part...	Prio...
ctoken	nFBEMO4Nw...	giphy.com	/	2024-12-..	73	✓	Lax	Me...	Me...	Me...
didomi_accept_c...	1	giphy.com	/	2025-01-..	21	✓	None	Me...	Me...	Me...
didomi_token	eyJ1Z2yX2ld...	giphy.com	/	2025-01-..	516	✓	None	Me...	Me...	Me...
euconsent-v2	CP3MULP3PA...	giphy.com	/	2025-01-..	73	✓	None	Me...	Me...	Me...
token	eyJhbGciOiU...	boot-failedark...	/	Session	138					

It is having a token key value pair in the cookies!

Let's Check if it is a JWT (Json Web Token) Token?

Catch our webinar on next-gen authorization: Okta Fine Grained Authorization →

JWNT Debugger Libraries Introduction Ask Crafted by Auth0 by Okta

JSON Web Tokens are an open, industry standard [RFC 7519](#) method for representing claims securely between two parties.

JWT.IO allows you to decode, verify and generate JWT.

LEARN MORE ABOUT JWT SEE JWT LIBRARIES

Debugger

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJc2VybmFtZSI6InN5c3RlbTAiLCJpYXQiOjE3MTE1MjE5NjN9.a6IID9VT4b5DxU8o7mbbQoA9_ms  
srH5PpfJ707KJcQ
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYLOAD: DATA

```
{  
  "username": "system0",  
  "iat": 1711521963  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
)  secret base64 encoded
```

Yes! It is a JWT Token! It has username -> **system0**

PAYLOAD: DATA

```
{  
  "username": "system0",  
  "iat": 1711521963  
}
```

VERIFY SIGNATURE

JWT Tokens use a key/secret for encoding, If we get the key we can change the Payload Data, without key, changing the payload will trigger data tempering & It won't work!



We can use the JWT Tool for Brute Forcing the Token with keys from a wordlist, it is a fantastic tool!

You are right, use this tool and the famous / common “rockyou” wordlist & You will get the key!

key is: winniethepooh

PAYLOAD: DATA

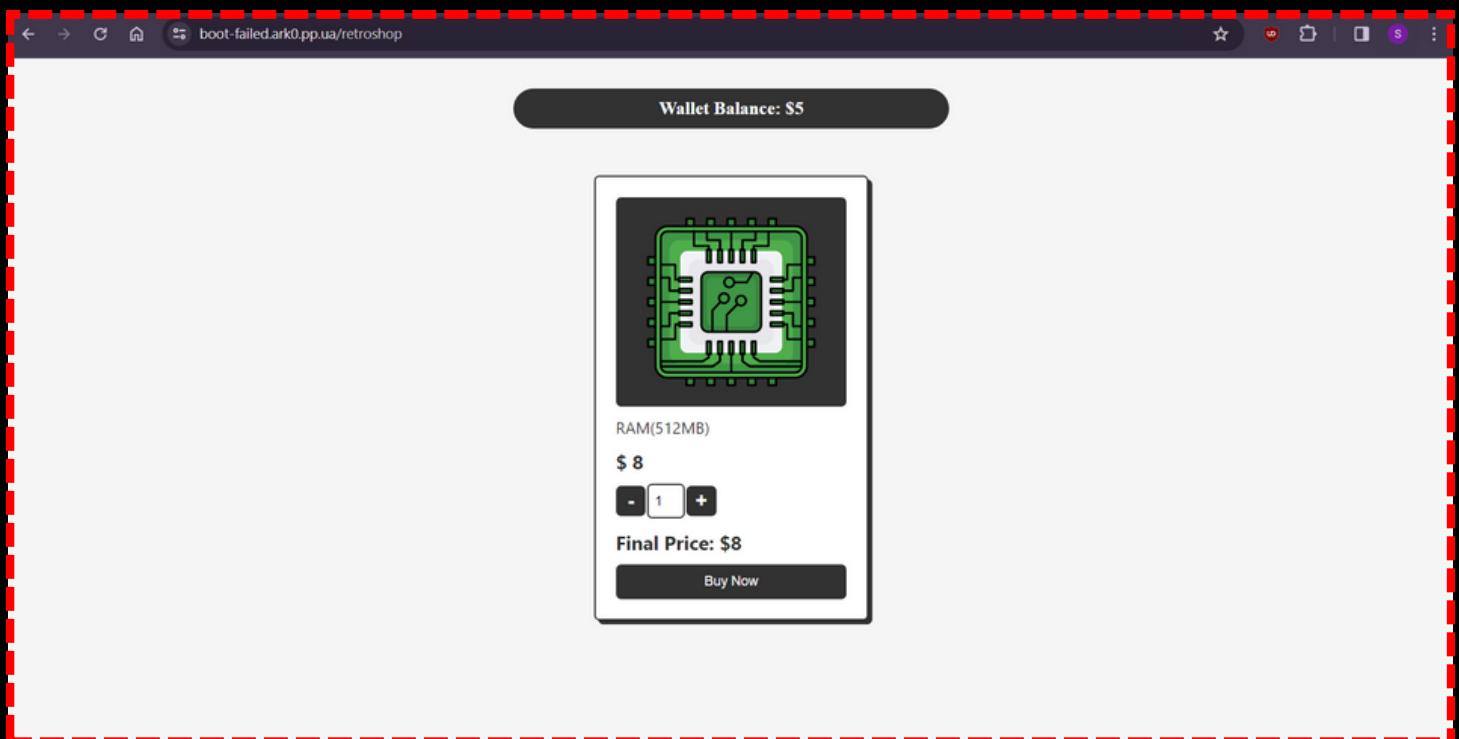
```
{  
    "username": "samarth",  
    "iat": 1711521963  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
    base64UrlEncode(header) + "." +  
    base64UrlEncode(payload),  
    winnieethepooh  
)  secret base64 encoded
```

Put The Key In the Signature Field & Change The Payload Data [**System0** ---> **samarth**]

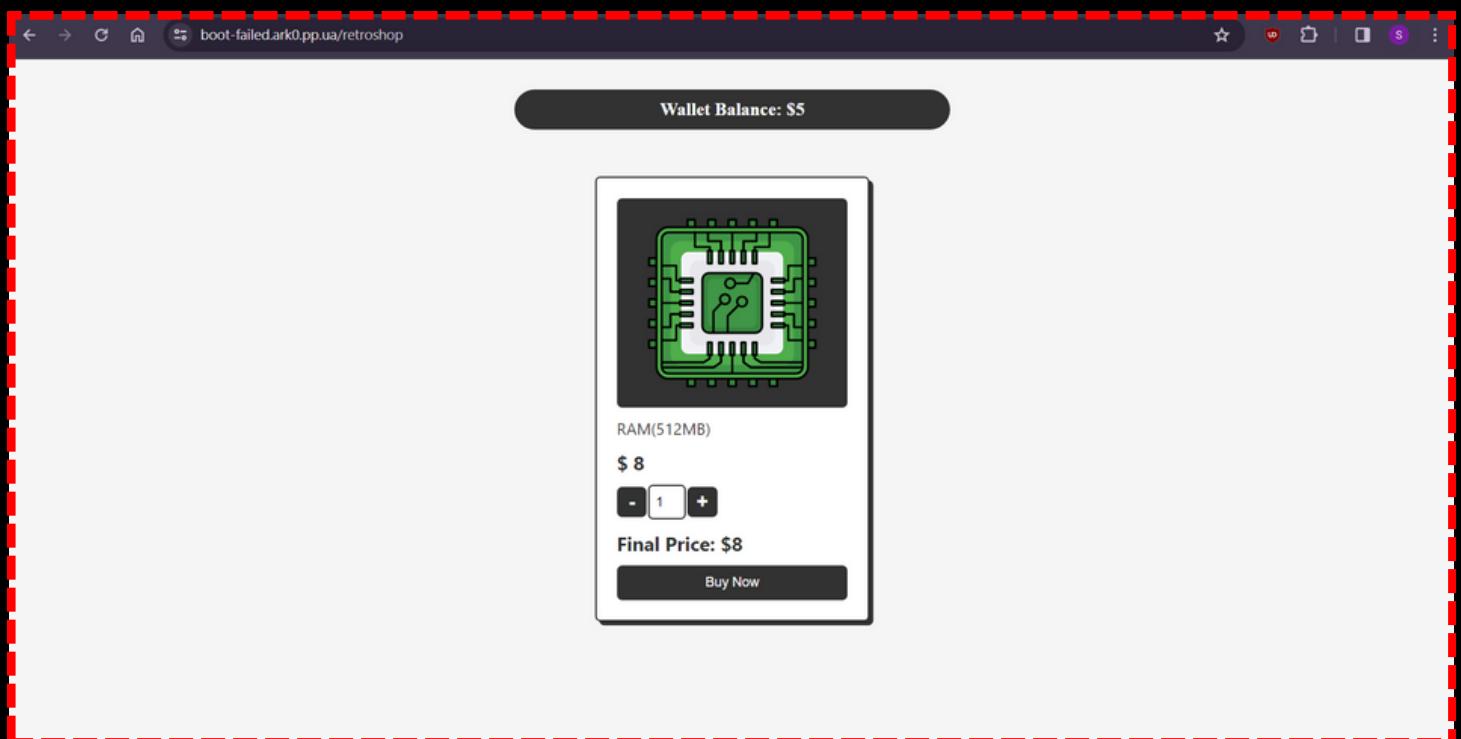
Copy The Encoded Token From Left Window and paste in the token value on the website.



Congrats, You are now logged into the shop!

Now Remeber, You have to upgrade the RAM to total 8GB But If You read the boot logs carefully you will see the system already has 4GB RAM so we will need 4GB More, Thus **8*512 MB = 4GB!**

```
[0.000000] Notice: System is Running out of RAM. Upgrade your memory to 8GB.  
[0.000000] Memory: 4096MB = 4096MB total
```



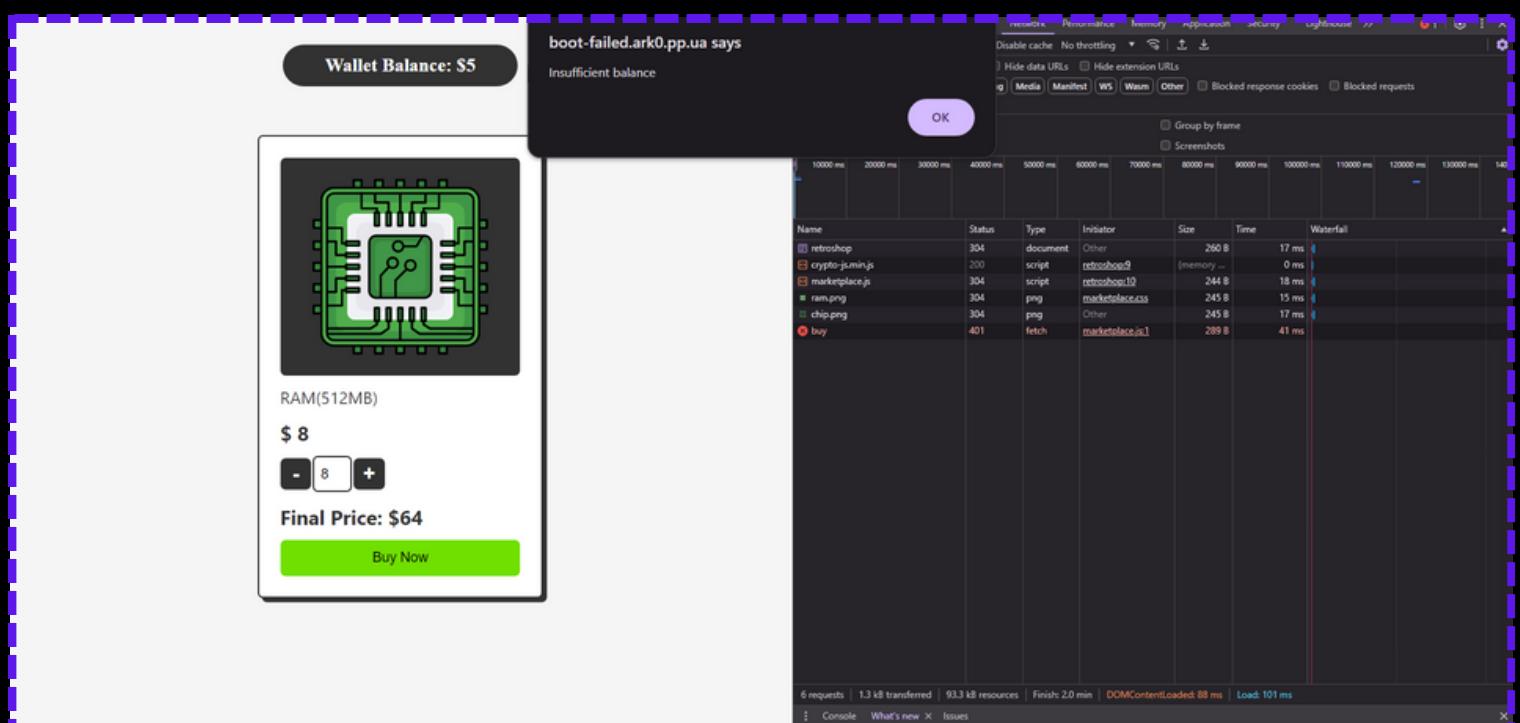
Congrats, You are now logged into the shop!

Now Remeber, You have to upgrade the RAM to total 8GB But If You read the boot logs carefully you will see the system already has 4GB RAM so we will need 4GB More, Thus **8*512 MB = 4GB!**

```
[0.000000] Notice: System is Running out of RAM. Upgrade your memory to 8GB.  
[0.000000] Memory: 4096MB = 4096MB total
```

But There is a catch 

You have wallet balance of 5\$ only!
Thus You will get “insufficient Balance” Error!



You can capture the network packets / activities from network tab in the developer tool.
Here we can see that the application is sending an **Post Request to “/buy” endpoint.**

Payload consist of [amount, price, hash1, hash2]

The screenshot shows the Network tab of a browser's developer tools. A request labeled 'buy' is selected. The 'Payload' tab is active, showing the following JSON object:

```
{amount: 8, price: 64, hash1: "c9f0f895fb98ab9159f51fd0297e236d",...}  
amount: 8  
hash1: "c9f0f895fb98ab9159f51fd0297e236d"  
hash2: "ecbe44e7c_ga245c`7b5_d44`5cbbgf6f_"  
price: 64
```

If we just modify the amount or price field without the respective hashes then we get error “**data tempering detected**”

So we will need to modify the respective hash to match that value; for that we must know a
the hashing type or cipher type.
use any online cipher identifiers
such as [dcode.fr](https://www.dcode.fr/cipher-detection)

We need to modify only the Price and its hash because we have already selected 8 amount of RAM Chips (512MB Each)

Search for a tool

Results

dCode's analyzer suggests to investigate:

- Warning The text has a short length, this can affect the quantity and reliability of the results (see FAQ)
- Warning Few or no significative results (see FAQ)

Rank	Tool
1	ROT-47 Cipher
2	Hexadecimal Data
3	ASCII Code
4	Hexadecimal (Base 16)
5	XOR Cipher

CIPHER IDENTIFIER

Cryptography > Cipher Identifier

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE [?](#)
ecbe44e7c_ga245c`7b5_d44`5cbgf6f_

★ CLUES/KEYWORDS (IF ANY)

SUMMARY

Summary

- Encrypted Message
- What is a cipher identifier? (Definition)
- How to decrypt a cipher
- How to recognize a cipher
- Why does the detector display a warning?
- Why does the analyzer/recognizer identify my cipher method?
- How does the cipher identifier work?

Similar pages

- Index of Coincidence
- Frequency Analysis
- Symbols Cipher List
- Gravity Falls Cipher

dcode.fr: Indicating hash2 is Rot-47 Cipher

You Can Even Use JavaScript Source File Of The Page Which is simply obfuscated, De-obfuscate online using any tool and you get to know how the hashes are made?...

Obfuscator.io Deobfuscator

A tool to undo obfuscation performed by obfuscator.io

Discord GitHub

```

75     'hash1': _0x566947,
76     'hash2': _0x35c26d
77   };
78 v   fetch(_0x2ff6b4(0x166), {
79     'method': 'POST',
80     'headers': {
81       'Content-Type': 'application/json'
82     },
83     'body': JSON[_0x2ff6b4(0x158)](_0x40bebe1)
84 v   )[_0x2ff6b4(0x167)][_0x10876e=>{
85     var _0x42057f = _0x2ff6b4;
86 v     return _0x10876e[_0x42057f(0x164)]()['then'][_0x35b762=>{
87       _0x10876e['ok'] ? alert(_0x35b762) : alert(_0x35b762);
88     }
89   };
90 v   )['catch'][_0xdabaa8=>{
91     var _0x45ed2c = _0x2ff6b4;
92     alert(_0x45ed2c(0x155));
93   }
94 };
95 );
96 }
97 
```

Deobfuscate

```

21   )
22 v   function rotNode(_0x3e9a95) {
23 v     return _0x3e9a95.replace(/([!-~])/g, function (_0x2d0c83) {
24 v       return String.fromCharCode(_0x2d0c83.charCodeAt(0x0) + 0xe) % 0x5e + 0x21);
25   });
26 }
27 const spell = '36cc6f4082acd41f3d05cc1d43387e70';
28 v   function buyNow() {
29   var _0x4b8702 =
30     parseInt(document.getElementById("itemCounter").value);
31   var _0x3a788f =
32     parseFloat(document.getElementById("finalPrice").innerText.replace("Final Price: $", ""));
33   var _0x566947 = CryptoJS.MD5(_0x4b8702.toString()).toString();
34   var _0x35c26d = rotNode(_0x3a788f +
35     '36cc6f4082acd41f3d05cc1d43387e70').toString();
36   var _0x4debe1 = {
37     'amount': _0x4b8702,
38     'price': _0x3a788f,
39     'hash1': _0x566947,
40     'hash2': _0x35c26d
41   };
42 
```

Simplify Expressions
 Simplify Properties
 Simplify Objects
 Remove Proxy Functions

Recover Strings
 Recover Control Flow
 Remove Dead Code

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'boolean'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

6436cc6f4082acd41f3d05cc1d43387e70

ROT-47 Cipher - dCode

Tag(s) : Substitution Cipher, Internet

ROT-47 CIPHER
Cryptography • Substitution Cipher • ROT-47 Cipher

ROT47 DECODER

★ ROT47 CIPHERTEXT ?
ecbe44e7c...ga245c`7b5_d44`5cbbgf6f...

► DECRYPT ROT47

See also: ROT Cipher – ROT-13 Cipher – Caesar Cipher

ROT47 ENCODER

Summary

- ★ ROT47 Decoder
- ★ ROT47 Encoder
- ★ What is Rot-47? (Definition)
- ★ How to encrypt using Rot-47?
- ★ How to decrypt Rot-47 cipher?
- ★ Why the shift of 47?
- ★ How to recognize ROT-47 ciphertext?

Upon Decoding With Rot-47 Cipher

We Get “6436cc6f4082acd41f3d05cc1d43387e70“

Where 64 is the price and later is something like a salt that is constant, You get to know this by analysing multiple hashes for different different price or simply by looking at the JavaScript Code

```

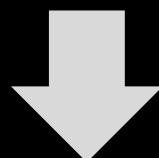
21  }
22 v function rotEnode(_0x3e9a95) {
23 v   return _0x3e9a95.replace(/[^~]/g, function (_0x2d0c83) {
24     return String.fromCharCode((_0x2d0c83.charCodeAt(0x0) + 0xe) %
0x5e + 0x21);
25   });
26 }
27 const spell = '36cc6f4082acd41f3d05cc1d43387e70';
28 v function buyNow() {
29   var _0x4b8702 =
30     parseInt(document.getElementById("itemCounter").value);
31   var _0x3a788f =
32     parseFloat(document.getElementById("finalPrice").innerText.replace("F
inal Price: $", ''));
33   var _0x566947 = CryptoJS.MD5(_0x4b8702.toString()).toString();
34   var _0x35c26d = rotEnode(_0x3a788f +
35     '36cc6f4082acd41f3d05cc1d43387e70').toString();
36   var _0x40ebe1 = {
37     'amount': _0x4b8702,
38     'price': _0x3a788f,
39     'hash1': _0x566947,
40     'hash2': _0x35c26d
41   };
42 }
```

Here you can see
spell is being
appended
in the hash2

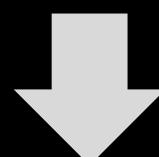
Even, If You don't know about the later part of that hash, modify the initial part of the hash i.e price from \$64 ---> \$1 (anything below 5\$)

So the hash2 must be the rot47 encryption of 1 & the spell/constant part as it is!

i.e **136cc6f4082acd41f3d05cc1d43387e70**



Now Encrypt it again with Rot47 Cipher



Final Hash2: **`be44e7c_ga245c` 7b5_d44` 5cbbgf6f_**

**Now Change The Value of
Price = 1 & The Hash2 =
`be44e7c_ga245c` 7b5_d44` 5cbbgf6f_
& Submit The Post Request to /buy endpoint**

The screenshot shows a Postman interface with a yellow dashed border. At the top, it says "Overview" and "POST https://boot-failed.ark0.pp.ua/buy". Below that is a toolbar with "Save", "Send", and other icons. The main area has tabs for "Params", "Authorization", "Headers (28)", "Body", "Pre-request Script", "Tests", and "Settings". The "Body" tab is selected, showing "raw" JSON input:

```
{"amount":8,"price":1,"hash1":"c9f0f895fb98ab9159f51fd0297e236d","hash2":"`be44e7c_ga245c`7b5_d44`5cbbgf6f_"}
```

Below the body, there's a preview section with tabs for "Pretty", "Raw", "Preview", "Visualize", and "HTML". The "Pretty" tab is selected, showing the response: "VishwaCTF{s3r_y0u_d353rv3_t0_w1n}".

At the bottom, status information is shown: "Status: 200 OK", "Time: 901 ms", "Size: 513 B", and "Save as example".

You will get the flag!!! 

Flag: VishwaCTF{s3r_y0u_d353rv3_t0_w1n}

I HOPE YOU ENJOY THIS CHALLENGE,
YOU CAN CONNECT WITH ME ON MY
[LINKEDIN](#) 

THANK YOU 