**CSEC380 - Principles of Web Application Security**

**Blog Post:RITSEC Demos**

**Professor: Chaim Sanders**

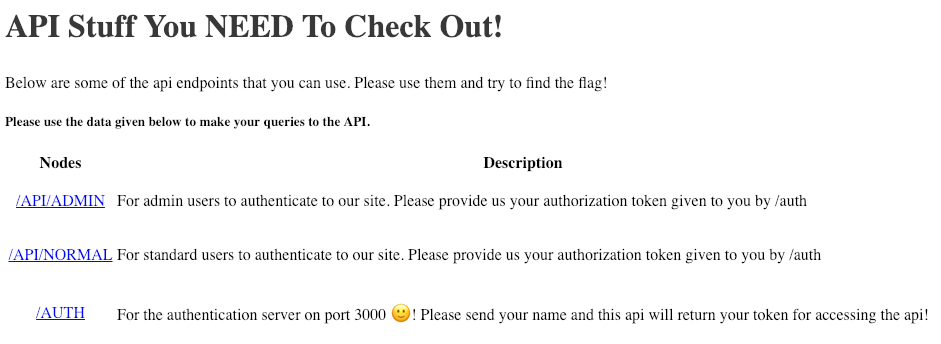
**By Scott Brink**

**April 24th, 2019**

This year I had the privilege of being RITSEC’s tech lead. RITSEC is the security club on RIT’s campus. They have four hour meetings each Friday, starting with a two hour education section. The first hour of the education section is used to teach students the basics of a topic related to security, and the second hour is dedicated to doing a CTF related to the topic that was just taught known as the weekly demos, which are created by the tech lead. I will be analyzing two of the more interesting challenges that were released this year.

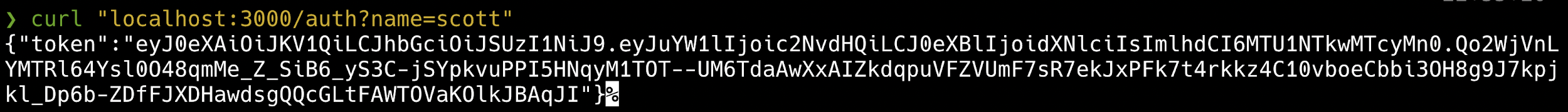
First, I will be analyzing the hard challenge for the week where we covered advanced web. The hint for this challenge was: “Check out our cool api server!” When you first look at the page, you see this description.

**Figure 1: API Server Home Page**



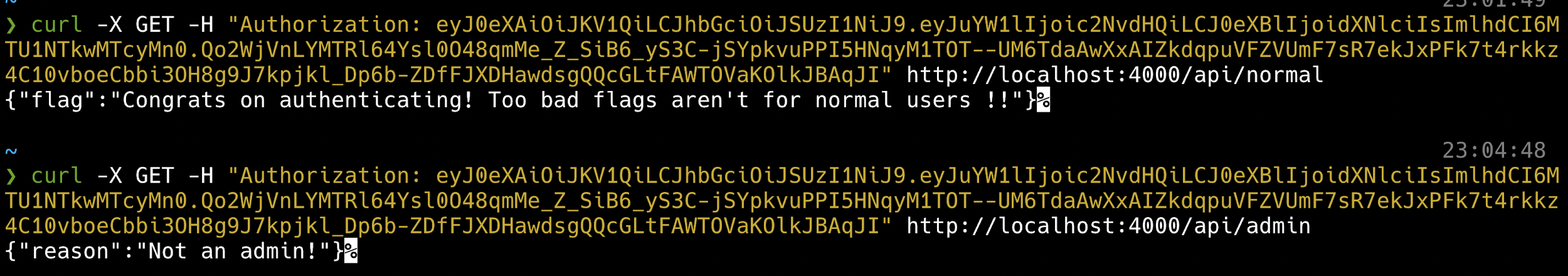
At this point we see that there will be three endpoints to look at, the **/api/norma**l and **/api/admin** pages on the first api server, and the authentication server on port 3000. Let’s attempt to authenticate to the **/auth** endpoint. By send this command **curl “**[**http://localhost:3000/auth?name=scott**](http://localhost:3000/auth?name=scott)**”** we are able to receive our token, shown below

**Figure 2: Received Token**

****

We then try to use this token to authenticate to **/api/normal/** on the api server. It works, however the server tells us that flags are not for normal users, therefore implying that we should attempt to reach **/api/admin/**. An attempt with the same token returns the error “Not an admin!”

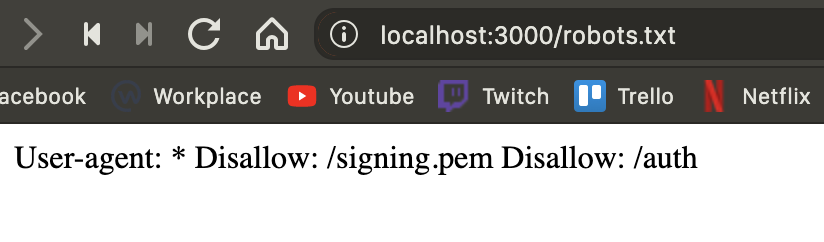
**Figure 3: Authentication attempts**

****

When you examine these tokens closely you can discern that they are JSON Web Tokens (JWT). A quick google search of JWT vulnerabilities leads to an interesting article: <https://auth0.com/blog/critical-vulnerabilities-in-json-web-token-libraries/?fbclid=IwAR19P0F1pXZATMGzw2t8sUxufjoOwd7xZKxQNISDlDLxTmmJgVw4CjMkifE>. This article was written in March of 2015, regarding a critical vulnerability within json web tokens. In short, if a server is using RSA to sign keys and receives a token that is signed with HMAC instead, it will use that the public key used in signing for RSA in the signing process. This can easily be abused if the public key to the server is known.

So now we have some direction. If we are able to get the public key from the authentication server, we may be able to craft ourselves a working token. To do this, I used dirbuster to find the existence of **robots.txt** on the server. When looking at this page I found exactly what I was looking for, the signing key! Upon further inspection, this key was found to be the public key.

**Figure 4: Auth Server’s robots.txt**

****

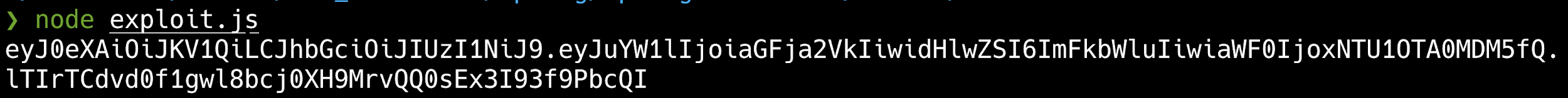
With the existence of a public key, JWT being used, and the previous vulnerability, it is time to exploit this server. The first step is to craft our JWT token in order to send it to the server. The exploit code is shown in figure 5.

**Figure 5: Exploit Code**



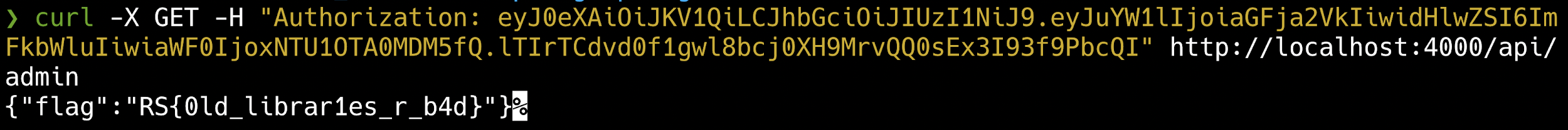
When this code is run, it will generate a token that will be able to authenticate to **/api/admin**.

**Figure 6: Token Generation**



Then, by sending this token to the server in an authorization header, the flag will be given!

**Figure 7: Successful Authentication**

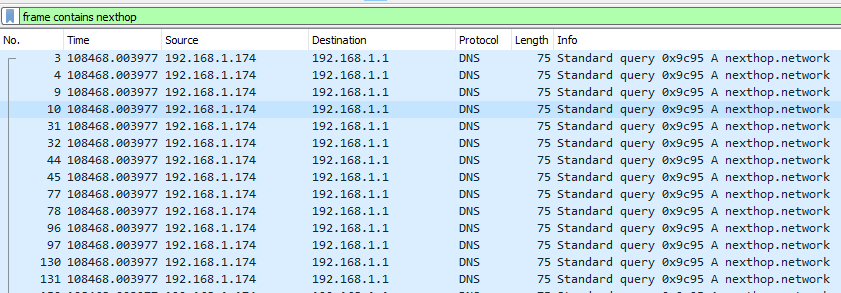
****

The vulnerability within the web server was due to the fact that it was using an outdated version of JWT. The fix for the problem that was demonstrated in this challenge is to update the version of JWT.

Second, I will be analyzing the hard challenge for the week where we covered introduction to networking. This challenge is not as in depth as the last challenge, however it does provide a somewhat real world scenario. The hint for this challenge was: “From our friends at nexthop!” The only information provided is a PCAP, which contains over 10,000 packets.

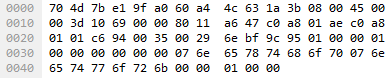
When approaching this challenge, it is important to really focus on the challenge hint. For this challenge, the hint specifically mentioned Nexthop. Using display filters will allow us to gain a extremely helpful hint. When using the display filter of “frame contains nexthop” we receive some interesting information.

**Figure 8: Frame Contains Nexthop**

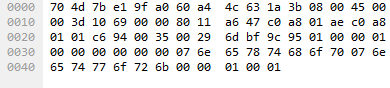
****

When looking at this capture with this display filter, we see hundreds upon hundreds of DNS packets all going to nexthop.network (the networking club’s website). Upon initial inspection, these packets all seem identical. However, when comparing the last byte of each packet, you notice something strange.

**Figure 9: Last Byte of Packet No. 3**

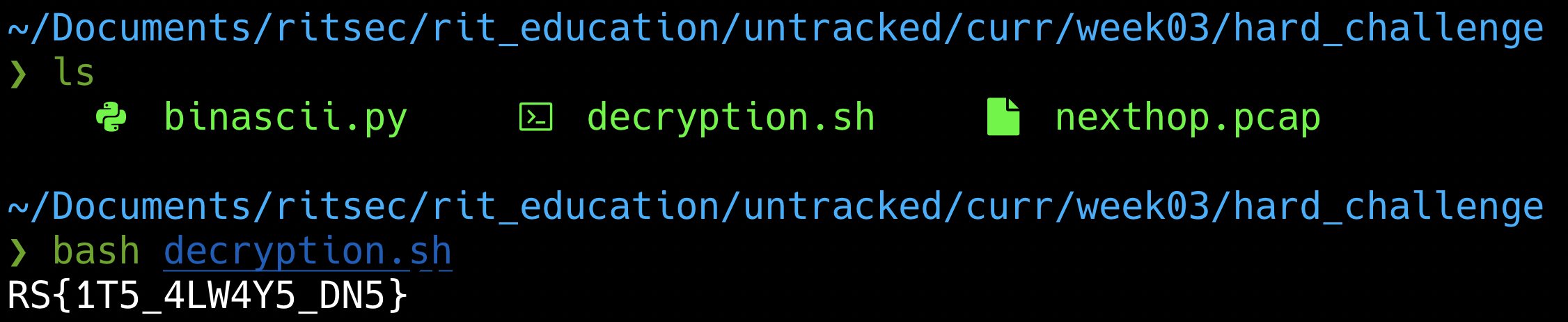


**Figure 10: Last Byte of Packet No. 4**



When scrolling through all the packets, it can be seen that the last byte of these packets alternate with 0s and 1s. By taking these bytes, appending them, and then converting the binary string to ascii, we are able to receive the flag.

**Figure 11: Solution**

****

While this is obviously not a **real** example of a covert channel, this could be a way to detect malicious actors attempting to exfiltrate data from within a company. Very few people are monitoring to make sure each DNS request is legit or not.

These were only two of the challenges out of over one hundred challenges that were created for the RITSEC demos this year. There are some challenges that were relatively easy so that beginners could get them, while some of the challenges that are given get to the level of difficulty explained above. One of the main challenges I have faced while making the weekly demos, is to create a path for the people attempting them. If the challenge does not have a logical path, then it is not a good challenge, as it will force people to guess on where to look in order to solve these challenges.

It has been very fun making challenges for this past year for RITSEC, and I look forward to whoever will be doing this in the future. I will continue to make CTF challenges, as I believe it is not only fun, but a great way to learn the ins and outs of a topic in security.