IDF Cyber Riddle - Sukkot 2020

**Shlomi Avidan**

This CTF like riddle is mainly about Steganography - the art of hiding data in plain sight. In this riddle we are greeted with a video consisting of Shofar horn sounds, which translate to an IP that gives us an image, in which we can find a RAR file guarded by a password. After we crack the password we get another image, whose colors, if represented in hex values and decoded, give us the flag.

Kol Shofar

First we are greeted by a video which looks like this:



We can see that this can be interpreted as Morse Code, where the long sounds are dashes (-), and the short sounds are dots (.). We get the following code:

**.---- ..--- ----. .-.-.- ..--- .---- ...-- .-.-.- ...-- ..--- .-.-.- ..--- -----**

Decoding the morse we get 129.213.32.20, an IP address!

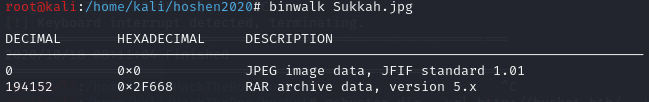
Sukkot.jpg

Typing the IP address into the URL bar redirects us to a website which contains only the following picture:



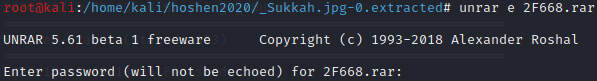
Lets download the picture and check if anything is interesting about it.

If we run binwalk on the image, a tool which checks for files hidden within a file, we will discover that this picture isn’t so innocent and has a RAR archive file hidden within it.



We can extract the RAR by typing **# binwalk -e Sukkah.jpg**

When we try to unrar the file we are greeted by a password protection:

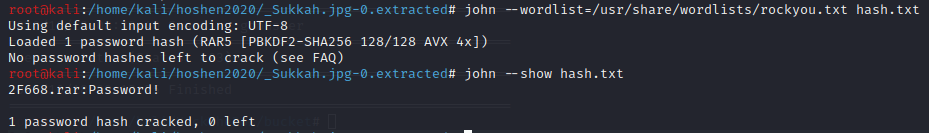


Fortunately for us, we can easily crack the password using a tool called johntheripper.

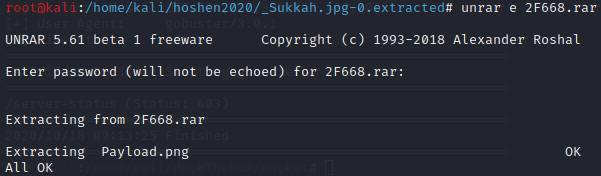
First we will convert the hash of the RAR and save it to the file hash.txt:



Then we can run johntheripper on the hash.txt file. We’ll use the wordlist rockyou.txt:



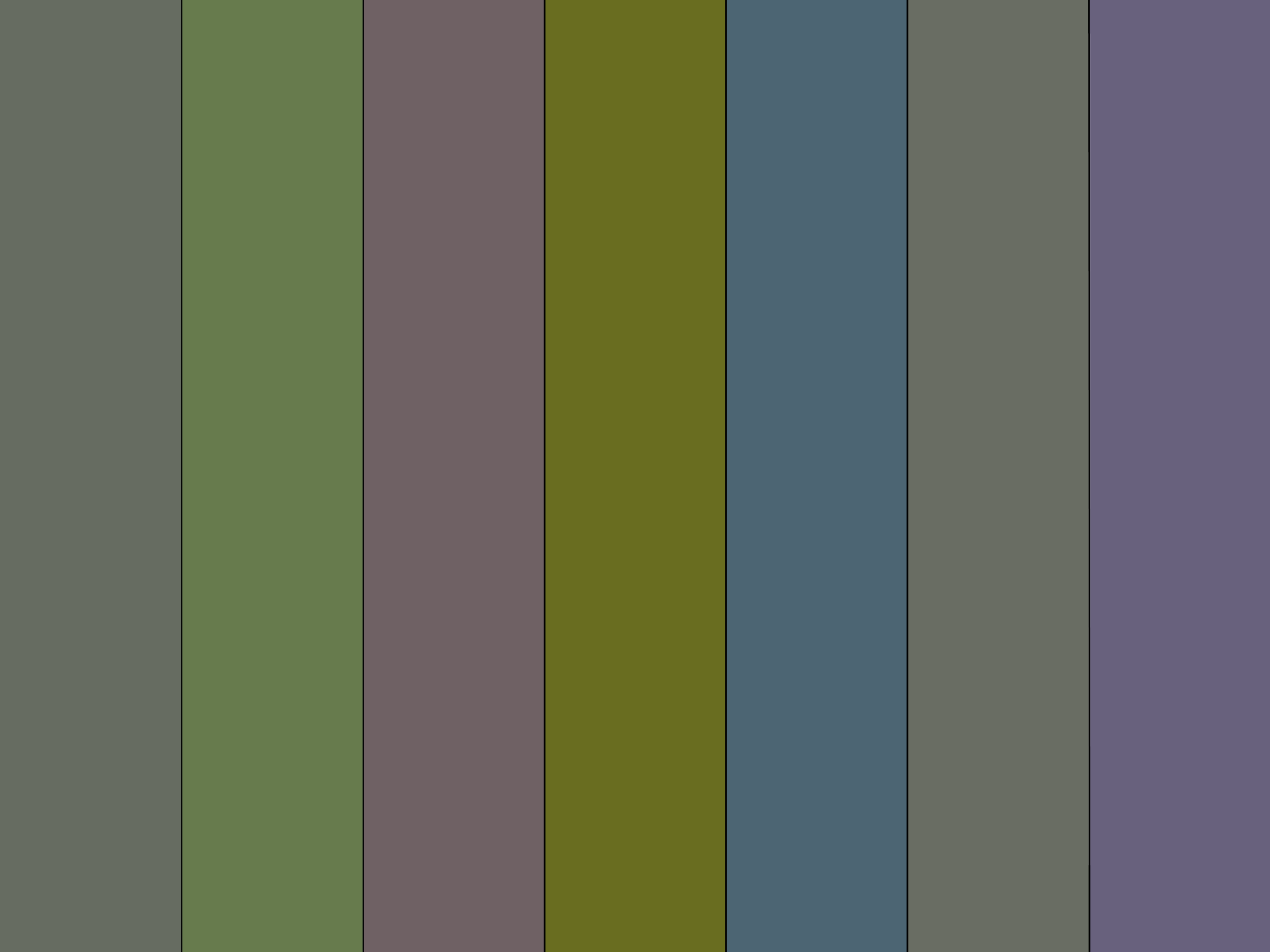
Now we know that the password for the RAR is ‘Password!’. We can use it to unrar the archive:



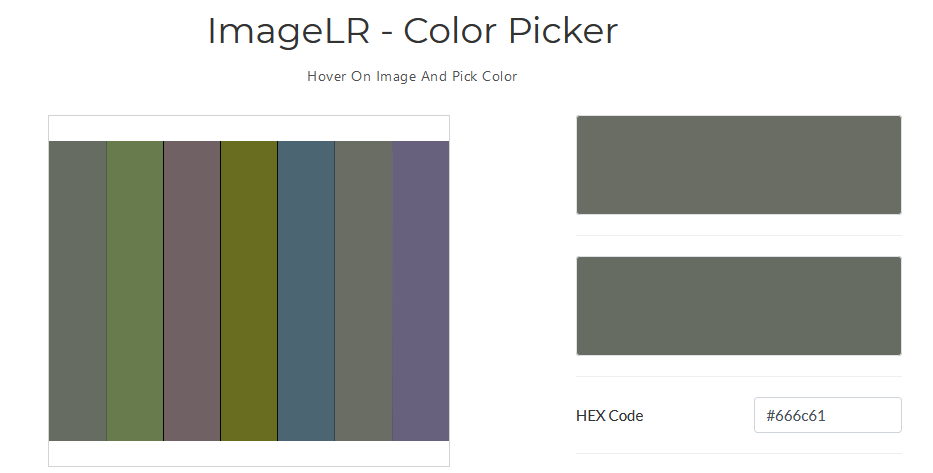
We get the image Payload.png, and now we are one step before revealing our flag.

Colors and numbers

We’ll take a look at Payload.png:



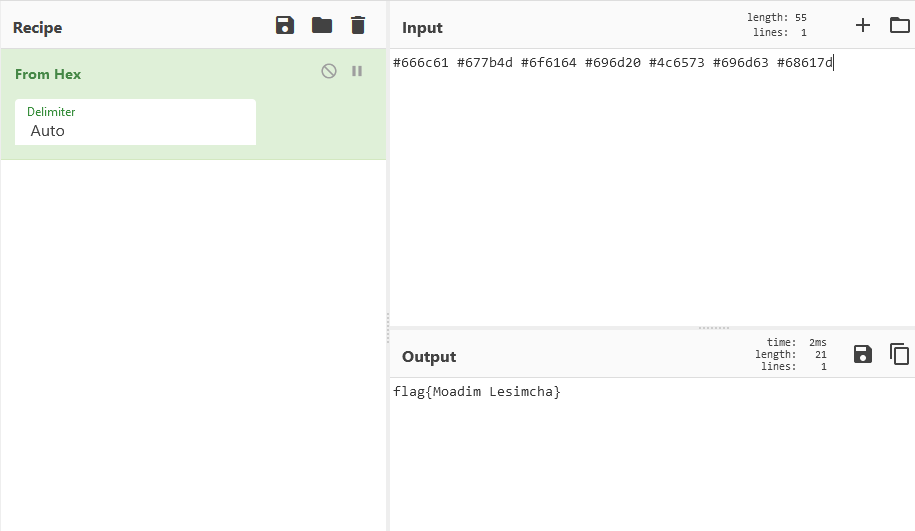
At first sight it doesn’t seem like there is anything special about it. After trying out some stuff on the image I came up with the idea of checking the RGB values of the colors. Using the site <https://imagelr.com> we can see the hex values of the colors:



We’ll do so for all colors and get the following values:

#666c61 #677b4d #6f6164 #696d20 #4c6573 #696d63 #68617d

We can see that all of the values, when split into bytes, are between 0x61-0x7E, this range represents the non-capital letters in the English language, and indeed when we decode these values we get the flag!



**(https://gchq.github.io/CyberChef)**

Final thoughts

This challenge was really nice and easy, and can be a very nice introduction to Stegnography to people who never touched the subject.