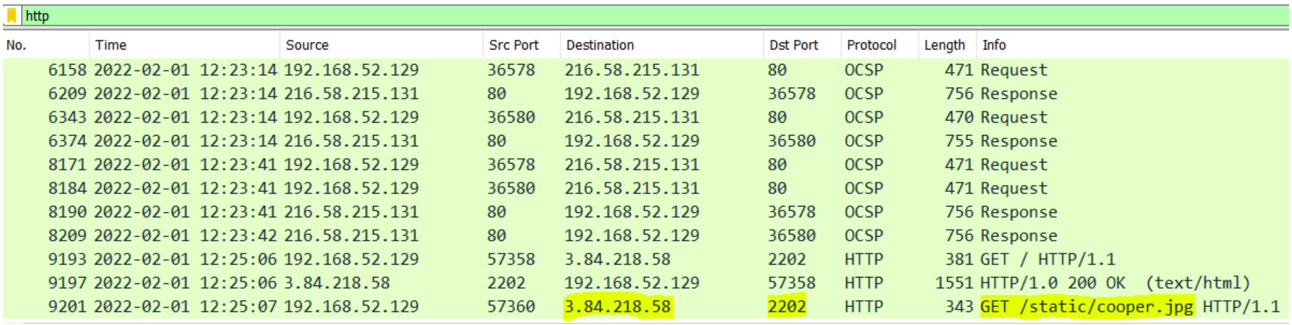
UAM – PCAP & CLOUD SSRF

Analyzing the traffic network

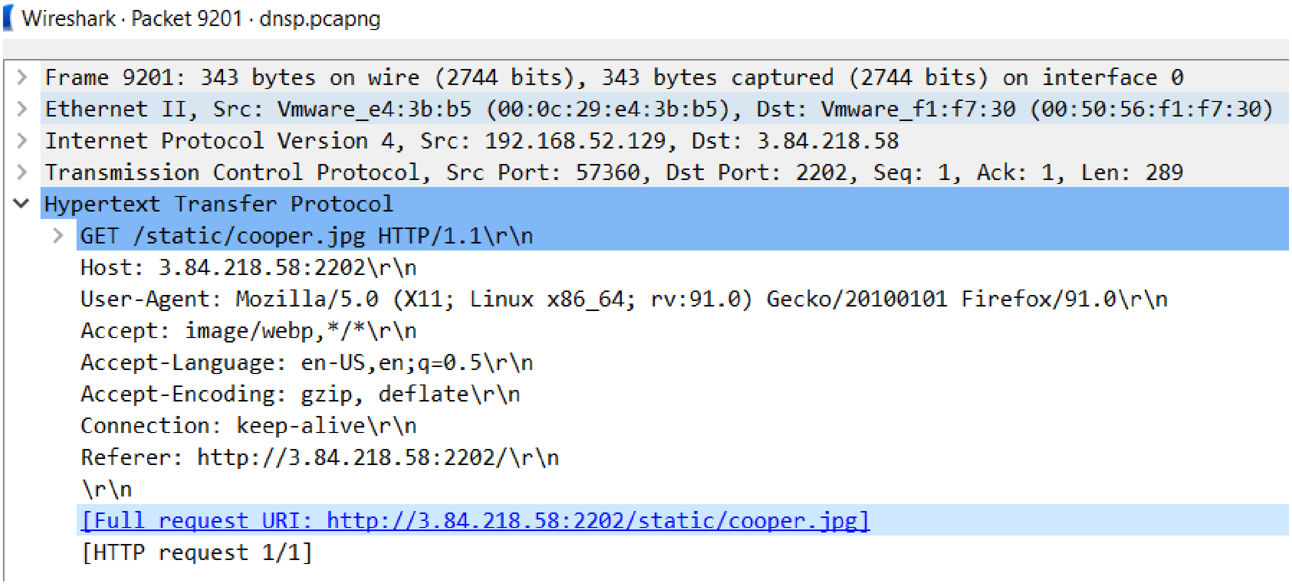
Start point with pcap named dnsp.pcappng as the context of the challenge talks about a fan from Alice Cooper and not talking about his snake that I have found is called Christopher. and the moment of all starts not work when surfing in a website that the user claims to be a malicious one I start analyzing with Wireshark filtering for the HTTP protocol and looking for something related with Alice cooper or Christopher.



http

No.	Time	Source	Src Port	Destination	Dst Port	Protocol	Length	Info
6158	2022-02-01 12:23:14	192.168.52.129	36578	216.58.215.131	80	OCSP	471	Request
6209	2022-02-01 12:23:14	216.58.215.131	80	192.168.52.129	36578	OCSP	756	Response
6343	2022-02-01 12:23:14	192.168.52.129	36580	216.58.215.131	80	OCSP	470	Request
6374	2022-02-01 12:23:14	216.58.215.131	80	192.168.52.129	36580	OCSP	755	Response
8171	2022-02-01 12:23:41	192.168.52.129	36578	216.58.215.131	80	OCSP	471	Request
8184	2022-02-01 12:23:41	192.168.52.129	36580	216.58.215.131	80	OCSP	471	Request
8190	2022-02-01 12:23:41	216.58.215.131	80	192.168.52.129	36578	OCSP	756	Response
8209	2022-02-01 12:23:42	216.58.215.131	80	192.168.52.129	36580	OCSP	756	Response
9193	2022-02-01 12:25:06	192.168.52.129	57358	3.84.218.58	2202	HTTP	381	GET / HTTP/1.1
9197	2022-02-01 12:25:06	3.84.218.58	2202	192.168.52.129	57358	HTTP	1551	HTTP/1.0 200 OK (text/html)
9201	2022-02-01 12:25:07	192.168.52.129	57360	3.84.218.58	2202	HTTP	343	GET /static/cooper.jpg HTTP/1.1

And at the end we see this jpg named cooper related to a destination IP 3.84.218.58. So I look in this packet in detail number 9201 and it shows an interesting website address.

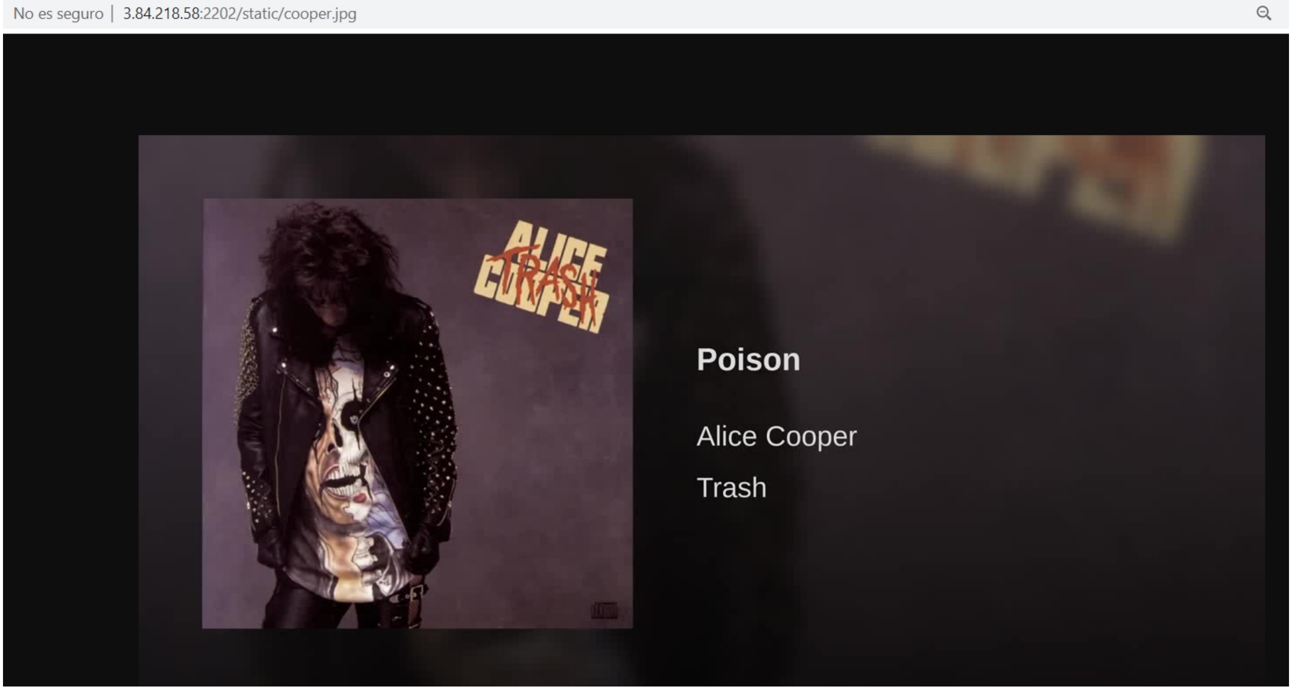


Wireshark · Packet 9201 · dnsp.pcapng

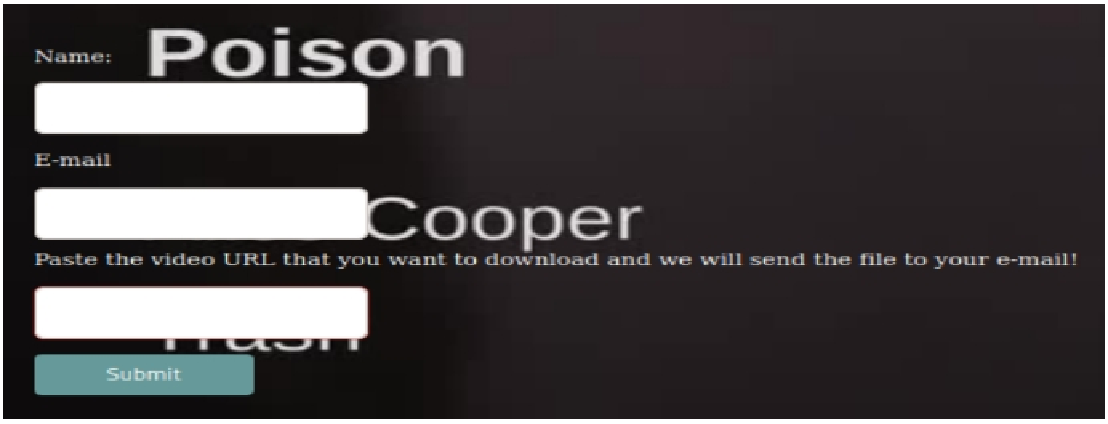
```
> Frame 9201: 343 bytes on wire (2744 bits), 343 bytes captured (2744 bits) on interface 0
> Ethernet II, Src: Vmware_e4:3b:b5 (00:0c:29:e4:3b:b5), Dst: Vmware_f1:f7:30 (00:50:56:f1:f7:30)
> Internet Protocol Version 4, Src: 192.168.52.129, Dst: 3.84.218.58
> Transmission Control Protocol, Src Port: 57360, Dst Port: 2202, Seq: 1, Ack: 1, Len: 289
v Hypertext Transfer Protocol
   > GET /static/cooper.jpg HTTP/1.1\r\n
     Host: 3.84.218.58:2202\r\n
     User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0\r\n
     Accept: image/webp,*/*\r\n
     Accept-Language: en-US,en;q=0.5\r\n
     Accept-Encoding: gzip, deflate\r\n
     Connection: keep-alive\r\n
     Referer: http://3.84.218.58:2202/\r\n
     \r\n
     [Full request URI: http://3.84.218.58:2202/static/cooper.jpg]
     [HTTP request 1/1]
```

CLOUD SSRF

Checking on the web it exists


No es seguro | 3.84.218.58:2202/static/cooper.jpg

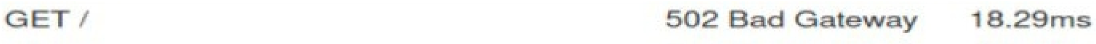
And if I see the root website some interesting boxes appears http://3.84.218.58:2202/


Name:

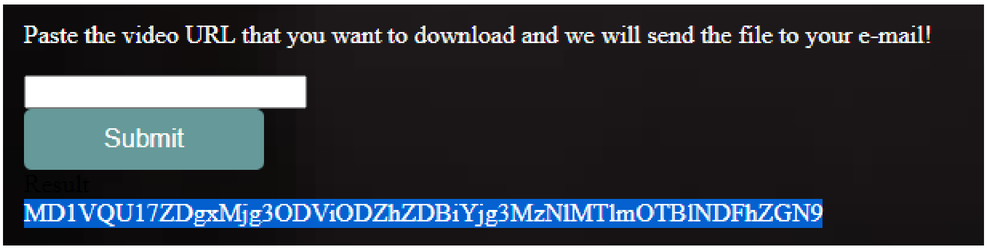
E-mail

Paste the video URL that you want to download and we will send the file to your e-mail!

Paste an URL and we will send the file...Hmm so interesting. That means it is communicating with the server. Let’s try an ngrok URL I see that I received a GET and a base64 is shown below the URL box.



```
GET /                                    502 Bad Gateway     18.29ms
```


Paste the video URL that you want to download and we will send the file to your e-mail!

Result

MD1VQU17ZDgxMjg3ODViODZhZDBiYjg3MzNlMTlmOTBlNDFhZGN9

That behaviour leads me to think about SSRF. I tried with http://127.0.0.1, with localhost, with 0.0.0.0 without exit. Try to search about ports, etc

After a while a clue is shown on the group. A draw about a cloud. Oh a cloud !!! maybe the localhost is not the common 127.0.0.1 but the cloud's one!!! But… ¿Which cloud??

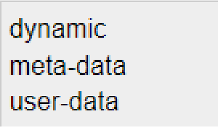
I think about the most famous AWS, Azure, etc. So I start searching how they work as I have never work with cloud.

Info AWS SSRF

Looking if I have some data in the response with the latest metadata:

http://169.254.169.254/latest

and decoding the base64 ZHluYW1pYwptZXRhLWRhdGEKdXNlci1kYXRh shown it returns:



```
dynamic
meta-data
user-data
```

This proofs the cloud service that is behind is AWS. Let's see if there is a iam role associated with the EC2 instance but we have an empty response.

http://169.254.169.254/latest/meta-data/iam

As there is no exist with the common vulnearability I will play with the meta-data information I can get, seen that dynamic and user-data does not retrieve anything else.

http://169.254.169.254/latest/meta-data/

And a very long base64 appears with a menu:

ami-id

ami-launch-index

ami-manifest-path

block-device-mapping/

events/

hibernation/

hostname

identity-credentials/

instance-action

instance-id

instance-life-cycle

instance-type

local-hostname

local-ipv4

mac

metrics/

network/

placement/

profile

public-hostname

public-ipv4

public-keys/

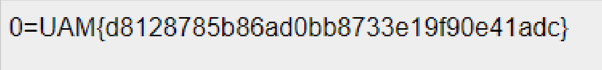
reservation-id

security-groups

services/

I have tried a lot of options here but the one that leaks the appreciated flag is

http://169.254.169.254/latest/meta-data/public-keys



```
0=UAM{d8128785b86ad0bb8733e19f90e41adc}
```

UAM{d8128785b86ad0bb8733e19f90e41adc}

Find me on:

@Ms_Arsenics

@Arsenics