

Starbucks captive portal service Parser

~~CAPPORT Quick Checker~~

Midori & Kir (capport beginner)

Background

M: Hey, IETF organizes a hackathon. I am interested in capport while I am a beginner in capport. Let's do something.

K: Sounds fun!

M: Starbucks provide their captive portal service in JAPAN. I think the design of their captive portal service is similar to what capport draft describes. Dump a log there and let's create a quick capport checker based on the dump!



According to draft-ietf-capport-architecture

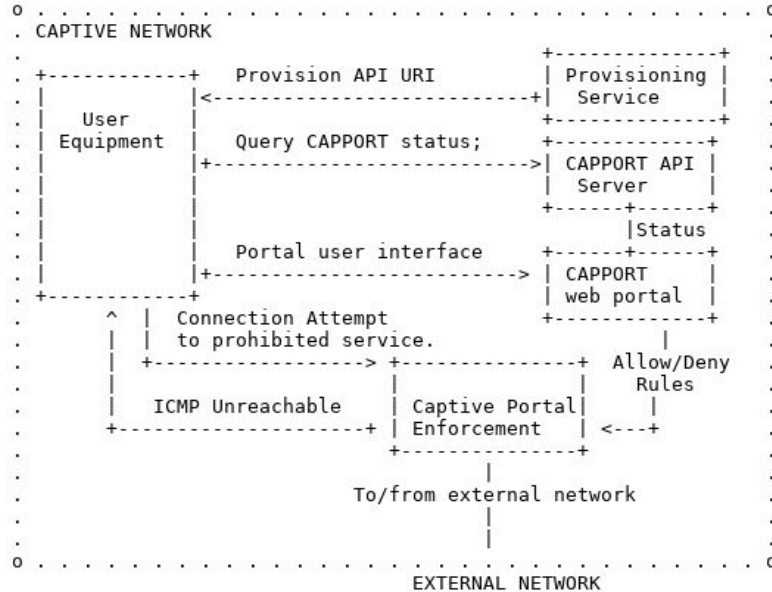


Figure 1: Captive Portal Architecture Component Diagram

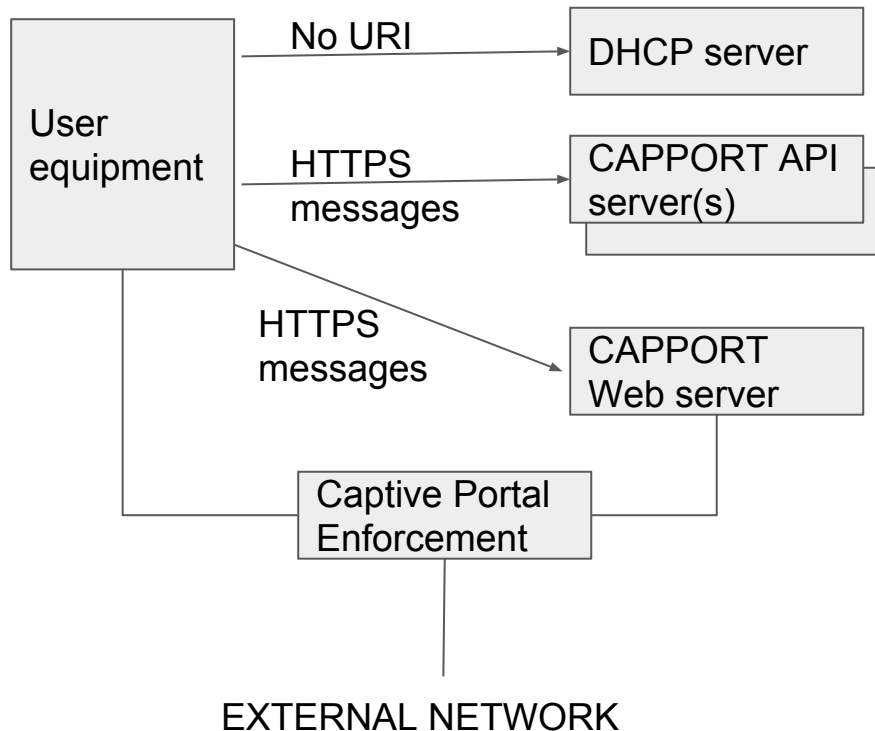
1. Either DHCP OFFER or ACK message includes the URI for the CAPPORT API server
2. HTTP(S) messages appear to connect to the external world
3. HTTPS messages will show up to connect to the portal API server
4. In order to finish the HTTP(S) connection, ICMP will be seen.
5. The status code in HTTP(S) response will be 200 after the web authentication.

:O

- Our dump captured in Starbucks is far from what we assumed
- No URI in DHCP messages
- No ICMP messages



Starbucks captive portal service may be simple



1. DHCP messages include basic information only

2. HTTP(S) messages appear to connect to the external world

3. HTTPS messages will show up to connect to the portal API server

4. The status code in HTTP(S) response will be 200 after the web authentication.

CAPPORT Quick Checker



Starbucks captive portal service

~~Quick Checker~~

Parser

TIMEOUT!



Check items

The image shows the Wireshark network traffic analysis interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. Below the menu is a toolbar with various icons for packet capture and analysis. The main display area is divided into three panes: the top pane shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info; the middle pane shows the details of the selected packet (HTTP); and the bottom pane shows the raw packet data in hexadecimal and ASCII. The selected packet is a GET request to /home/midori/data/ietf10... with a status code of 200.

No.	Time	Source	Destination	Protocol	Length	Info
2792	47.68775300	103.5.140.65	10.27.23.252	TLSv1.2	11010	Application Data
2793	47.68784200	10.27.23.252	103.5.140.65	TCP	66	37437→443 [ACK] Seq=1515 Ack=152247 Win=169472 Len=0 TSval=4294944895 TSecr=
2794	47.69004400	23.44.174.232	10.27.23.252	TCP	66	80→36462 [ACK] Seq=1 Ack=289 Win=926720 Len=0 TSval=2718742287 TSecr=4294944
2795	47.69008900	23.44.174.232	10.27.23.252	TCP	66	[TCP Window Update] 80→36462 [ACK] Seq=1 Ack=289 Win=30048 Len=0 TSval=27187
2796	47.69011100	103.5.140.65	10.27.23.252	TLSv1.2	8274	Application Data
2797	47.69014600	10.27.23.252	103.5.140.65	TCP	66	37437→443 [ACK] Seq=1515 Ack=160455 Win=173568 Len=0 TSval=4294944896 TSecr=
2798	47.69017600	103.5.140.65	10.27.23.252	TCP	2802	[TCP segment of a reassembled PDU]
2799	47.69021700	10.27.23.252	103.5.140.65	TCP	66	37437→443 [ACK] Seq=1515 Ack=163191 Win=170880 Len=0 TSval=4294944896 TSecr=
2800	47.69249000	10.27.23.252	23.44.174.232	HTTP	450	GET /home/midori/data/ietf10... (text/plain)
2801	47.69249000	10.27.23.252	23.44.174.232	TCP	66	36462→80 [ACK] Seq=289 Ack=385 Win=30336 Len=0 TSval=4294944897 TSecr=271874
2802	47.69254100	103.5.140.65	10.27.23.252	TLSv1.2	5538	Application Data

Frame 2800: 450 bytes on wire (3600 bits), 450 bytes captured (3600 bits) on interface 0
Ethernet II, Src: AnubakNet 00:c9:d0 (00:1a:1e:00:c9:d0), Dst: IntelCor 41:ca:ae (64:80:99:41:ca:ae)
Internet Protocol Version 4, Src: 23.44.174.232 (23.44.174.232), Dst: 10.27.23.252
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 36462 (36462), Seq: 1, Ack: 289, Len: 384
Hypertext Transfer Protocol
Line-based text data: text/plain

File: /home/midori/data/ietf10... Packets: 18631 Display

1. DHCP messages include basic information only

2. HTTP(S) messages appear to connect to the external world

3. HTTPS messages will show up to connect to the portal API server

4. The status code in HTTP(S) response will be 200 after the web authentication.

```
"cnt": 1848,  
"content_type": "Change Cipher Spec",  
"protocol": "HTTPS",  
"time": 1510373020.044456  
,  
{  
  "cnt": 2799,  
  "dst": "10.27.23.252",  
  "protocol": "HTTP",  
  "ret_code": 200,  
  "time": 1510373034.409945  
}
```

Conclusion

- Proposed capport architecture and RFC7710 MAY be in deployment stage in JAPAN
- Fun to guess the architecture of the captive portal service from dump
- Martin, thank you so much for supporting us to work on capport hackethon on remote