Lab 6 Report

By Julian Flum and Miftahul Huq

Activity 1

Q1. Why does password complexity specifically matter in Wi-Fi?

Because this password will be passed through wireless signals for authentication, and it can possibly get grabbed via a packet sniffer or other third party. As a result, it needs to be properly complex so it's harder to crack if it's encrypted. Simpler passwords will be easier to decrypt.

Q2. Indicate two new features this AP supports.

- This AP now supports 802.11ax for Wifi6. (This is one of the latest versions of Wifi, which extends the bandwidth to the 6 GHz Band)
- It now has MIMO Technology. (This means there's multiple inputs, multiple outputs, which allows for more clients to connect at one time.)

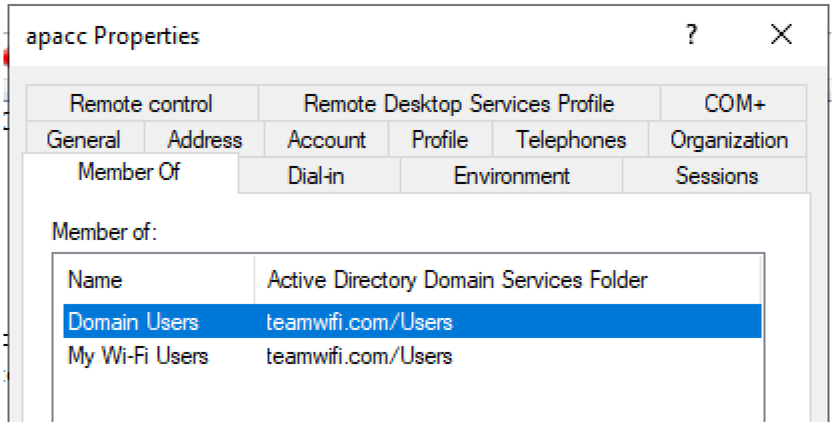
Q3. If you have an 802.11ax-enabled device, use Google's speedtest to compare the speed you get through an 802.11ax connection and through an 802.11ac connection (e.g., RIT network).

802.11ax
Download: 521.3 Mbps
Upload: 396.6 Mbps

802.11ac
Download: 283.2 Mbps
Upload: 150.6 Mbps

This is because 802.11ax is able to take advantage of more frequencies, making it able to transmit more data more quickly than 802.11ac

Activity 2

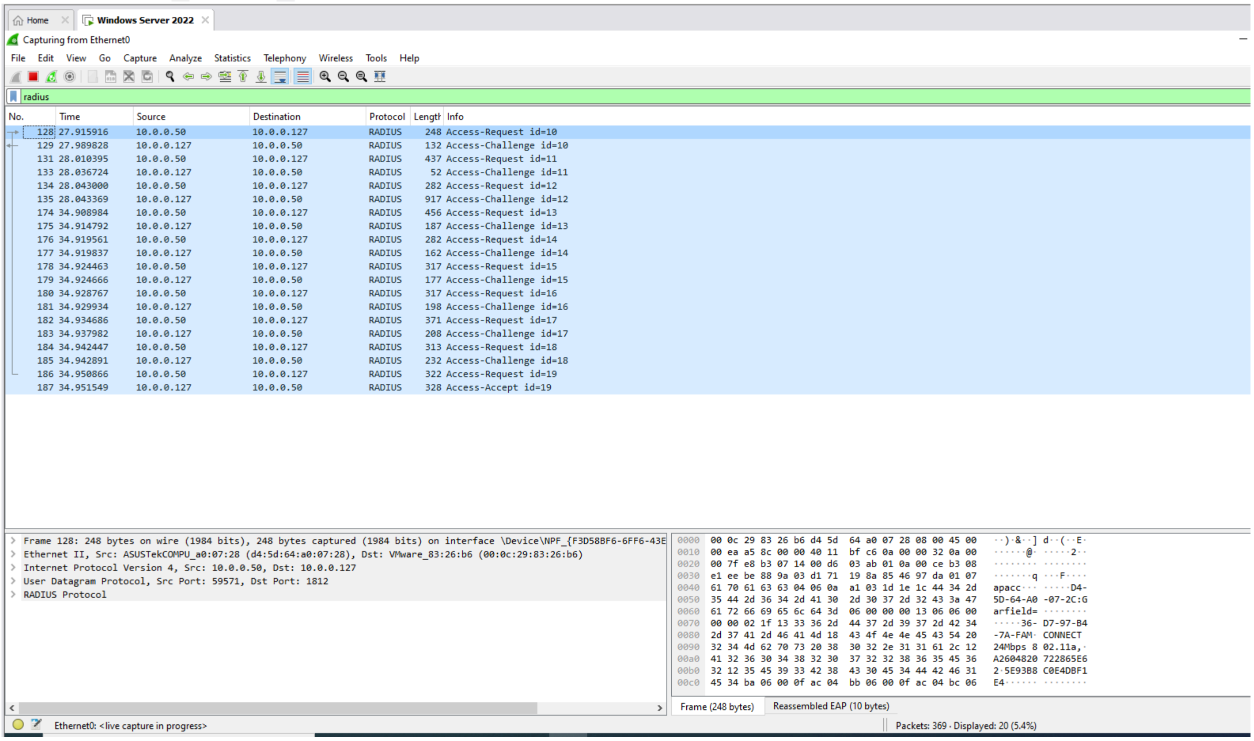


Member of:

Name	Active Directory Domain Services Folder
Domain Users	teamwifi.com/Users
My Wi-Fi Users	teamwifi.com/Users

Q4. Does the user need to be a RADIUS client too? Explain.

They do not. Since the AP is set as a RADIUS Client, and the user is communicating with the AP as a middleman, the AP handles the RADIUS communication itself.



```
No.    Time       Source      Destination  Protocol Length Info
   128 27.915916  10.0.0.50   10.0.0.127   RADIUS    248 Access-Request id=10
   129 27.989828  10.0.0.127  10.0.0.50    RADIUS    132 Access-Challenge id=10
   131 28.010395  10.0.0.50   10.0.0.127   RADIUS    437 Access-Request id=11
   133 28.036724  10.0.0.127  10.0.0.50    RADIUS     52 Access-Challenge id=11
   134 28.043000  10.0.0.50   10.0.0.127   RADIUS    282 Access-Request id=12
   135 28.043369  10.0.0.127  10.0.0.50    RADIUS    917 Access-Challenge id=12
   174 34.908984  10.0.0.50   10.0.0.127   RADIUS    456 Access-Request id=13
   175 34.914792  10.0.0.127  10.0.0.50    RADIUS    187 Access-Challenge id=13
   176 34.919561  10.0.0.50   10.0.0.127   RADIUS    282 Access-Request id=14
   177 34.919837  10.0.0.127  10.0.0.50    RADIUS    162 Access-Challenge id=14
   178 34.924463  10.0.0.50   10.0.0.127   RADIUS    317 Access-Request id=15
   179 34.924666  10.0.0.127  10.0.0.50    RADIUS    177 Access-Challenge id=15
   180 34.928767  10.0.0.50   10.0.0.127   RADIUS    317 Access-Request id=16
   181 34.929934  10.0.0.127  10.0.0.50    RADIUS    198 Access-Challenge id=16
   182 34.934686  10.0.0.50   10.0.0.127   RADIUS    371 Access-Request id=17
   183 34.937982  10.0.0.127  10.0.0.50    RADIUS    208 Access-Challenge id=17
   184 34.942447  10.0.0.50   10.0.0.127   RADIUS    313 Access-Request id=18
   185 34.942891  10.0.0.127  10.0.0.50    RADIUS    232 Access-Challenge id=18
   186 34.950866  10.0.0.50   10.0.0.127   RADIUS    322 Access-Request id=19
   187 34.951549  10.0.0.127  10.0.0.50    RADIUS    328 Access-Accept id=19
```

```
> Frame 128: 248 bytes on wire (1984 bits), 248 bytes captured (1984 bits) on interface \Device\NPF_{F3D58BF6-6FF6-43E
> Ethernet II, Src: ASUSTekCOMPU_a0:07:28 (d4:5d:64:a0:07:28), Dst: VMware_83:26:b6 (00:0c:29:83:26:b6)
> Internet Protocol Version 4, Src: 10.0.0.50, Dst: 10.0.0.127
> User Datagram Protocol, Src Port: 59571, Dst Port: 1812
> RADIUS Protocol
```

```
0000  00 0c 29 83 26 b6 d4 5d  64 a0 07 28 08 00 45 00   ··)·&··] d··(··E·
0010  00 ea a5 8c 00 00 40 11  bf c6 0a 00 00 32 0a 00   ······@· ·····2··
0020  00 7f e8 b3 07 14 00 d6  03 ab 01 0a 00 ce b3 08   ········ ········
0030  e1 ee be 88 9a 03 d1 71  19 8a 85 46 97 da 01 07   ·······q ···F····
0040  61 70 61 63 63 04 06 0a  a1 03 1d 1e 1c 44 34 2d   apacc··· ·····D4-
0050  35 44 2d 36 34 2d 41 30  2d 30 37 2d 32 43 3a 47   5D-64-A0 -07-2C:G
0060  61 72 66 69 65 6c 64 3d  06 00 00 00 13 06 06 00   arfield= ········
0070  00 00 02 1f 13 33 36 2d  44 37 2d 39 37 2d 42 34   ·····36- D7-97-B4
0080  2d 37 41 2d 46 41 4d 18  43 4f 4e 4e 45 43 54 20   -7A-FAM· CONNECT 
0090  32 34 4d 62 70 73 20 38  30 32 2e 31 31 61 2c 12   24Mbps 8 02.11a,·
00a0  41 32 36 30 34 38 32 30  37 32 32 38 36 35 45 36   A2604820 722865E6
00b0  32 12 35 45 39 33 42 38  43 30 45 34 44 42 46 31   2·5E93B8 C0E4DBF1
00c0  45 34 ba 06 00 0f ac 04  bb 06 00 0f ac 04 bc 06   E4······ ········
```

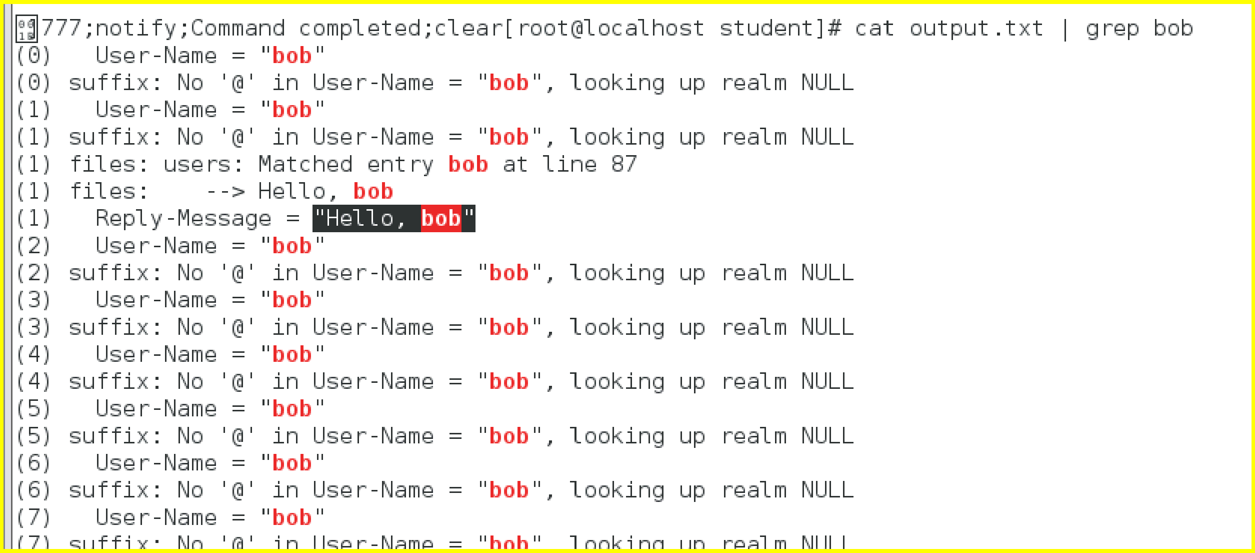
Ethernet0: <live capture in progress>

Packets: 369 · Displayed: 20 (5.4%)

Activity 3

Q5. On the authentication server, how can you tell Bob was able to successfully authenticate? Include and explain a screenshot from either Wireshark or the radiusd terminal log to support your answer.

We can tell, since when Bob connects, a certain message plays, namely “Hello, bob”, which is meant to signal that they’ve connected. This would not occur unless authentication was successful.



```
777;notify;Command completed;clear[root@localhost student]# cat output.txt | grep bob
(0)   User-Name = "bob"
(0) suffix: No '@' in User-Name = "bob", looking up realm NULL
(1)   User-Name = "bob"
(1) suffix: No '@' in User-Name = "bob", looking up realm NULL
(1) files: users: Matched entry bob at line 87
(1) files:     --> Hello, bob
(1)   Reply-Message = "Hello, bob"
(2)   User-Name = "bob"
(2) suffix: No '@' in User-Name = "bob", looking up realm NULL
(3)   User-Name = "bob"
(3) suffix: No '@' in User-Name = "bob", looking up realm NULL
(4)   User-Name = "bob"
(4) suffix: No '@' in User-Name = "bob", looking up realm NULL
(5)   User-Name = "bob"
(5) suffix: No '@' in User-Name = "bob", looking up realm NULL
(6)   User-Name = "bob"
(6) suffix: No '@' in User-Name = "bob", looking up realm NULL
(7)   User-Name = "bob"
(7) suffix: No '@' in User-Name = "bob", looking up realm NULL
```

Activity 4

Q6. Would you prefer EAP-TLS over MSCHAPv2 as the inner authentication in PEAP? Please explain your answer.

EAP-TLS is the preferred inner authentication, because where MSCHAPv2 is one-way certificate, EAP-TLS is two-way certificate, which is more secure, and prevents a compromised client password from single handedly resulting in unauthorized access.

SSID: Garfield
wireless security: lasagna123

Router Login: GarfieldCat
Password: jon_arbuckle