Wifi Security Final Project (CS 460)

Team Members:
Frederick Wirjo (wirjo2)
TianXing Dun (dun2)

Objective

The objective of our project is to conduct a survey of wireless access points around campus and determine the security of that wireless network based on whether or not data was encrypted.

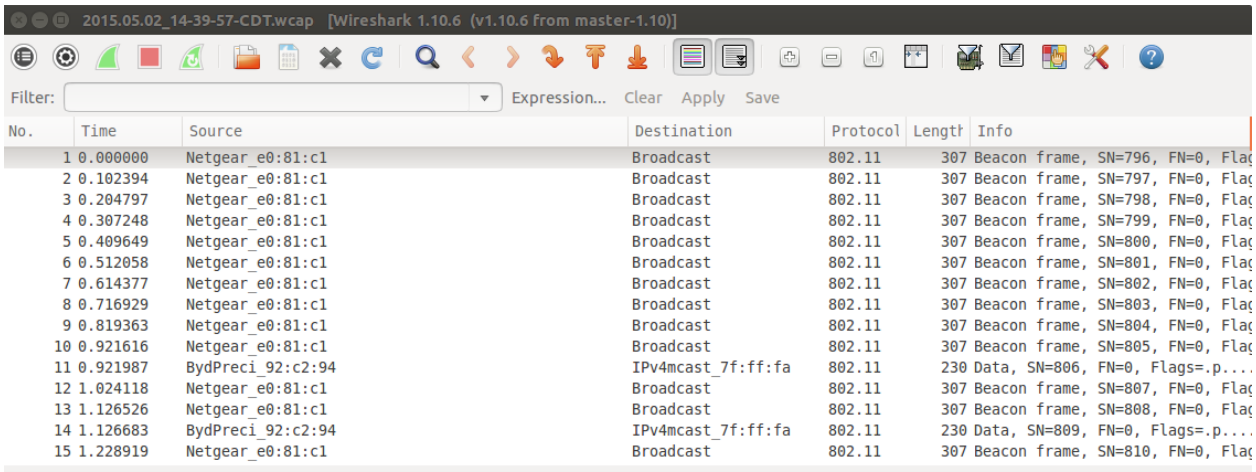
Method

To accomplish this, we analyzed network traffic of commercial and university networks by using the Wireless Diagnostics tool on Mac OS X to generate .wcap files and then opened these files in Wireshark to see what information we can obtain. While on a network, one person was responsible for using the Wireless Diagnostics tool for capturing network traffic while another person was responsible for using another computer (computer X) to browse the internet. When reading the .wcap files, we would look for the IP address of specific computer, look for any TCP streams, and determine if data is passed in plain text.

Public AP's tested:
Starbucks, McDonalds, UIUCNet, IllinoisNet, Espresso Royale, Sushi Ichiban, Mia Zas

Public AP at Sushi Ichiban

The public AP at sushi ichiban requires a password to login. After analyzing the files we captured, we were not able to capture any tcp traffic from the network. A screenshot of the file is shown below.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=796, FN=0, Flag
2	0.102394	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=797, FN=0, Flag
3	0.204797	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=798, FN=0, Flag
4	0.307248	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=799, FN=0, Flag
5	0.409649	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=800, FN=0, Flag
6	0.512058	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=801, FN=0, Flag
7	0.614377	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=802, FN=0, Flag
8	0.716929	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=803, FN=0, Flag
9	0.819363	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=804, FN=0, Flag
10	0.921616	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=805, FN=0, Flag
11	0.921987	BydPreci_92:c2:94	IPv4mcast_7f:ff:fa	802.11	230	Data, SN=806, FN=0, Flags=.p....
12	1.024118	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=807, FN=0, Flag
13	1.126526	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=808, FN=0, Flag
14	1.126683	BydPreci_92:c2:94	IPv4mcast_7f:ff:fa	802.11	230	Data, SN=809, FN=0, Flags=.p....
15	1.228919	Netgear_e0:81:c1	Broadcast	802.11	307	Beacon frame, SN=810, FN=0, Flag

The only information we could obtain is broadcast traffic using 802.11 protocol. This means that this AP (access point) is secure.

The following screenshot shows the security protocol of the wifi at sushi ichiban. Notice that the wifi security protocol at Sushi Ichiban is WPA2 instead of WPA. This is considered more secure than WPA because WPA2 uses AES encryption algorithm and 802.1x-based authentication. Both of these mechanisms ensure that only authorized devices are allowed access to the internet, making sniffing impossible. In conclusion,

the public AP at sushi ichiban is secure to use.

```
Terminal  Shell  Edit  View  Window  Help                                   15%  Sat 2:38 PM
fwirjo — bash — 156×40
        status: inactive
p2p0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2304
        ether 0e:40:08:bc:dc:58
        media: autoselect
        status: inactive
awdl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1452
        ether 9a:8b:f5:b0:2b:ad
        inet6 fe80::988b:f5ff:feb0:2bad%awdl0 prefixlen 64 scopeid 0x9
        nd6 options=1<PERFORMNUD>
        media: autoselect
        status: active
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=63<RXCSUM,TXCSUM,TSO4,TSO6>
        ether 6e:40:08:cb:95:00
        Configuration:
                id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
                maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
                root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
                ipfilter disabled flags 0x2
        member: en1 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 5 priority 0 path cost 0
        member: en2 flags=3<LEARNING,DISCOVER>
                ifmaxaddr 0 port 6 priority 0 path cost 0
        nd6 options=1<PERFORMNUD>
        media: <unknown type>
        status: inactive
isr5860:~ fwirjo$ airport -s
                            SSID BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
                          HOME-7B02 00:1d:d2:49:7b:00 -79  11,-1   Y  US WPA(PSK/TKIP,AES/TKIP) WPA2(PSK/TKIP,AES/TKIP)
                       sushiichiban e0:91:f5:e0:81:c2 -45  11      N  -- WPA2(PSK/AES/AES)
                         Cisco25065 48:f8:b3:33:10:e1 -48  11      Y  -- WPA(PSK/AES,TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
                          HOME-91AE 64:66:b3:94:b3:2a -82  5,+1    Y  -- WPA2(PSK/AES/AES)
                        xfinitywifi 46:70:09:05:1c:00 -69  6       Y  US NONE
                          HOME-1C02 40:70:09:05:1c:00 -68  6       Y  US WPA(PSK/TKIP,AES/TKIP) WPA2(PSK/TKIP,AES/TKIP)
                        xfinitywifi 5a:23:8c:ae:91:a1 -73  1       Y  -- NONE
                        xfinitywifi 5a:23:8c:ae:84:a9 -86  1       Y  -- NONE
HogwartsSkoolWitchcraftNWizardry 58:23:8c:ae:84:a7 -86  1       Y  -- WPA(PSK/AES,TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
                         Cisco25065 48:f8:b3:33:10:e2 -58  149,+1  Y  -- WPA(PSK/AES,TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
                       sushiichiban e0:91:f5:e0:81:c1 -59  44,+1   Y  -- WPA2(PSK/AES/AES)
isr5860:~ fwirjo$
```

Public AP at Mcdonalds

Logging in to the public wireless network at McDonalds was trivial. To log on to the network, simply click a connect button in the browser. By analyzing the files we captured, we were able to obtain internet traffic data from other users on the same network. In this setup, one team member was responsible for browsing the internet with one device while another team member was responsible for using the Wireless Diagnostics sniffer tool. As you can see in the screenshot below, we were able to get the user's device(IPAD) as well as the website they're connected to (http://www.etonline.com). We were also able to determine what information is being transferred. In the connection shown in screenshot 1, the file being transferred is a text file named f.txt. In the connection shown in screenshot 2, we were able to find that the file being transferred is a Image/gif file. Clearly, the connection information is passed in plain text and is broadcasted over the internet. We conclude that AP at Mcdonalds is insecure.

Screenshot 1

Stream Content

```
GET /pagead/expansion_embed.js?source=safeframe HTTP/1.1
Host: pagead2.googlesyndication.com
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (iPad; CPU OS 8_1_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4
Accept-Language: en-us
Referer: http://www.etonline.com/
news/163808_what_feud_jennette_mccurdy_and_ariana_grande_are_friends_we_text_probably_once
_a_week/
Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK
P3P: policyref="http://www.googleadservices.com/pagead/p3p.xml", CP="NOI DEV PSA PSD IVA
IVD OTP OUR OTR IND OTC"
Content-Type: text/javascript; charset=UTF-8
ETag: 4778157287651931086
Date: Sat, 02 May 2015 20:07:38 GMT
Expires: Sat, 02 May 2015 21:07:38 GMT
X-Content-Type-Options: nosniff
Content-Disposition: attachment; filename="f.txt"
Content-Encoding: gzip
Server: cafe
```

screenshot 2

Stream Content

```
%2bj3VVJ6RiEq4s4V04zlvDm1C49R0ynHYQ7uAzH1JoEi%2b7mdk8a8kk1%2bZJhvkFuW%2bN3dwLTkQ%
2ffyJUgFmSAOJ2t9aiaC9qcjfXios3xyhSK3tThX5ZKE3O4k7WDCu1%
2b6qIxRWwrl41L6ifNFd6lidMGnjUzkhbWUmxcFEAjwGDQ6c37iZD5uxiUEfd8l5zbYIirMZucg65APdEtZil2fU7y
lJLg%3d
User-Agent: Mozilla/5.0 (iPad; CPU OS 8_1_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive

[658 bytes missing in capture file]HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: image/gif
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Server: Microsoft-IIS/7.5
P3P: CP="NON DSP COR CURa PSA PSD OUR BUS NAV STA"
X-Powered-By: ASP.NET
Date: Sat, 02 May 2015 20:41:51 GMT
Content-Length: 43

GIF89a.............!.......,...........D..;
```

Public AP at Mia Zas

There were two wireless networks at Mia Zas. One was labeled “Zas1” and the other was labeled “Zas2”. Both of these networks do not have wifi passwords. By analyzing the .wcap file while on the Zas1 network, we were able to discover that that there was a DNS query for www.yahoo.com from the corresponding IP address that my partner was using (IP address == 192.168.99.34). In addition, we found a TCP stream that showed exactly what website my partner was on (www.cargurus.com). It also shows what device my partner was using (an iPad), next to the User-Agent line. Two screenshots of this TCP stream are shown below. Because this data is in plain text, we conclude that the wireless network for Zas1 is insecure. On both Zas1 and Zas2 network, we were able to capture plain HTTP data from different ip addresses and view what websites others were looking at.

Screenshot 1

Follow TCP Stream (tcp.stream eq 55)

Stream Content

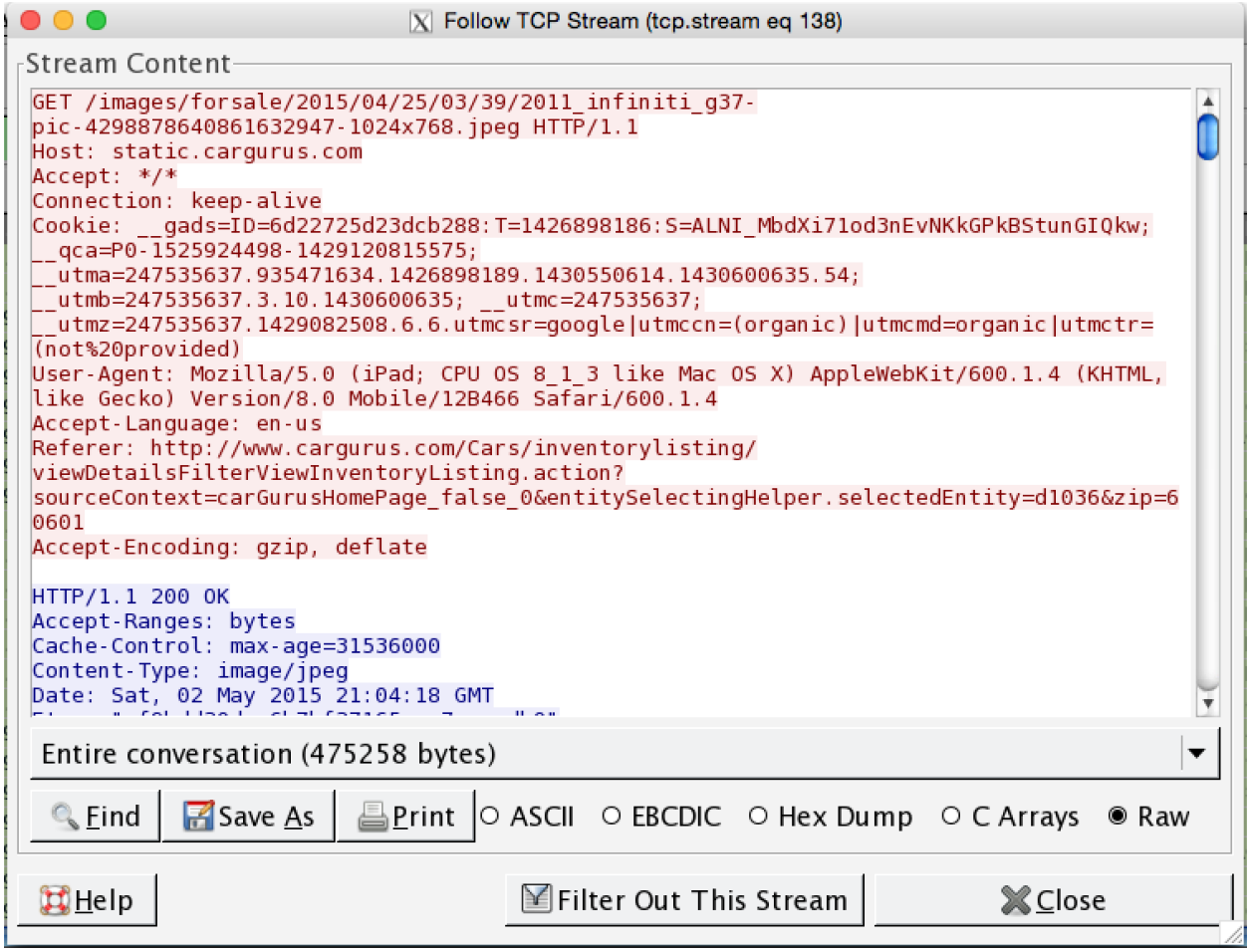
```
GET /ads/user-lists/991772620/?fmt=3&num=1&cv=7&frm=0&url=http%3A//www.cargurus.com/
&data=VER%3D1.0%3BAUTO_cat%3DCAR%3BAUTO_interest%3DBUY%3BURL%3D%2FCars%
2Findex.html&random=1620124324 HTTP/1.1
Host: www.google.com
Accept: */*
Connection: keep-alive
Cookie: AID=AJHaeXIDU5vUuCGFxiO1qcK3YuTlYzec1NKxdKMzw-optP1r6nN_Kg;
APISID=ITzQcgD0pcTBfEkf/AsPmfRJyPSBENS67z; HSID=Azmgpt0dkzrGW4PWt;
NID=67=Qu45m12YQSixamaJYkp1W4dnZoZRY8tYICSJvL7UvDMzf2r7Iyi5E4CMjw7FsL_O070_5gZgYPbRnBG2
JPSE4_65NIV95fwVnC74EnQgjeoSei1AeN-
ITAa30E6DvgJA0zbqpL0bxHAsVKMimZdbJEhzT2Kuz8FICVO1Lr1XUa2bAwJZICD3sN1QWxKrMoKDlzc8OznlcB
MrBrN3slm4FudDHIR6KBaDKRDiaai26MCCA3IChn9WFkZV86bEYFDJ1SH6J_rM-D8; OGP=-4061152:;
OGPC=4061152-3:;
PREF=ID=6b2cca208f8b417b:U=5dd50ea38f595f4c:FF=0:LD=en:TM=1419402172:LM=1421830810:S=zV
nomsOM4L8R5lmr; SID=DQAAAPQAAAATXcYEkAJ0xHpQC44IdWcagCHedRcm1IBPhYHXfNoiyv-
cQIDkryvxTzvDoDRyusDQVwdGgBTey-
v6Wvn1ridfwCMmL6BlmWxw7ASiPy_SuHVT5rnH8NDYhWgAx_wG_0uvNJ1OI0lmy-
z1F8kBYcBUKnkj754TJGoh_0HGV3W0lJi3UZB5k1rT_tdXWs5FQdO4n-qGTdKWv_v8XgWOwwSTBfvsh-51cSNy-
Nc6wjcarOqz5OZen-gwyLHbGEiXMGs65DR8LK1JB8C4bnBcrYoF3V8GIsUJoxFvZH7tDfaQJ9MC2cuy0C17asK-
fXjmWJpGRkryAQywLLhZ1BCU0K1f
User-Agent: Mozilla/5.0 (iPad; CPU OS 8_1_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4
Accept-Language: en-us
Referer: http://www.cargurus.com/
Accept-Encoding: gzip, deflate
```

Entire conversation (4010 bytes)

Find | Save As | Print | ○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ◉ Raw

Help | Filter Out This Stream | Close

Screenshot 2



Public AP at Starbucks

Logging in to the network at Starbucks was trivial. No password was required except a simple login page when a user is connected to the network. By analyzing the .wcap file obtained, we were able to discover websites that were currently being visited at that time. With the same setup, one team member was responsible for browsing the internet with one device while another team member was responsible for using the Wireless Diagnostics sniffer tool. Needless to say, we obtained more data in our .wcap file than just data from the team member that was browsing the internet. Two screenshots of different TCP streams are shown below. With this information, we can see the device that user is using, as well as the website that user was visiting.

X11 Applications Edit Window Help 38% Mon 10:58 AM

2015.05.02_16-13-14-CDT.wcap [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture A

Filter: tcp.stream eq 2

No.	Time	Source
11893	12.387820	172.31.98.6
11894	12.387892	172.31.98.6
12699	13.315593	172.31.98.6
12727	13.396124	199.59.150.
12729	13.396313	199.59.150.
12733	13.396654	172.31.98.6
12735	13.396824	172.31.98.6
13641	14.307260	172.31.98.6
13755	14.385927	199.59.150.
13757	14.386151	199.59.150.
13761	14.386484	172.31.98.6
13762	14.386557	172.31.98.6
16032	15.314366	172.31.98.6
16305	15.397196	199.59.150.
16307	15.397372	199.59.150.
16311	15.397872	172.31.98.6
16312	15.397946	172.31.98.6
18071	16.305690	172.31.98.6
18280	16.385881	199.59.150.
18282	16.386030	199.59.150.
18286	16.386650	172.31.98.6
18287	16.386730	172.31.98.6
20639	17.317000	172.31.98.6
20695	17.397905	199.59.150.

Follow TCP Stream (tcp.stream eq 2)

Stream Content

```
GET /i/adsct?
p_id=Twitter&p_user_id=0&txn_id=l4o6d&tw_sale_amount=0&tw_order_quantity=0 HTTP/1.1
Host: t.co
Accept: */*
Connection: keep-alive
Cookie: muc=f00323bc-eb5d-4915-ac70-da19c4339b2e
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_2) AppleWebKit/600.3.18
(KHTML, like Gecko) Version/8.0.3 Safari/600.3.18
Accept-Language: en-us
Referer: http://mashable.com/2015/05/02/dave-goldberg-dies/?utm_cid=mash-com-fb-main-
link
Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK
cache-control: no-cache, no-store, must-revalidate, pre-check=0, post-check=0
content-encoding: gzip
content-length: 65
content-type: image/gif;charset=utf-8
date: Sat, 02 May 2015 21:13:15 GMT
expires: Tue, 31 Mar 1981 05:00:00 GMT
last-modified: Sat, 02 May 2015 21:13:15 GMT
pragma: no-cache
server: tsa_a
status: 200 OK
x-connection-hash: 85c01db17b829aeaf66ce1057afbe771
```

Entire conversation (50164 bytes)

Find | Save As | Print | ○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ◉ Raw

Help | Filter Out This Stream | Close

```
0000  00 00 2c 00 4b 08 1c 00  fb 62 96 07 00 00 00 00   ..,.K... .b......
0010  10 00 99 16 40 01 a3 01  40 01 02 00 99 16 9d 22   ....@... @......"
0020  1f 09 0d 00 18 0c 00 00  04 00 00 00 88 01 30 00   ........ ......0.
0030  9c 1c 12 17 70 68 78 31  c1 d1 82 9a 9c 1c 12 c9   ....phx1 ........
0040  77 06 80 c1 00 00 aa aa  03 00 00 00 08 00 45 00   w....... ......E.
```

File: "/Users/fwirjo/Desktop... | Packets: 66308 · Displayed: 268 (0.4%) · Load time: 0:01.052 | Profile: Default

w Go Capture A

eam eq 337

Follow TCP Stream (tcp.stream eq 337)

Stream Content

```
GET /js/84098505.js HTTP/1.1
Host: cdn.optimizely.com
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (iPad; CPU OS 8_1_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4
Accept-Language: en-us
Referer: http://www.allseattletours.com/?
pref=02&aid=ltg1149&gclid=CjwKEAjw65GqBRCj3fLFwK2SpWoSJABa3E3c-
noJc4_Ekiab0sppEryKi4plokv4Yywu5X7_dmauxxoCiiDw_wcB
Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK
Content-Encoding: gzip
Accept-Ranges: bytes
Cache-Control: max-age=120
Content-Type: text/javascript
Date: Sat, 02 May 2015 21:13:54 GMT
Etag: "35864dcd13d2897c5b2bd3d2fd90c03c"
Last-Modified: Fri, 09 Jan 2015 03:28:35 GMT
Server: ECS (ord/4CE0)
Timing-Allow-Origin: *
Timing-Allow-Origin: *
Vary: Accept-Encoding
x-amz-id-2: kRggASB0Rq0y28WFAfT+HNVyjwfNTrm9VBgi4hyPKXPdr8XXF+nnecA/kfK2S
```

Entire conversation (50298 bytes)

Public AP at Espresso Royale

Since Espresso Royale is one of the more popular cafes on campus, we decided to test the public ap security there out of curiosity. There were several wireless networks at Espresso Royale, none of which require password to login. By analyzing the files we captured, we were able to obtain internet traffic data of other users that were on the same network. In this setup, one team member was responsible for browsing the internet with one device while another team member was responsible for using the Wireless Diagnostics sniffer tool. From the wcap files, we were able to track the device used(ipad) as well as the website visited, as you can see at the screenshot 1. As shown at screenshot 2, we're also able to determine the type of information being transferred. In the example of screenshot 2, the file being transferred is a text/javascript.

screenshot 1

Stream Content

```
GET / HTTP/1.1
Host: www.cargurus.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Cookie: __uvt=; uvts=2q1HAabgGiQ5BLCR; baseZip_asOf=61801_1429369356280;
cargurusGeoLocation=41.885%3A-87.622%3AChicago%2C+IL;
CarGurusUserT=63.252.64.201.1426898185451; mySavedListings=%7B%22id%22%3A%22ea2cb7c1-
dbbd-4ff6-999e-a1679dcd2635%22%7D;
pastListingSearches="{@s@:@USED@,@d@:50,@t@:1430539200000,@e@:@d1036@,@y1@:0,@y2@:0,@z@:@6
0601@,@l@:@en@}/{@s@:@USED@,@d@:100,@t@:1430539200000,@e@:@d376@,@z@:@60601@,@l@:@en@}/
{@s@:@USED@,@d@:50,@t@:1430539200000,@e@:@d589@,@z@:@60601@,@l@:@en@}/
{@s@:@USED@,@d@:50,@t@:1430539200000,@e@:@lb40@,@z@:@60601@,@l@:@en@}/
{@s@:@USED@,@d@:50,@t@:1430539200000,@e@:@lb42@,@z@:@60601@,@l@:@en@}/
{@s@:@USED@,@d@:50,@t@:1430539200000,@e@:@d733@,@z@:@60601@,@l@:@en@}/";
preferredContactInfo=Y2l0eT1DaGljYWdvKnBvc3RhbENvZGU9NjA2MDEqc3RhdGU9SUwqY291bnRyeT1VUyo_;
__gads=ID=6d22725d23dcb288:T=1426898186:S=ALNI_MbdXi71od3nEvNKkGPkBStunGIQkw;
__qca=P0-1525924498-1429120815575;
__utma=247535637.935471634.1426898189.1430550614.1430600635.54;
__utmb=247535637.4.10.1430600635; __utmz=247535637.1429082508.6.6.utmcsr=google|utmccn=
(organic)|utmcmd=organic|utmctr=(not%20provided)
User-Agent: Mozilla/5.0 (iPad; CPU OS 8_1_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4
Accept-Language: en-us
Accept-Encoding: gzip, deflate
```

screenshot 2

```
GET /button/checkOAuth.esi HTTP/1.1
Host: wd-edge.sharethis.com
Connection: keep-alive
Accept: */*
User-Agent: Mozilla/5.0 (iPad; CPU OS 8_1_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Version/8.0 Mobile/12B466 Safari/600.1.4
Accept-Language: en-us
Referer: http://www.etonline.com/
news/163808_what_feud_jennette_mccurdy_and_ariana_grande_are_friends_we_text_probably_once
_a_week/
Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK
Server: nginx/1.6.2
Content-Length: 22
Content-Type: text/javascript
Vary: Accept-Encoding
Expires: Sat, 02 May 2015 21:28:48 GMT
Cache-Control: max-age=0, no-cache, no-store
Pragma: no-cache
Date: Sat, 02 May 2015 21:28:48 GMT
Connection: keep-alive
```

IllinoisNet

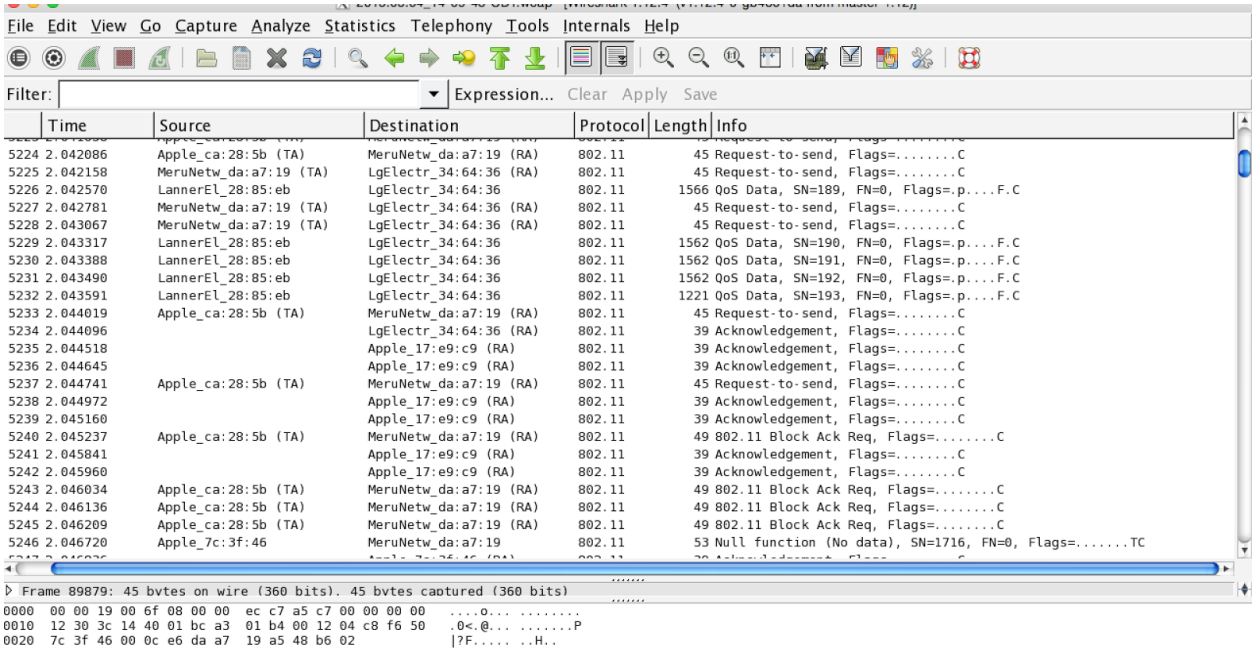
```
isr5860:~ fwirjo$ airport -s
                      UIUCnet 00:0c:e6:da:3a:e4 -46  6       Y  US NONE
                IllinoisNet 00:0c:e6:da:8c:f2 -46  6       Y  US WPA2(802.1x/AES/AES)
                IllinoisNet 48:d7:05:f0:86:90 -78  1       Y  US WPA2(PSK/AES/AES)
                      UIUCnet 00:0c:e6:da:3b:c7 -52  36,+1   Y  US NONE
                IllinoisNet 00:0c:e6:da:a7:19 -52  36,+1   Y  US WPA2(802.1x/AES/AES)

1 IBSS network found:
                          SSID BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
                Dell_device 7a:56:25:4d:d5:91 -79  10      N  -- NONE
```

```
isr5860:~ fwirjo$ airport -I
     agrCtlRSSI: -74
     agrExtRSSI: 0
    agrCtlNoise: -92
    agrExtNoise: 0
          state: running
        op mode: station
     lastTxRate: 216
        maxRate: 400
lastAssocStatus: 0
    802.11 auth: open
      link auth: wpa2
          BSSID: 0:c:e6:da:a7:19
           SSID: IllinoisNet
            MCS: 5
        channel: 36,1
```

IllinoisNet requires a password to login. We noticed that the link authentication is wpa2. WPA2 has support for encryption and authentication. As mentioned before, WPA2 uses

AES encryption algorithm and 802.1x authentication. These security measures ensure that users are unable to easily sniff wireless traffic and that data sent across the network is encrypted. When we tried to analyze the network through Wireshark, we were unable to find any TCP streams and the packets were all 802.11 packets. A screenshot of this is shown below.



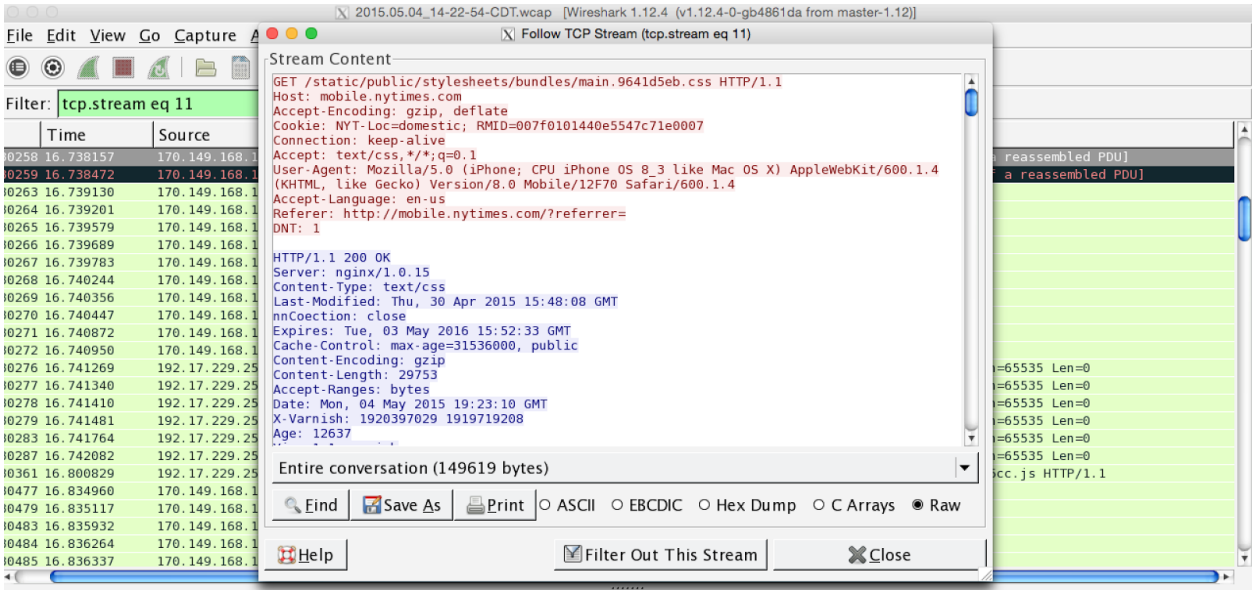
UIUCNet

```
isr5860:~ fwirjo$ airport -s
                            SSID BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
                      Front Desk 5c:96:9d:90:ca:6d -87  11      Y  -- WPA2(PSK/AES/AES)
                         UIUCnet 00:0c:e6:da:3a:e4 -47  6       Y  US NONE
                     IllinoisNet 00:0c:e6:da:8c:f2 -47  6       Y  US WPA2(802.1x/AES/AES)
                          XMWifi 8c:be:be:22:9b:ea -85  3       Y  -- WPA(PSK/AES,TKIP/TKIP) WPA2(PSK/AES,TKIP/TKIP)
                     IllinoisNet 48:d7:05:f0:86:90 -78  1       Y  US WPA2(PSK/AES/AES)
                         UIUCnet 00:0c:e6:da:3b:c7 -51  36,+1   Y  US NONE
                     IllinoisNet 00:0c:e6:da:a7:19 -51  36,+1   Y  US WPA2(802.1x/AES/AES)

1 IBSS network found:
                            SSID BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
                     Dell_device 7a:56:25:4d:d5:91 -72  10      N  -- NONE
```

```
isr5860:~ fwirjo$ airport -I
     agrCtlRSSI: -68
     agrExtRSSI: 0
    agrCtlNoise: -92
    agrExtNoise: 0
          state: running
        op mode: station
     lastTxRate: 135
        maxRate: 400
lastAssocStatus: 0
    802.11 auth: open
      link auth: none
          BSSID: 0:c:e6:da:3b:c7
           SSID: UIUCnet
            MCS: 7
        channel: 36,1
```

UIUCNet also requires a password to login, just like IllinoisNet. However, the link authentication is none, compared to IllinoisNet which has wpa2. This means that the network is less secure than that of IllinoisNet. Thus, we were able to obtain a TCP stream and view websites that other user visited. In this setup, I connected to UIUCnet on my laptop as well as my iPhone. Then I used my iPhone to browse the web while using my laptop to sniff traffic. Shown below is a screenshot of data obtained. You can see that the device I was using (iPhone), as well as the website i was visiting (mobile.nytimes.com).



Conclusion

After intensive analysis on public networks around campus, we've decided to compile a list of which wireless AP's are secure and which one's are insecure. The main criteria we used to decide whether a wireless AP was secure or not was based on data encryption. If we were able to obtain plain HTTP information about the websites other users were visiting on the network, then the network is insecure.

Secure Networks:

IllinoisNet

Sushi Ichiban

Insecure Networks:
McDonalds
Starbucks
Espresso Royale
Mia Zas
UIUCnet