

Education in Network Security



show version

- Username Ryan

- Research and Development Assistant at DePaul University
- B.S Information Assurance & Security Engineering from DePaul University
- Participated in over 25 security challenges over my collegiate career



show run | inc netriders



Round 1 Results

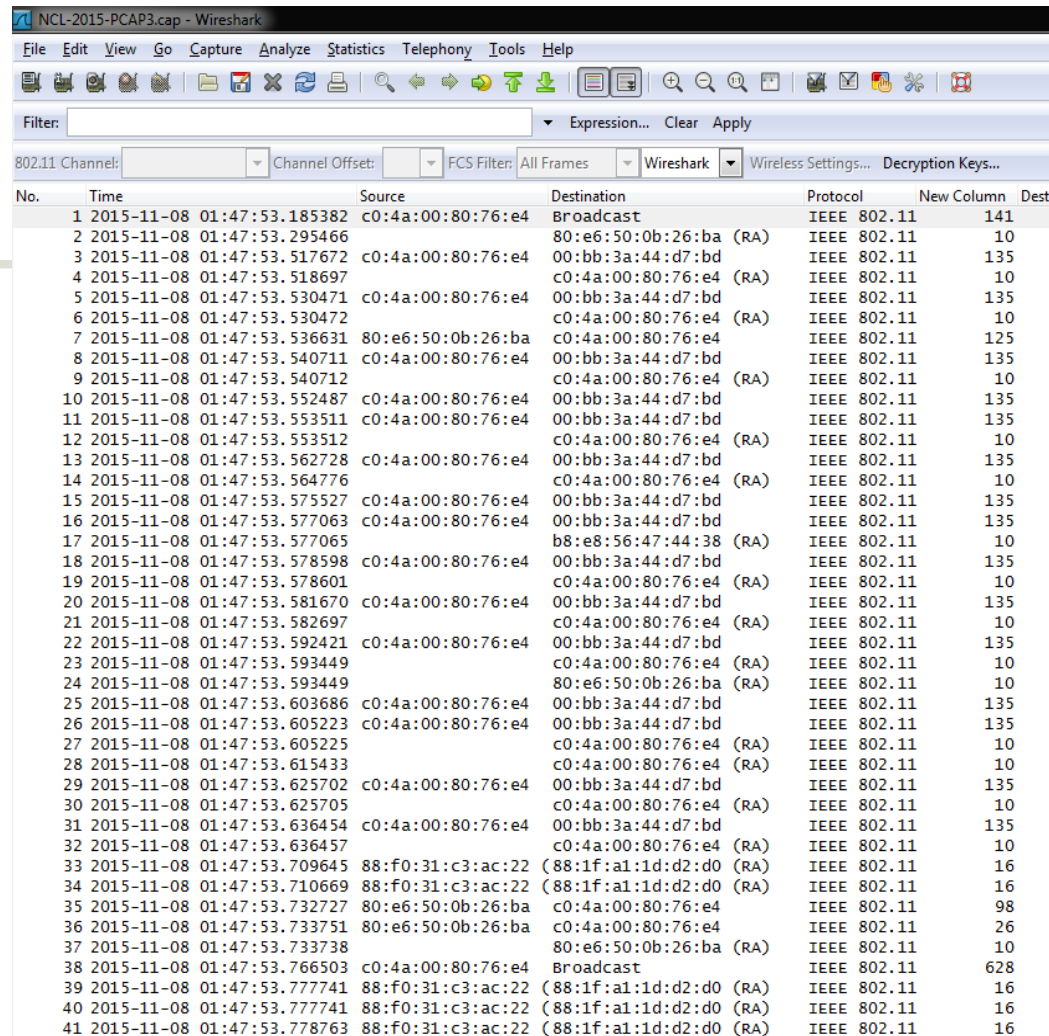
Posted November 1, 2011

Cisco Networking Academy NetRiders USA and Canada 2011 Post-Secondary Competition

- What is the most commonly used exterior routing protocol?
 - BGP
 - OSPF
 - EIGRP
 - RIP

show run | b NationalCyberLeague

- Jeopardy style CTF
- Categories include:
 - Log analysis
 - Reverse Engineering
 - Exploitation
 - Forensics
 - OSINT
 - **Network Traffic Analysis**



The image shows a Wireshark interface displaying a network capture file named 'NCL-2015-PCAP3.cap'. The main pane shows a list of 41 network packets. The columns include No., Time, Source, Destination, Protocol, and New Column. The packets are filtered by '802.11 Channel:'. The source and destination addresses are mostly 'c0:4a:00:80:76:e4' and '80:e6:50:0b:26:ba'. The protocol is 'IEEE 802.11'. The 'New Column' column shows packet lengths ranging from 10 to 628 bytes.

No.	Time	Source	Destination	Protocol	New Column
1	2015-11-08 01:47:53.185382	c0:4a:00:80:76:e4	Broadcast	IEEE 802.11	141
2	2015-11-08 01:47:53.295466	c0:4a:00:80:76:e4	80:e6:50:0b:26:ba (RA)	IEEE 802.11	10
3	2015-11-08 01:47:53.517672	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
4	2015-11-08 01:47:53.518697	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
5	2015-11-08 01:47:53.530471	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
6	2015-11-08 01:47:53.530472	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
7	2015-11-08 01:47:53.536631	80:e6:50:0b:26:ba	c0:4a:00:80:76:e4	IEEE 802.11	125
8	2015-11-08 01:47:53.540711	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
9	2015-11-08 01:47:53.540712	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
10	2015-11-08 01:47:53.552487	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
11	2015-11-08 01:47:53.553511	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
12	2015-11-08 01:47:53.553512	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
13	2015-11-08 01:47:53.562728	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
14	2015-11-08 01:47:53.564776	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
15	2015-11-08 01:47:53.575527	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
16	2015-11-08 01:47:53.577063	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
17	2015-11-08 01:47:53.577065	c0:4a:00:80:76:e4	b8:e8:56:47:44:38 (RA)	IEEE 802.11	10
18	2015-11-08 01:47:53.578598	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
19	2015-11-08 01:47:53.578601	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
20	2015-11-08 01:47:53.581670	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
21	2015-11-08 01:47:53.582697	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
22	2015-11-08 01:47:53.592421	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
23	2015-11-08 01:47:53.593449	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
24	2015-11-08 01:47:53.593449	c0:4a:00:80:76:e4	80:e6:50:0b:26:ba (RA)	IEEE 802.11	10
25	2015-11-08 01:47:53.603686	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
26	2015-11-08 01:47:53.605223	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
27	2015-11-08 01:47:53.605225	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
28	2015-11-08 01:47:53.615433	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
29	2015-11-08 01:47:53.625702	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
30	2015-11-08 01:47:53.625705	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
31	2015-11-08 01:47:53.636454	c0:4a:00:80:76:e4	00:bb:3a:44:d7:bd	IEEE 802.11	135
32	2015-11-08 01:47:53.636457	c0:4a:00:80:76:e4	c0:4a:00:80:76:e4 (RA)	IEEE 802.11	10
33	2015-11-08 01:47:53.709645	88:f0:31:c3:ac:22	(88:1f:a1:1d:d2:d0) (RA)	IEEE 802.11	16
34	2015-11-08 01:47:53.710669	88:f0:31:c3:ac:22	(88:1f:a1:1d:d2:d0) (RA)	IEEE 802.11	16
35	2015-11-08 01:47:53.732727	80:e6:50:0b:26:ba	c0:4a:00:80:76:e4	IEEE 802.11	98
36	2015-11-08 01:47:53.733751	80:e6:50:0b:26:ba	c0:4a:00:80:76:e4	IEEE 802.11	26
37	2015-11-08 01:47:53.733738	80:e6:50:0b:26:ba	80:e6:50:0b:26:ba (RA)	IEEE 802.11	10
38	2015-11-08 01:47:53.766503	c0:4a:00:80:76:e4	Broadcast	IEEE 802.11	628
39	2015-11-08 01:47:53.777741	88:f0:31:c3:ac:22	(88:1f:a1:1d:d2:d0) (RA)	IEEE 802.11	16
40	2015-11-08 01:47:53.777741	88:f0:31:c3:ac:22	(88:1f:a1:1d:d2:d0) (RA)	IEEE 802.11	16
41	2015-11-08 01:47:53.778763	88:f0:31:c3:ac:22	(88:1f:a1:1d:d2:d0) (RA)	IEEE 802.11	16

show int f0/2

Custom Protocol

The hackers have **created their own custom protocol** for private communication. Luckily, police officers have managed to obtain the documentation describing the protocol. Use it to fill out this report.

Overview

The communication between the client and server will contain three types of messages: Initialization, Encrypt Request, and Encrypt Response. A connection is started with the client sending an Initialization message, which contains the number of Encrypt Requests that the client wishes to make. Then, the server will send the length of its response. Then, the client sends their Encrypt Requests to the server. After all of the Encrypt Requests have been received, the server will finish sending a single Encrypt Response which contains hashes of all of the data that was sent by the client.

Initialization (Client -> Server)

1. N - A 4-byte integer in network byte order that represents the number of Encrypt Requests that will be sent.

Encrypt Request (Client -> Server)

1. Check - A fixed 2-byte integer in network byte order that verifies the integrity of the message.
2. Len - A 4-byte integer in network byte order that represents the length of the data in bytes.
3. Data - The data that will be encrypted.

Encrypt Response (Server -> Client)

1. Count - The length of the data, in bytes, that follows.
2. Hashes - The encrypted hashes requested by the client. Each hash is in the form of a fixed-length chunk. These hashes are in the same order that the requests were made.



Network Traffic Analysis Report

Question

What is the IP address of the server?

What is the IP address of the client?

What port is the server listening on?

What is the magic 2-byte ID in decimal?

How many encrypt requests were made by the client?

What is the length of the first encrypt request?

What is the length of the second encrypt request?

How large is an individual encrypt hash in bytes?

What was the encrypt response (in the form 0xFFFF) for the first request?



Filter: ip.addr==10.1.0.20 && ip.addr==10.1.0.217

Channel: Channel Offset: FCS Filter: All Frames Wireshark Wireless Settings... Decryption Keys...

Time	Source	Destination	Protocol	New Column	Destination Port	Sequence number	Acknowledgement number	TCP length	Destination Port	Info
5497 2015-10-27 16:44:55.841725	10.1.0.217	10.1.0.20	ICMP	100						Echo (ping) re
5498 2015-10-27 16:44:55.841804	10.1.0.20	10.1.0.217	ICMP	100						Echo (ping) re
5515 2015-10-27 16:44:56.268572	10.1.0.217	10.1.0.20	TCP	76		0		0	60123	42455 > 60123
5516 2015-10-27 16:44:56.268652	10.1.0.20	10.1.0.217	TCP	76		0		1	0	42455 60123 > 42455
5517 2015-10-27 16:44:56.268659	10.1.0.217	10.1.0.20	TCP	68		1		1	0	60123 42455 > 60123
5518 2015-10-27 16:44:56.268676	10.1.0.217	10.1.0.20	TCP	72		1		1	4	60123 42455 > 60123
5520 2015-10-27 16:44:56.268717	10.1.0.20	10.1.0.217	TCP	68		1		5	0	42455 60123 > 42455
5522 2015-10-27 16:44:56.268749	10.1.0.217	10.1.0.20	TCP	70		5		1	2	60123 42455 > 60123
5528 2015-10-27 16:44:56.268926	10.1.0.20	10.1.0.217	TCP	72		1		5	4	42455 60123 > 42455
5529 2015-10-27 16:44:56.268938	10.1.0.217	10.1.0.20	TCP	68		7		5	0	60123 42455 > 60123
5544 2015-10-27 16:44:56.308697	10.1.0.20	10.1.0.217	TCP	68		5		7	0	42455 60123 > 42455
5545 2015-10-27 16:44:56.308703	10.1.0.217	10.1.0.20	TCP	484		7		5	416	60123 42455 > 60123
5546 2015-10-27 16:44:56.308838	10.1.0.20	10.1.0.217	TCP	68		5		423	0	42455 60123 > 42455
5547 2015-10-27 16:44:56.308843	10.1.0.20	10.1.0.217	TCP	100		5		423	32	42455 60123 > 42455
5548 2015-10-27 16:44:56.308853	10.1.0.217	10.1.0.20	TCP	68		423		37	0	60123 42455 > 60123

Frame 5518: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)

Linux cooked capture

Internet Protocol, Src: 10.1.0.217 (10.1.0.217), Dst: 10.1.0.20 (10.1.0.20)

Transmission Control Protocol, Src Port: 42455 (42455), Dst Port: 60123 (60123), Seq: 1, Ack: 1, Len: 4

Source port: 42455 (42455)

Destination port: 60123 (60123)

[Stream index: 116]

Sequence number: 1 (relative sequence number)

[Next sequence number: 5 (relative sequence number)]

Acknowledgement number: 1 (relative ack number)

Header length: 32 bytes

Flags: 0x18 (PSH, ACK)

Window size: 29312 (scaled)

Checksum: 0x1519 [validation disabled]

Options: (12 bytes)

[SEQ/ACK analysis]

[Timestamps]

Data (4 bytes)

Data: 00000005

[Length: 4]

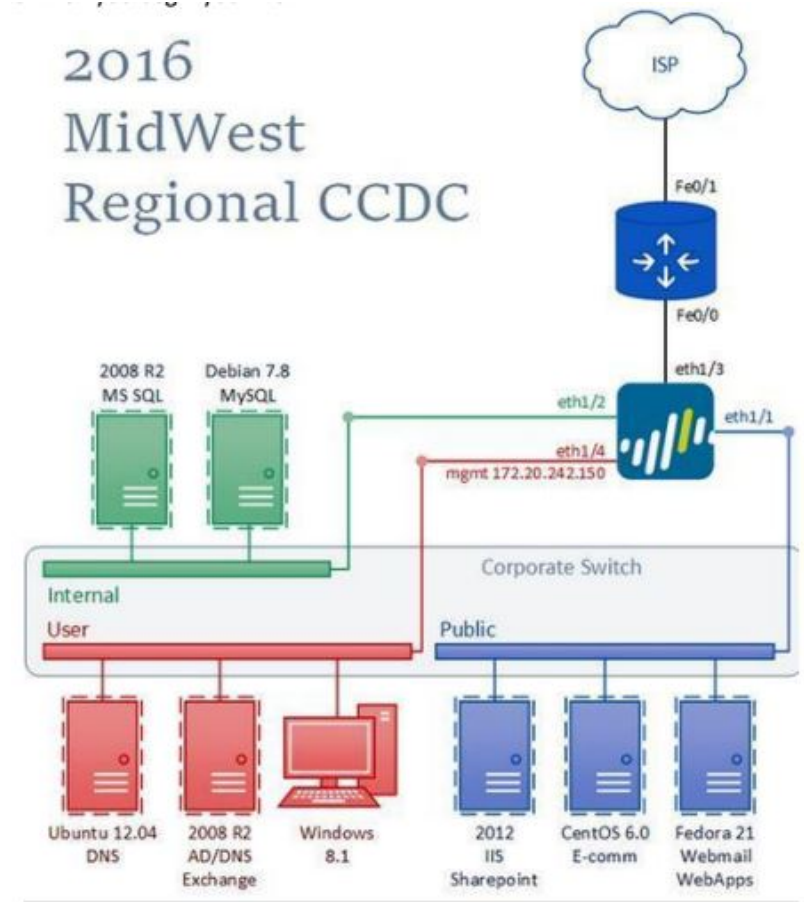
```

0 00 04 00 01 00 06 00 0c 29 e2 ef 41 00 d9 08 00 .....).A....
0 45 00 00 38 03 68 40 00 40 06 22 6a 0a 01 00 d9 E..8.h@.@."j....
0 0a 01 00 14 a5 d7 ea db e0 5f c0 a0 96 21 2c 41 ....._.!.,A
0 80 18 00 e5 15 19 00 00 01 01 08 0a 14 bd c3 f9 .....
0 6e 4d 58 18 00 00 00 05 nMX....

```

show run | inc CCDC

- Collegiate Cyber Defense Competition
 - Red versus Blue
 - Strictly defensive
 - Teams of 8 students
 - Includes business injects



show int e0/3

```
interface FastEthernet0/7
  description *** ADS Port ***
  switchport access vlan 999
  switchport trunk native vlan 999
  switchport trunk allowed vlan none
  switchport mode access
  switchport nonegotiate
  switchport port-security
  switchport port-security aging time 10
  switchport port-security aging type inactivity
  switchport port-security mac-address sticky
  ip access-group ip-device-list in
  shutdown
  mls qos cos override
  storm-control broadcast level 0.00
  storm-control multicast level 0.00
  storm-control unicast level 0.00
  no cdp enable
  spanning-tree portfast
  spanning-tree bpduguard enable
  --More--
```


show int fa0/3

```
scheduler allocate 20000 1000
event manager applet config
  event cli pattern "show* (run*|star*|conf*)" sync yes
  action 1.0 cli command "show running-config | exclude ^snmp-server"
  action 2.0 puts "$_cli_result"
  action 3.0 set $_exit_status "0"
event manager applet flash
  event cli pattern "show* flash*" sync yes
  action 1.0 cli command "show flash: | exclude .tcl"
  action 2.0 puts "$_cli_result"
  action 3.0 set $_exit_status "0"
event manager applet users
  event cli pattern "show* users*" sync yes
  action 1.0 cli command "show users | exclude .vty"
  action 2.0 puts "$_cli_result"
  action 3.0 set $_exit_status "0"
event manager applet system
  event timer cron name system cron-entry "*/15 * * * *"
  action 1 cli command "enable"
  action 2 cli command "tclsh flash:sem.tcl"
!
end
```

show int g0/3

- Firewall Configuration

- How do you defend against an enemy that has root access to your firewall???

```
[dadmin@PA-VM]$ echo "Hello from redteam" | wall
[dadmin@PA-VM]$
Broadcast message from dadmin (Sat Apr 2 15:01:16 2016):

Hello from redteam

[dadmin@PA-VM]$
Broadcast message from dadmin (Sat Apr 2 15:01:36 2016):

nice

[dadmin@PA-VM]$

[dadmin@PA-VM]$

[dadmin@PA-VM]$
Broadcast message from dadmin (Sat Apr 2 15:02:24 2016):

how did you get a bash shell
```

delete flash:

- Twitter: @r_haley
- LinkedIn: [linkedin.com/in/ryanhaley](https://www.linkedin.com/in/ryanhaley)

