Configure Packet Captures on AireOS WLC

Contents

Introduction

This document describes how to run a packet dump on a AireOS Wireless LAN Controller (WLC).

Requirements

Cisco recommends that you have knowledge of these topics:

- Command Line Interface (CLI) access to the WLC.
- PC with Wireshark installed

Components Used

The information in this document is based on these software and hardware versions:

- WLC v8.3
- Wireshark v2 or later

Note: This feature is available since AireOS version 4.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Limitations

The packet logging captures only bidirectional Control Plane (CP) to Data Plane (DP) packets in WLC. Those packets which are not sent from WLC Data plane to/from control plane (that is, foreign to anchor tunneled traffic, DP-CP drops and so on) cannot be captured.

Examples of types of traffic to/from the WLC processed at the CP are:

- Telnet
- SSH
- HTTP
- HTTPS
- SNMP
- NTP
- RADIUS
- TACACS+
- Mobility Messages
- CAPWAP control
- NMSP
- TFTP/FTP/SFTP
- Syslog
- IAPP

The traffic to/from the client is processed in the Data Plane (DP) except for: 802.11 management, 802.1X/EAPOL, ARP, DHCP and Web Authentication.

Background Information

This method displays the packets sent and/or received at CPU level of the WLC in hex format, which then be translated to a .pcap file with Wireshark. It is helpful in cases where communication between a WLC and a Remote Authentication Dial-In User Service (RADIUS) server, an Access Point (AP) or other controllers needs to be verified in a quick way with a packet capture at the WLC level but a port-span is hard to perform.

Configure

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Enable packet logging in WLC

Step 1. Log in to the WLC CLI.

Due to the quantity and speed of logs that this feature displays it is recommended to login to the WLC by SSH and not by console (SSH is preferred since the output is faster than console).

Step 2. Apply an Access Control List (ACL) to limit which traffic is captured.

In the given example the capture shows the traffic to/from the WLC management interface (IP address 172.16.0.34) and the RADIUS server (172.16.56.153).

```
<#root>

>

debug packet logging acl ip 1 permit 172.16.0.34 172.16.56.153

>

 debug packet logging acl ip 2 permit 172.16.56.153 172.16.0.34
```

Tip: To capture all the traffic to/from the WLC it is recommended to apply an ACL that discards the SSH traffic to/from the host that initiated the SSH session. These are the commands that you can use to build the ACL:

```
> debug packet logging acl ip 1 deny <WLC-IP> <host-IP> tcp 22 any
> debug packet logging acl ip 2 deny <host-IP> <WLC-IP> tcp any 22
> debug packet logging acl ip 3 permit any any
```

Step 3. Configure the format readable by Wireshark.

```
<#root>

>

debug packet logging format text2pcap
```

Step 4. Enable packet logging feature.

This example shows how to capture 100 received/transmitted packets (it supports 1 - 65535 packets):

```
<#root>

>

debug packet logging enable all 100
```

Step 5. Log the output to a text file.

Note: By default, it only logs 25 received packets with the command **debug packet logging enable**.

Note: Instead of **all** you can use **rx** or **tx** to capture only received or transmitted traffic.

For further details about configuring packet logging feature consult this link:

Cisco Wireless Controller Configuration Guide, Release 8.3, Using the Debug Facility

Verify

Use this section in order to confirm that your configuration works properly.

Use the given command to verify the current configuration of packet logging.

```
<#root>
```

```
>

show debug packet


Status........................................... rx/tx            !!! This means the capture is active

Number of packets to display..................... 100

Bytes/packet to display.......................... 0

Packet display format............................ text2pcap


   Driver ACL:
      [1]: disabled
      [2]: disabled
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
   Ethernet ACL:
      [1]: disabled
      [2]: disabled
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
   IP ACL:
      [1]: permit s=172.16.0.34 d=172.16.56.153 any
      [2]: permit s=172.16.56.153 d=172.16.0.34 any
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
   EoIP-Ethernet ACL:
      [1]: disabled
      [2]: disabled
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
   EoIP-IP ACL:
      [1]: disabled
      [2]: disabled
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
   LWAPP-Dot11 ACL:
      [1]: disabled
      [2]: disabled
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
   LWAPP-IP ACL:
      [1]: disabled
      [2]: disabled
      [3]: disabled
      [4]: disabled
      [5]: disabled
      [6]: disabled
```

Reproduce the needed behavior to generate the traffic.

An output similar to this appears:

```
rx len=108, encap=unknown, port=2
0000 E0 89 9D 43 EF 40 C8 5B 76 1D AB 51 81 00 09 61 `..Co@H[v.+Q...a
0010 08 00 45 00 00 5A 69 81 00 00 80 01 78 A7 AC 10 ..E..Zi.....x',.
0020 00 38 AC 10 00 22 03 03 55 B3 00 00 00 00 45 00 .8,.."..U3....E.
0030 00 3E 0B 71 00 00 FE 11 58 C3 AC 10 00 22 AC 10 .>.q..~.XC,..",.
0040 00 38 15 B3 13 88 00 2A 8E DF A8 a1 00 0E 00 0E .8.3...*._(!....
0050 01 00 00 00 00 22 F1 FC 8B E0 18 24 07 00 C4 00 ....."q|.`.$..D.
0060 F4 00 50 1C BF B5 F9 DF EF 59 F7 15 t.P.?5y_oYw.
rx len=58, encap=ip, port=2
0000 E0 89 9D 43 EF 40 C8 5B 76 1D AB 51 81 00 09 61 `..Co@H[v.+Q...a
0010 08 00 45 00 00 28 69 82 40 00 80 06 38 D3 AC 10 ..E..(i.@...8S,.
0020 00 38 AC 10 00 22 F6 3A 00 16 AF 52 FE F5 1F 0C .8,.."v:../R~u..
0030 40 29 50 10 01 01 52 8A 00 00 @)P...R...
rx len=58, encap=ip, port=2
0000 E0 89 9D 43 EF 40 C8 5B 76 1D AB 51 81 00 09 61 `..Co@H[v.+Q...a
0010 08 00 45 00 00 28 69 83 40 00 80 06 38 D2 AC 10 ..E..(i.@...8R,.
0020 00 38 AC 10 00 22 F6 3A 00 16 AF 52 FE F5 1F 0C .8,.."v:../R~u..
0030 41 59 50 10 01 00 51 5B 00 00 AYP...Q[..
rx len=58, encap=ip, port=2
0000 E0 89 9D 43 EF 40 C8 5B 76 1D AB 51 81 00 09 61 `..Co@H[v.+Q...a
0010 08 00 45 00 00 28 69 84 40 00 80 06 38 D1 AC 10 ..E..(i.@...8Q,.
0020 00 38 AC 10 00 22 F6 3A 00 16 AF 52 FE F5 1F 0C .8,.."v:../R~u..
0030 43 19 50 10 01 05 4F 96 00 00 C.P...O...
```

Remove ACLs from packet logging

In order to disable the filters applied by the ACLs use these commands:

```
<#root>

>

debug packet logging acl ip 1 disable


>

debug packet logging acl ip 2 disable
```

Disable packet logging

In order to disable the packet logging without removing the ACLs simply use this command:

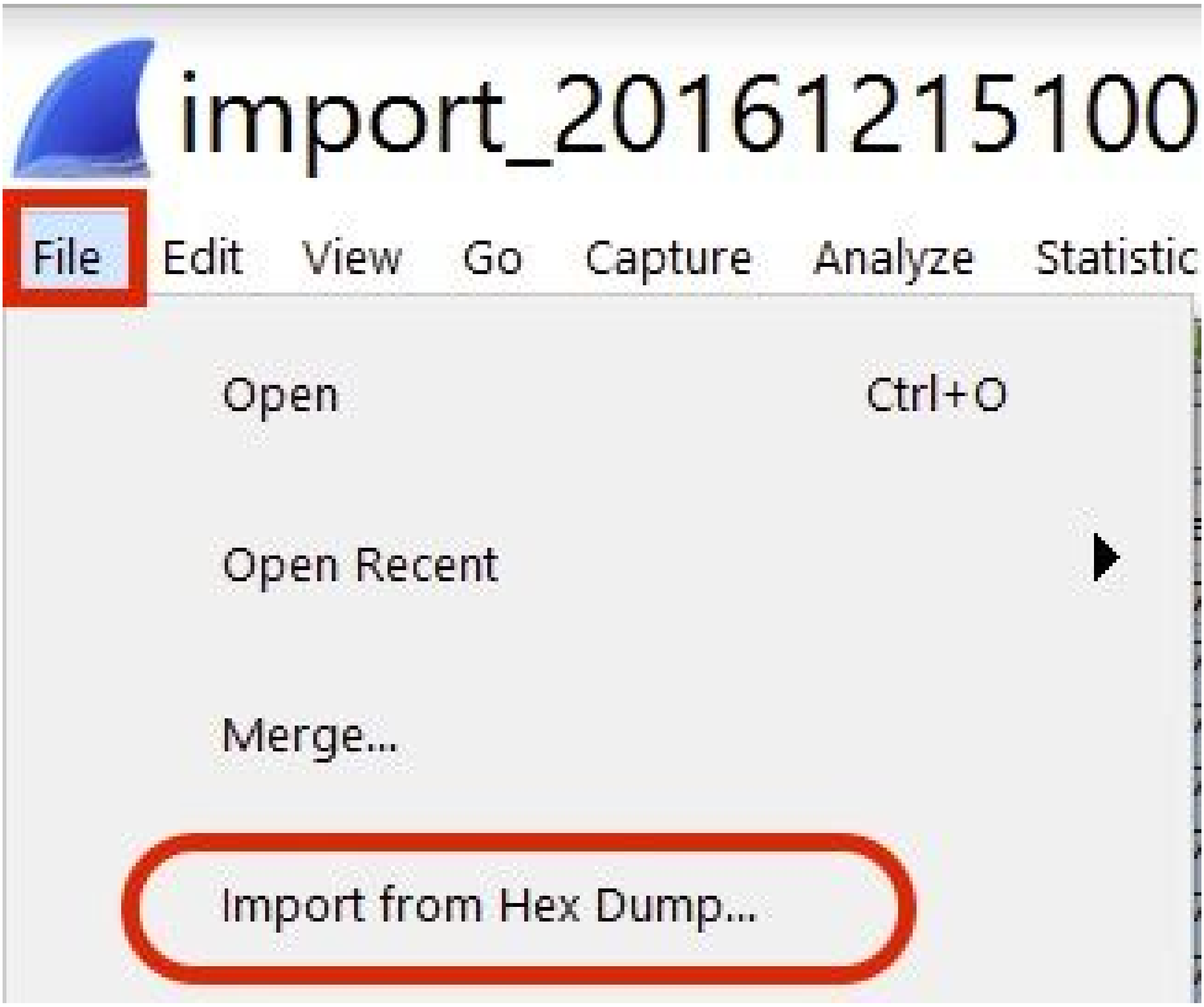
```
<#root>

>

debug packet logging disable
```

Convert packet logging output to a .pcap file

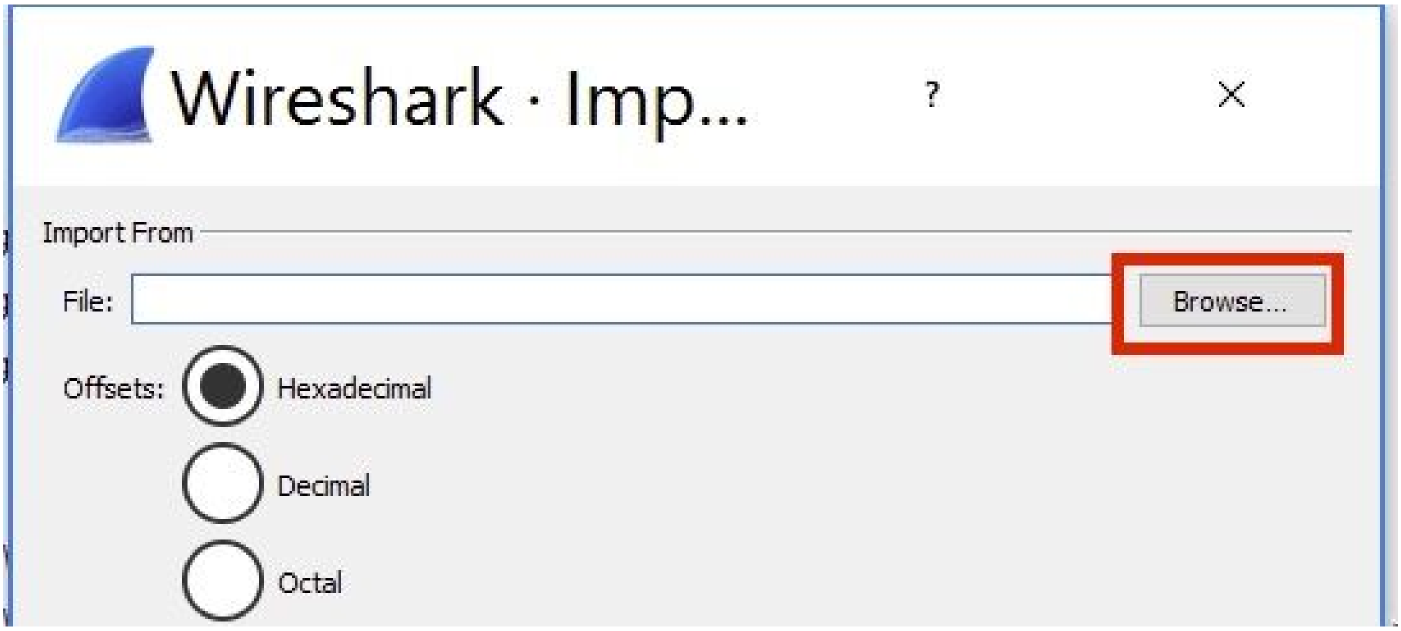
Step 1. Once the output finishes, collect it and save it to a text file.

Ensure that you gather a clean log, otherwise Wireshark can show corrupted packets.

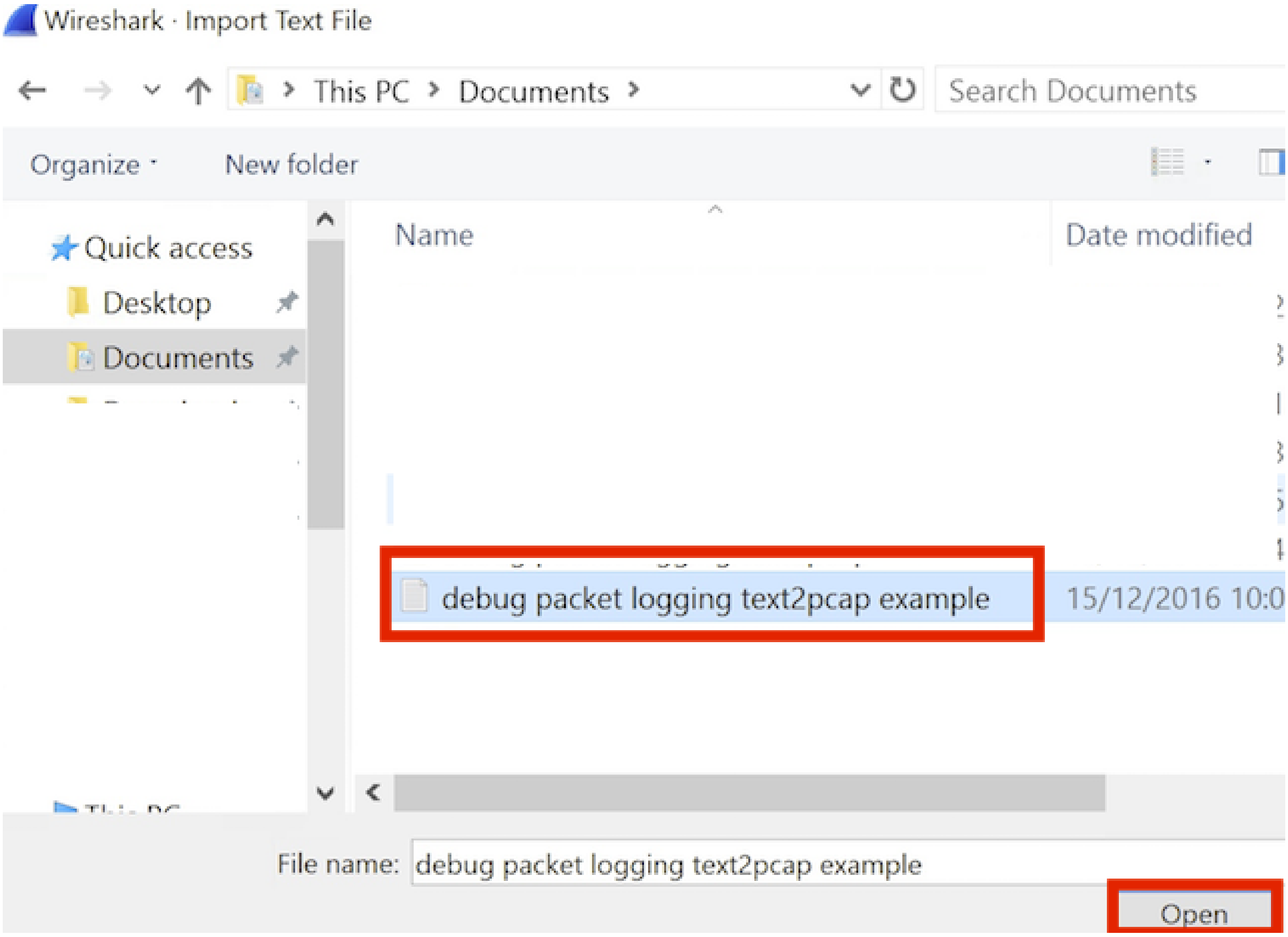
Step 2. Open Wireshark and navigate to **File>Import from Hex Dump...**



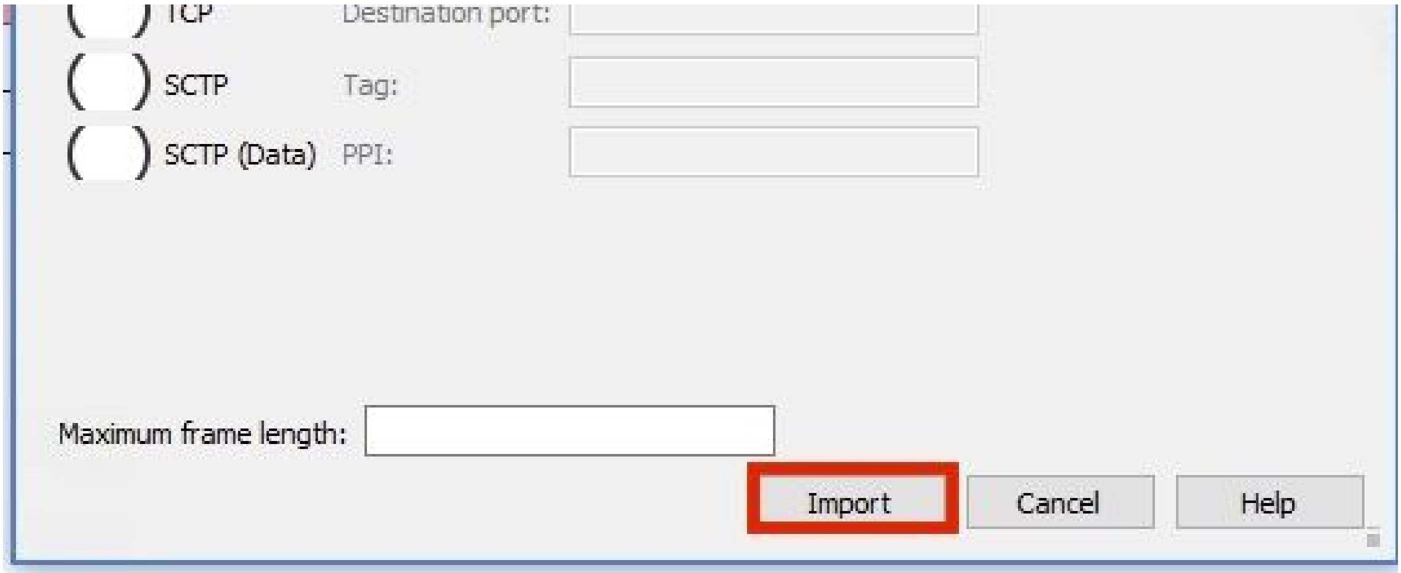
Step 3. Click **Browse.**



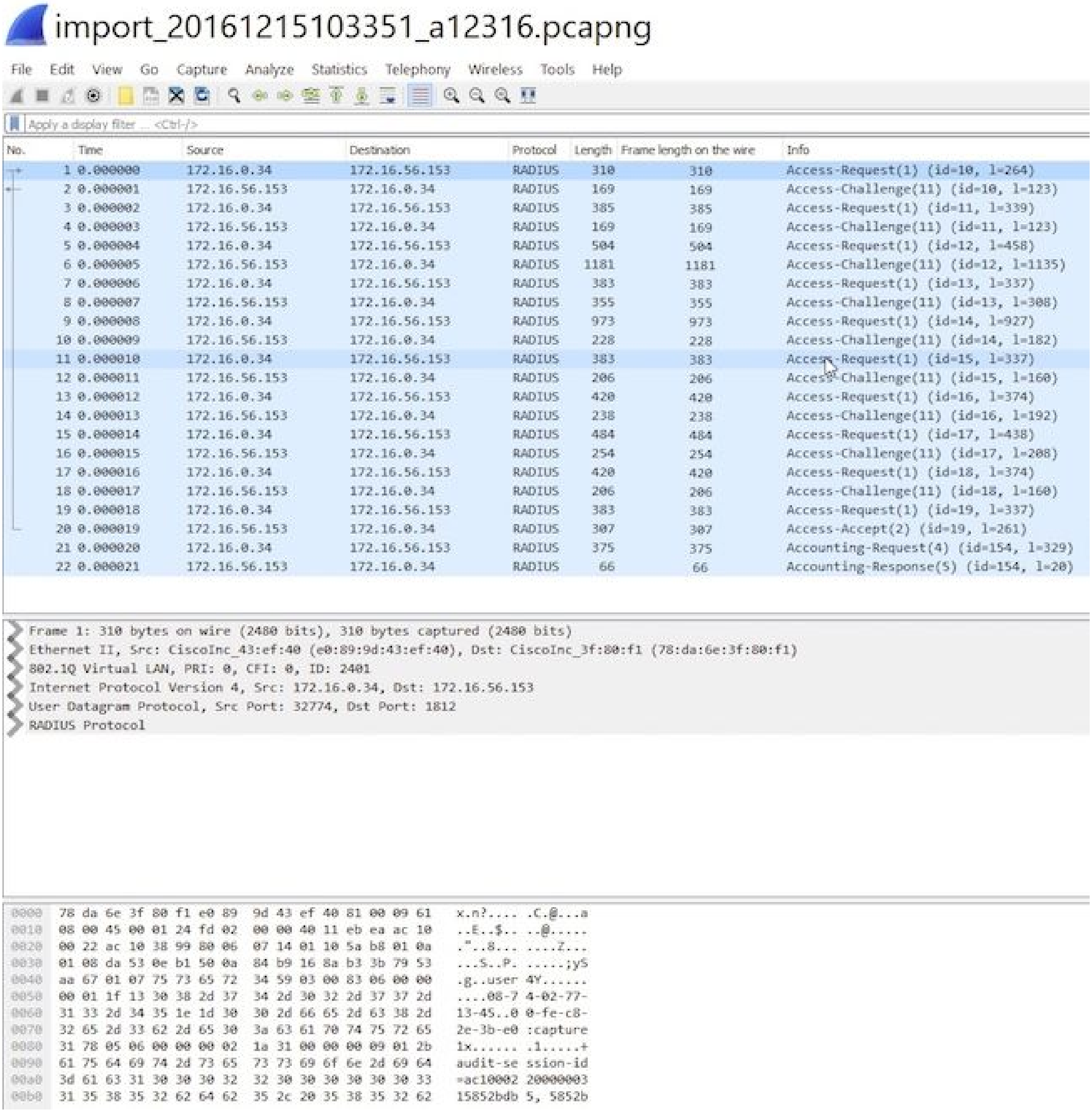
Step 4. Select the text file where you saved the packet logging output.



Step 5. Click **Import.**



Wireshark shows the file as .pcap.


import_20161215103351_a12316.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time       Source          Destination     Protocol  Length  Frame length on the wire  Info
   1 0.000000   172.16.0.34     172.16.56.153   RADIUS     310     310    Access-Request(1) (id=10, l=264)
   2 0.000001   172.16.56.153   172.16.0.34     RADIUS     169     169    Access-Challenge(11) (id=10, l=123)
   3 0.000002   172.16.0.34     172.16.56.153   RADIUS     385     385    Access-Request(1) (id=11, l=339)
   4 0.000003   172.16.56.153   172.16.0.34     RADIUS     169     169    Access-Challenge(11) (id=11, l=123)
   5 0.000004   172.16.0.34     172.16.56.153   RADIUS     504     504    Access-Request(1) (id=12, l=458)
   6 0.000005   172.16.56.153   172.16.0.34     RADIUS    1181    1181    Access-Challenge(11) (id=12, l=1135)
   7 0.000006   172.16.0.34     172.16.56.153   RADIUS     383     383    Access-Request(1) (id=13, l=337)
   8 0.000007   172.16.56.153   172.16.0.34     RADIUS     355     355    Access-Challenge(11) (id=13, l=308)
   9 0.000008   172.16.0.34     172.16.56.153   RADIUS     973     973    Access-Request(1) (id=14, l=927)
  10 0.000009   172.16.56.153   172.16.0.34     RADIUS     228     228    Access-Challenge(11) (id=14, l=182)
  11 0.000010   172.16.0.34     172.16.56.153   RADIUS     383     383    Access-Request(1) (id=15, l=337)
  12 0.000011   172.16.56.153   172.16.0.34     RADIUS     206     206    Access-Challenge(11) (id=15, l=160)
  13 0.000012   172.16.0.34     172.16.56.153   RADIUS     420     420    Access-Request(1) (id=16, l=374)
  14 0.000013   172.16.56.153   172.16.0.34     RADIUS     238     238    Access-Challenge(11) (id=16, l=192)
  15 0.000014   172.16.0.34     172.16.56.153   RADIUS     484     484    Access-Request(1) (id=17, l=438)
  16 0.000015   172.16.56.153   172.16.0.34     RADIUS     254     254    Access-Challenge(11) (id=17, l=208)
  17 0.000016   172.16.0.34     172.16.56.153   RADIUS     420     420    Access-Request(1) (id=18, l=374)
  18 0.000017   172.16.56.153   172.16.0.34     RADIUS     206     206    Access-Challenge(11) (id=18, l=160)
  19 0.000018   172.16.0.34     172.16.56.153   RADIUS     383     383    Access-Request(1) (id=19, l=337)
  20 0.000019   172.16.56.153   172.16.0.34     RADIUS     307     307    Access-Accept(2) (id=19, l=261)
  21 0.000020   172.16.0.34     172.16.56.153   RADIUS     375     375    Accounting-Request(4) (id=154, l=329)
  22 0.000021   172.16.56.153   172.16.0.34     RADIUS      66      66    Accounting-Response(5) (id=154, l=20)
```

```
> Frame 1: 310 bytes on wire (2480 bits), 310 bytes captured (2480 bits)
> Ethernet II, Src: CiscoInc_43:ef:40 (e0:89:9d:43:ef:40), Dst: CiscoInc_3f:80:f1 (78:da:6e:3f:80:f1)
> 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 2401
> Internet Protocol Version 4, Src: 172.16.0.34, Dst: 172.16.56.153
> User Datagram Protocol, Src Port: 32774, Dst Port: 1812
> RADIUS Protocol
```

```
0000  78 da 6e 3f 80 f1 e0 89  9d 43 ef 40 81 00 09 61   x.n?.... .C.@...a
0010  08 00 45 00 01 24 fd 02  00 00 40 11 eb ea ac 10   ..E..$.. ..@.....
0020  00 22 ac 10 38 99 80 06  07 14 01 10 5a b8 01 0a   ."..8... ....Z...
0030  01 08 da 53 0e b1 50 0a  84 b9 16 8a b3 3b 79 53   ...S..P. .....;yS
0040  aa 67 01 07 75 73 65 72  34 59 03 00 83 06 00 00   .g..user 4Y......
0050  00 01 1f 13 30 38 2d 37  34 2d 30 32 2d 37 37 2d   ....08-7 4-02-77-
0060  31 33 2d 34 35 1e 1d 30  30 2d 66 65 2d 63 38 2d   13-45..0 0-fe-c8-
0070  32 65 2d 33 62 2d 65 30  3a 63 61 70 74 75 72 65   2e-3b-e0 :capture
0080  31 78 05 06 00 00 00 02  1a 31 00 00 00 09 01 2b   1x...... .1.....+
0090  61 75 64 69 74 2d 73 65  73 73 69 6f 6e 2d 69 64   audit-se ssion-id
00a0  3d 61 63 31 30 30 30 32  32 30 30 30 30 30 30 33   =ac10002 20000003
00b0  31 35 38 35 32 62 64 62  35 2c 20 35 38 35 32 62   15852bdb 5, 5852b
```

Note: Be aware that the time stamps are not accurate nor the delta time between the frames.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

Related Information

- AP Packet Dump
- Fundamentals of 802.11 Wireless Sniffing
- Cisco Technical Support & Downloads