Convert Access Point Packet Dumps for Wireshark

Contents

Introduction	
Prerequisites	
Procedure	
Perform Packet	Dump
Output File Clea	mup
Cleanup Packet	Summary Information
Remove starting	spaces and offset colons
Correct packet of	ffset
Separate Packet	Bytes
Convert the Tex	t File to PCAP
Via Wireshark	<u>iui</u>
Via command li	ne
Troubleshooting	t
Text File is Cor	rect but Text2pcap Cannot Read Any Packets
Inconsistent Of	set

Introduction

This document describes how to convert a COS Access Point generated packet dump to PCAP format for Wireshark as a workaround to the size limitation.

Prerequisites

- Notepad++ Available only on Windows
- Text2pcap installed included on regular installations of Wireshark

Procedure

Perform Packet Dump

Capture an AP packet dump by running the command **debug traffic wired <multiple options> verbose** on the AP command line. You can choose between multiple filters and interfaces.

Log the session in the terminal.

Be careful to send the least amount of keystrokes when doing so, the more printable characters on the file that do not belong to the capture itself the more cleanup you need to do before conversion. The easiest way to do it is a console session for the packet dump, replicate the issue, stop the dump and immediately end the session.

If you are performing the dump via ssh use a filter to only capture the traffic of interest. Otherwise the capture contains the ssh session packets.

Refer to Troubleshoot COS APs for complete instructions on how to configure the capture.

When you are done, stop the capture with the command **undebug all**. The resulting file looks like this:

```
AP-9105>en
Password:
AP-9105#debug traffic wired udp
  capture capture packets in pcap file
  verbose Verbose Output
  <cr>
AP-9105#debug traffic wired udp verbose
AP-9105#reading from file /dev/click_wired_log, link-type EN10MB (Ethernet)
22:35:17.1669188 IP CSCO-W-PF320YP6.lan.60354 > 239.255.255.250.3702: UDP, length 656
        0x0000: 0100 5e7f fffa 806d 971d a040 0800 4500
        0x0010: 02ac d4bb 0000 0111 cd11 c0a8 64d1 efff
        0x0020: fffa ebc2 0e76 0298 757b 3c3f 786d 6c20
        0x0030: 7665 7273 696f 6e3d 2231 2e30 2220 656e
        0x0040: 636f 6469 6e67 3d22 7574 662d 3822 3f3e
<truncated>
undebug 0x0070: 444c 4e41 444f 432f 312e 3530 2050 6c61
        0x0080: 7469 6e75 6d2f 312e 302e 342e 320d 0a4d 0x0090: 414e 3a20 2273 7364 703a 6469 7363 6f76
        0x00a0: 6572 220d 0a53 543a 2073 7364 703a 616c
        0x00b0: 6c0d 0a4d 583a 2033 0d0a 0d0a
a11
<truncated>
tcpdump: pcap_loop: error reading dump file: Interrupted system call
All possible debugging has been turned off
<end of file>
```

Output File Cleanup

Remove any information which is not part of the packet dump itself. Delete the lines containing the dump command, any prompt which contains the hostname (APname#) and any other unrelated syslog messages present in the file.

Pay special attention to the undebug command since it can be printed before a packet content as shown above. Afer the cleanup, the resulting file looks like this:

Cleanup Packet Summary Information

The start of a new packet is detected when a new offset 000000 appears. Text2pcap can handle the summary information printed before each packet, to avoid issues is best to remove them.

In Notepad++ navigate to **Search>Find** And select the **Mark** tab, ensure the **Search Mode** is **Extended**.

On the **Find what:** field enter the symbol > and click **Mark All.** This action bookmarks all lines containing the > symbol.

1	Mark		×
I	Find Replace Find in Files Find in Projects	tark	
	Eind what: 🔰	v (Mark Al
		(Clear all marks
	Bookmark line	In selection	Copy Marked Text
	Backward direction	Ć	Close
	Match whole word only	-	
	Match gase		
	Virag around		
	Search Mode	🖂 Tri	ansparency
	Normal		On losing focus
	Extended (\n, \r, \t, \0, \x)		Always
	Regular expression Imatches newline		<u> </u>

Notepad++ mark dialog box with Find what field with the chevron character inside.

After Marking the headers, Notepad++ highlights all document lines like this:



Packet dump snippet wiht highlighted line which contains the chevron.

Navigate to **Search>Bookmark** and click on **Remove bookmarked lines**. After doing so, the file looks like this snippet:

0x0000:	0100	5e7f	fffa	806d	971d	a040	0800	4500
0x0010:	02ac	d4bb	0000	0111	cd11	c0a8	64d1	efff
0x0020:	fffa	ebc2	0e76	0298	757b	3c3f	786d	6c20
0x0030:	7665	7273	696f	6e3d	2231	2e30	2220	656e

Remove starting spaces and offset colons

Navigate to Search>Find And select the Replace Tab, ensure the Search Mode is Extended.

On the **Find what:** field enter **8 white spaces**. Leave the **Replace with:** field empty and click on **Replace all**. This replaces all 8 consecutive white spaces at the start of every line with nothing, effectively deleting them. The replace dialog looks like this image.

Replace			×
Find Replace Find in Files Find in Projects Mark			
Find what:		▲ ▼ Find Next	
Replace with:		<u>R</u> eplace	
	In selection	Replace <u>A</u> ll	
Backward direction		Replace All in All Opened Doc <u>u</u> ments	
Match whole word only		Close	
🖌 Wrap around			
Search Mode		Transparenc <u>y</u>	
<u>N</u> ormal		 On losing focus 	
Extended (\n, \r, \t, \0, \x)		 Always 	
Regular expression matches newline			
Replace All: 2935 occurrences were replaced in entire file	2		.:

Notepad++ Replace dialog box with Find what field with 8 spaces.

The resulting file after this operation looks like this snippet:

0x0000:	0100	5e7f	fffa	806d	971d	a040	0800	4500
0x0010:	02ac	d4bb	0000	0111	cd11	c0a8	64d1	efff
0x0020:	fffa	ebc2	0e76	0298	757b	3c3f	786d	6c20
0x0030:	7665	7273	696f	6e3d	2231	2e30	2220	656e
0x0040:	636f	6469	6e67	3d22	7574	662d	3822	3f3e
0x0050:	3c73	6f61	703a	456e	7665	6c6f	7065	2078
0x0060:	6d6c	6e73	3a73	6f61	703d	2268	7474	703a
0x0070:	2f2f	7777	772e	7733	2e6f	7267	2f32	3030

Navigate to **Search>Find** And select the **Replace** Tab, ensure the **Search Mode** is **Extended**. Enter : (notice the blank space after the colon) on the **Find what:** field. Leave the **Replace with:** field empty and click on **Replace all**.

This replaces all colons and first spaces after the offset.

Replace		×
Find Replace Find in Files Find in Projects Mark		
Find what:	v ti v	Find Next
Replace with:		<u>R</u> eplace
	In selection	Replace <u>All</u>
Backward direction		Replace All in All Opened Documents
Match whole word only		
Match <u>c</u> ase		Close
Vrap around		
Search Mode	r 😒	ransparency
<u>N</u> ormal		On losing focus
Extended (\n, \r, \t, \0, \x)		 Always
Regular expression matches newline		
Replace All: 2935 occurrences were replaced in entire file	e	

Notepad++ Replace dialog box with Find what field filled by a colon and a space.

After the previous operation, the resulting output file looks like this snippet:

0x000001005e7ffffa806d971da040080045000x001002acd4bb00000111cd11c0a864d1efff0x0020fffaebc20e760298757b3c3f786d6c200x003076657273696f6e3d22312e302220656e0x0040636f64696e673d227574662d38223f3e0x00503c736f61703a456e76656c6f706520780x00606d6c6e733a736f61703d22687474703a0x00702f2f7777772e77332e6f72672f323030

Correct packet offset

Text2pcap expects packet offset inside each packet as a 6 character hex string, but AP packet dumps use 0x to symbolize the offset instead. To correct it navigate to **Search>Find** And select the **Replace** Tab, ensure

the Search Mode is **Extended**.

Enter **x** on the **Find what:** field. Fill the **Replace with:** field with **0** and click on **Replace all.** This replaces all x inside the offset with 0 to match the expected offset format for Text2pcap.

Replace		X
Find Replace Find in Files Find in Projects		
<u>F</u> ind what: x	V → Find Next	
Rep <u>l</u> ace with: 0	v <u>R</u> eplace	
	In selection Replace All	
Backward direction Match whole word only	Replace All in All Opened Doc <u>u</u> ments	
Match <u>c</u> ase	Close	
Wrap around		
Search Mode	Transparency	
Extended (\n, \r, \t, \0, \x)		
Regular expression matches newline		
Replace All: 2935 occurrences were replaced in e	file	

Notepad++ Replace dialog box with Find what field filled with the character x and Replace field filled with the character 0.

After the previous operation, the resulting output file looks like this snippet:

00000001005e7ffffa806d971da0400800450000001002acd4bb00000111cd11c0a864d1efff000020fffaebc20e760298757b3c3f786d6c2000003076657273696f6e3d22312e302220656e000040636f64696e673d227574662d38223f3e0000503c736f61703a456e76656c6f70652078

Separate Packet Bytes

Text2pcap data format requires for each pair of hex values to be separated by a space, an incorrect format causes Text2pcap to read packet data as an offset and fail.

Navigate to Search>Find and select the Replace Tab, ensure the Search Mode is Regular expression.

Enter ([0-9a-f][0-9a-f])([0-9a-f][0-9a-f]) (notice the leading space) on the Find what: field.

Fill the **Replace with:** field with 12 (notice the leading space) and click on **Replace all**.

The replace operation finds the hex bytes of the packet and inserts a space between each pair. The regex matches a space followed by a hex digit pair, saves them on capture group 1, then takes the adjacent pair of hex digits, saves them on capture group 2. The replacement prints both required spaces as well as the content of each capture group.

It takes multiple seconds or minutes depending on the length of the file. It utilizes a lot of RAM while running If the file is large, be patient.

Replace		×	<
Find Replace Find in Files Find in Projects M	lark		
Find what: ([0-9a-f][0-9a-f])([0-9a-f][0	-9a-f]) v t -	🔺 🔍 🔽 Find Next	
Replace with: \1 \2			
	In selection	Replace <u>A</u> ll	
Backward direction		Replace All in All Opened Documents	
Match whole word only			
Match <u>c</u> ase		Close	
🗹 Wra <u>p</u> around			
Search Mode	v 1	ransparenc <u>y</u>	
Normal		 On losing focus 	
○ Extended (\n, \r, \t, \0, \x)		Always	
Regular expression matches newline			
		U	

Notepad++ Replace dialog box with the find what filled with a regullar expression and the Replace field filled by another regular expression.

After the previous operation, the resulting output file looks like this snippet and is ready to be converted by Text2pcap.

 000000
 01
 00
 5e
 7f
 ff
 fa
 80
 6d
 97
 1d
 a0
 40
 08
 00
 45
 00

 000010
 02
 ac
 d4
 bb
 00
 01
 11
 cd
 11
 c0
 a8
 64
 d1
 ef
 ff

 000020
 ff
 fa
 eb
 c2
 0e
 76
 02
 98
 75
 7b
 3c
 3f
 78
 6d
 6c
 20

 000030
 76
 65
 72
 73
 69
 6f
 6e
 3d
 22
 31
 2e
 30
 22
 20
 65
 6e

 000040
 63
 6f
 64
 69
 6e
 67
 3d
 22
 75
 74
 66
 2d
 38
 22
 3f
 3e

 000050
 3c
 73
 6f
 61
 70
 3a
 25
 6c
 6f
 70
 3a
 74
 60
 65
 20
 78

 000060
 6d
 6c
 6

Convert the Text File to PCAP

Via Wireshark GUI

To convert the complete file to pcap, open Wireshark and navigate to **File>Import from hex dump**, a dialog box appears.

File: C:/Atsers/mariomed/Downloads/debug wired sample - Copy.log Browse Hex Dump Regular Expression Offsets: Hexadecimal Decimal Octal O Actal None Direction indication: ASCII identification: ASCII identification: ////////////////////////////////////	📕 Wireshark - Im	port From Hex Dum	P			×
Hex Dump Regular Expression Offsets: Hexadedmal Decimal Octal Octal Octal Direction indication: Image: State Sta	File: C:/Users/mar	iomed/Downloads/deb	ug wired san	ple - Copy.log		Browse
Offsets: Hexadecimal Decimal Decimal Decimal Decimal Ascell identification: Ascell identification: Timestamp format: Her/1041:165.16f No dummy header Encapsulation Encapsulation Encapsulation Detimation address: Destination address: Destination address: Destination address: Destination address: Destination address: Destination port: Scrip Tag: Destination port: Scrip Tag: Scrip	Hex Dump R	egular Expression				
Direction indication: ASCII identification: Timestamp format: VH: VM: VS. VF (No former will be applied) Encapsulation Encapsulation Encapsulation Encapsulation Encapsulation Protocol (dec): DP version: Destination address: Destination address: Destination address: Destination port: SCITP Tag: SCITP Tag: SCI	Offsets: O Her O Dec O Oct Nor	kadecimal cimal tal ne				
ASCII identification:	Direction indication	an: 🗆				
Interface name: Fake IF, Import from Hex Dump	ASCILIBERINGAU	on: ()		A. 4		
Encapsulation Type: Ethernet No dummy header Ethernet Ethertype (hex): IP Protocol (dec): Ethernet: Destination address: Destination address: Destination port: SCTP Destination port: SCTP Tag: SCTP Tag: Destination PP1: ExportPDU Dissector data	Timestamp format:	%H: %M: %S. %F		(No format will b	e applied)	
Encapsulation Type: Ethernet No dummy header Ethernet Ethertype (hex): DP Protocol (dec): Destination address: Destination address: Destination address: OUDP Source port: OTOP Destination port: SCTP Tag: SCTP Tag: SCTP Tag: Destination port: Destination port: Destina	Encapsulation					
No dummy header Ethernet Ethertype (hex): IP Protocol (dec): Source address: Destination address: Destination port: SCTP SCTP Top Destination port: SCTP (Data) PP1: Destor Destination Interface name: Fake IF, Import from Hex Dump Maximum frame length: Import Cancel	Encapsulation Typ	pe: Ethernet				~
○ Evernet Ethertype (nex): ○ IP Protocol (dec): IP version: ○ Source address: Destination address: ○ UDP Source port: Other is a state of the sta	No dummy he	sader				
Outper Source address: Destination address: Outper Source port: Outper Sctrp Tag: Sctrp (Data) PP1: ExportPDU Dissector Interface name: Fake IF, Import from Hex Dump	O IP	Ethertype (nex): Protocol (dec):			IP versions	12-4
Destination address: O UDP Source port: O TCP Destination port: O SCTP Tag: O SCTP (Data) PPI: DesportPOU Dissector data	0.5	Source address:			ar thur and to	011.0
O UDP Source port: O TCP Destination port: O SCTP Tag: O SCTP (Data) PPI: O ExportPDU Dissector Interface name: Fake IF, Import from Hex Dump Maximum frame length: Import Interface name: Fake IF, Import from Hex Dump		Destination address:				
O TCP Destination port: O SCTP Tag: O SCTP (Data) PPI: O ExportPOU Dissector Interface name: Fake IF, Import from Hex Dump Maximum frame length:	O UDP	Source port:				
O SCTP Tag: O SCTP (Data) PPI: O ExportPOU Dissector data V	O TOP	Destination port:				
O SCTP (Data) PP1: data O ExportPDU Dissector Interface name: Fake IF, Import from Hex Dump Maximum frame length: Import Import Cancel	○ SCTP	Tag:				
O ExportPOU Dissector data ✓ Interface name: Fake IF, Import from Hex Dump Maximum frame length: Import Cancel Help	O SCTP (Data)	PPI:				
Interface name: Fake IF, Import from Hex Dump Maximum frame length: Import Cancel Help	 ExportPOU 	Dissector	data			\sim
Interface name: Fake IF, Import from Hex Dump Maximum frame length: Import Cancel Help						
Maximum frame length: Import Cancel Help	Interface come:	also 15. Tennert from M	ev Demo			
Maximum trame length: Import Cancel Help	a roomace manager	unit any amport more me	ex comp			
Import Cancel Help	Maximum frame len	gth:				
				Im	port Cancel	Help

Click on the **Browse...** button and select the dump text file. Ensure the selected offset type is **Hexadecimal**, **Encapsulation type** is **Ethernet** and **No dummy header** is selected.

Click **Import** to start the conversion process.

Via command line

To convert a text file to a pcap file in the windows command line, run **<path to wireshark install folder>\text2pcap.exe <path to text file pcap> <output file path>.**

You can optionally add wireshark folder to your PATH otherwise you need to run text2pcap referencing the entire path to the text2pcap.exe every time you convert a file. Text2pcap.exe is located inside the wireshark install folder.



Windows command line output after the succesful packet dump conversion

Text2pcap also includes multiple regex options to pre-process the text file, please refer to the <u>Text2pcap</u> <u>manual page</u> for more information.

Troubleshooting

Text File is Correct but Text2pcap Cannot Read Any Packets

Text2pcap cannot read certain file encodings produced by terminal emulators commonly used (Secure CRT, Putty or others).

Change to an encoding readable by Text2pcap with Notepad++. Go to **Encoding>UTF-8** and save the file, then convert to pcap again.

File Edit	Search View	Encoding	Language	Settings	Tools	Macro	Run	Plugin	s Wind	dow ?								
) 6 8		ANS	I					⊒ ¶	, <u>=</u> </td <td>) Ø (</td> <td>₿ fx 『</td> <td>₽ ⊘</td> <td>\odot</td> <td></td> <td>▶ 🛍 </td> <td>📮 🎫 🖼</td> <td>× • 1</td> <td>* × I</td>) Ø (₿ fx 『	₽ ⊘	\odot		▶ 🛍	📮 🎫 🖼	× • 1	* × I
	1000 CONT OF 100	• UTF-	8				- 1 M - 1		<u> 24</u>		p	La serie esta esta esta esta esta esta esta est	, _ 				- 12	
193	00010	UTF-	8-BOM			a7	76	16	77	5b	02	d8	9c	07	f6	b1		
194	00011	UTF-	16 LE BOM			۱b	fd	15	15	6f	74	d7	f5	6c	30	72		
195	00012	Char	acter sets		>	C	46	1a	20	cd	e3	fb	ef	10	3f	5d		
196	00013	Com	vert to ANSI			52	9e	03	a4	a7	fd	dØ	3e	d8	18	27		
197	00014	Con	vert to UTF-8	POM		93	f9	24	16	b3	57	4f	93	e7	20	e2		
198	00015	Con	vert to UTF-1	6 BE BOM		€	d5	57	64	90	01	72	57	b9	83	6d		
199	00016	Com	vert to UTF-1	6 LE BOM		₿a	6f	f8	b5	ce	8a	8e	9a	a0	9c	42		

Notepad++ encoding menu options.

Inconsistent Offset

This error appears when the bytes of the data portion on a packet are not correctly separated into pairs, this causes Text2pcap to assume the start of a new packet and fails to interpret.

Search for any packet bytes without separation or strings in the middle of a packet content such as the undebug all command.

C:\Users\mariomed>text2pcap "C:\Users\mariomed\Downloads\debug wired sample - Copy.log" output.pcap								
Input from: C:\Users\mariomed\Downloads\debug wired sample - Copy.log								
Output to: output.pcap								
Dutput format: pcapng								
** (text2pcap:81244) 10:30:46.781149 [(none) MESSAGE] Inconsistent offset. Expecting 75, got 80. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.781712 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.782136 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.782446 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.782599 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.782748 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.782891 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.783033 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.783169 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.783319 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet								
** (text2pcap:81244) 10:30:46.783456 [(none) MESSAGE] Inconsistent offset. Expecting 10. got 10. Ignoring rest of packet								

Windows command line output after invalid file is attempted to convert. Inconsistent offset is printed to the terminal multiple times.