为Wireshark转换接入点数据包转储

目录

 「

 1

 大米条件

 少歌

 少歌

 力数据包转储

 输出文件清除

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

 1

简介

本文档介绍如何将COS接入点生成的数据包转储转换为Wireshark的PCAP格式,以解决大小限制问题。

先决条件

- 记事本++ 仅适用于Windows
- 已安装Text2pcap -包含在定期安装的Wireshark中

步骤

执行数据包转储

通过在AP命令行上运行debug traffic wired <multiple options> verbose命令,捕获AP数据包转储。 您可以在多个过滤器和接口之间进行选择。

在终端中记录会话。

执行此操作时,请注意发送最少量的击键,文件上不属于捕获本身的可打印字符越多,您在转换前 需要做的清理就越多。

最简单的方法是为数据包转储创建控制台会话,复制问题,停止转储并立即结束会话。

如果通过ssh执行转储,请使用过滤器仅捕获感兴趣的流量。否则,捕获包含ssh会话数据包。

有关如何配置捕获的完整说明,请参阅<u>COS AP故障排除</u>。

完成后,使用命令undebug all停止捕获。生成的文件如下所示:

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>

输出文件清除

删除不属于数据包转储本身的任何信息。删除包含dump命令、包含主机名(APname#)的任何提示以 及文件中存在的任何其他无关系统日志消息的行。

请特别注意undebug命令,因为它可以在数据包内容之前打印,如上所示。清除后,生成的文件如 下所示:

清除数据包摘要信息

当出现新的偏移00000时,检测到新数据包的开始。Text2pcap可以处理每个数据包之前打印的摘 要信息,为避免出现问题最好将其删除。

在记事本中++导航到搜索>查找,然后选择标记选项卡,确保搜索模式为扩展。

在查找内容:字段中输入符号>,然后单击全部标记。此操作会将包含>符号的所有行加入书签。

Mark		×
Find Replace Find in Files Find in Projects Mark		
Eind what: 🔰	V Mark Al	D
	Clear all marks	
Bookmark line	In selection Copy Marked Text	D
Backward direction	Close	D
Match whole word only		
Match gase		
virag around		
Search Mode	Transparency	7
● <u>N</u> ormal	On losing focus	
Extended (Vn, V, Vt, V0, Vc)	 Always 	
Regular expression I matches newline	· · · · ·	

"记事本++"对话框的"查找内容"字段中包含Chevron字符。

标记标题后,记事本++突出显示所有文档行,如下所示:

	·····································	🖍 debug wired sample - Copy.log 🗵
1 🔵	22:35:17.1669188 IP CSCO-W-PF320YP6.lan.60354 > 239.255.255.250.3702: UDP, length 656	
2	0x0000: 0100 5e7f fffa 806d 971d a040 0800 4500	
3	0x0010: 02ac d4bb 0000 0111 cd11 c0a8 64d1 efff	
4	0x0020: fffa ebc2 0e76 0298 757b 3c3f 786d 6c20	

突出显示的包含Chevron的行的数据包转储代码段。

导航到搜索>书签,然后单击删除带书签的行。执行此操作后,文件看起来与以下代码片段类似:

0x0000:01005e7ffffa806d971da040080045000x0010:02acd4bb00000111cd11c0a864d1efff0x0020:fffaebc20e760298757b3c3f786d6c200x0030:76657273696f6e3d22312e302220656e

删除起始空格和偏移冒号

导航到搜索>查找,然后选择替换选项卡,确保搜索模式为扩展。

在查找内容:字段上输入8个空格。将替换为:字段保留为空,然后单击全部替换。这会将每行开头的8个连续空格全部替换为nothing,从而有效地删除它们。"替换"对话框类似于此图像。

Replace			×
Find Replace Find in Files Find in Projects Mark			
Eind what:		▲ ▼ Find Next	
Replace with:			
	In selection	Replace <u>A</u> ll	
Backward direction		Replace All in All Opened Doc <u>u</u> ments	
Match case		Close	
🕑 Wrap around			
Search Mode	T 😒	ransparenc <u>v</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			

记事本++"替换"对话框为"查找包含8个空格的字段"。

此操作后生成的文件类似于以下代码段:

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

7665	7273	696f	6e3d	2231	2e30	2220	656e
636f	6469	6e67	3d22	7574	662d	3822	3f3e
3c73	6f61	703a	456e	7665	6c6f	7065	2078
6d6c	6e73	3a73	6f61	703d	2268	7474	703a
2f2f	7777	772e	7733	2e6f	7267	2f32	3030
	7665 636f 3c73 6d6c 2f2f	7665 7273 636f 6469 3c73 6f61 6d6c 6e73 2f2f 7777	7665 7273 696f 636f 6469 6e67 3c73 6f61 703a 6d6c 6e73 3a73 2f2f 7777 772e	76657273696f6e3d636f64696e673d223c736f61703a456e6d6c6e733a736f612f2f7777772e7733	76657273696f6e3d2231636f64696e673d2275743c736f61703a456e76656d6c6e733a736f61703d2f2f7777772e77332e6f	76657273696f6e3d22312e30636f64696e673d227574662d3c736f61703a456e76656c6f6d6c6e733a736f61703d22682f2f7777772e77332e6f7267	76657273696f6e3d22312e302220636f64696e673d227574662d38223c736f61703a456e76656c6f70656d6c6e733a736f61703d226874742f2f7777772e77332e6f72672f32

导航到搜索>查找,然后选择替换选项卡,确保搜索模式为扩展。在查找内容:字段中输入:(注意 冒号后面的空格)。将替换为:字段留空,然后单击全部替换。 这将替换偏移后的所有冒号和第一个空格。

Replace			\times
Find Replace Find in Files Find in Projects Mark			
<u>F</u> ind what:		▲ ▼ Find Next	
Replace with:	v Itiv		
	In selection	Replace <u>A</u> ll	
Backward direction		Replace All in All Opened	
Match whole word only			
Match <u>c</u> ase		Close	
🕑 Wrap around			
Search Mode	ו 🔽	Transparency	
<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			.:

"记事本++替换"对话框为"查找用冒号和空格填充的字段"。

在上一次操作之后,生成的输出文件类似于以下代码段:

0x000001005e7ffffa806d971da040080045000x001002acd4bb00000111cd11c0a864d1efff0x0020fffaebc20e760298757b3c3f786d6c200x003076657273696f6e3d22312e302220656e0x0040636f64696e673d227574662d38223f3e0x00503c736f61703a456e76656c6f706520780x00606d6c6e733a736f61703d22687474703a0x00702f2f7777772e77332e6f72672f323030

正确的数据包偏移

Text2pcap要求每个数据包内的数据包偏移量是6个字符的十六进制字符串,但AP数据包转储使用 0x来表示偏移量。要更正此问题,请导航到搜索>查找,然后选择替换选项卡,确保搜索模式为扩 展。

在查找内容:字段中输入x。用0填充替换为:字段,然后单击全部替换。这将用0替换偏移量内的 所有x,以匹配Text2pcap的预期偏移量格式。

Replace			×
Find Replace Find in Files Find in Projects Mark			
Eind what: 🗴	v (**	▲ ▼ Find Next	
Rep <u>l</u> ace with: 0			
	In select <u>i</u> on	Replace <u>A</u> ll	
Backward direction Match whole word only.		Replace All in All Opened Doc <u>u</u> ments	
Match case		Close	
Search Mode	V T	ransparenc <u>y</u>	
		On losing focus	
Extended (\n, \r, \t, \0, \x) Regular expression matches newline		Always	
Replace All: 2935 occurrences were replaced in entire file			.:;

记事本++"替换"对话框中的"查找用字符x填充的字段"和"替换用字符0填充的字段"。

在上一次操作之后,生成的输出文件类似于以下代码段:

00000001005e7ffffa806d971da0400800450000001002acd4bb00000111cd11c0a864d1efff000020fffaebc20e760298757b3c3f786d6c2000003076657273696f6e3d22312e302220656e00040636f64696e673d227574662d38223f3e000503c736f61703a456e76656c6f70652078

单独的数据包字节

Text2pcap数据格式要求每对十六进制值用空格分隔,不正确的格式会导致Text2pcap将数据包数据 读取为偏移量并失败。

导航到搜索>查找,然后选择替换选项卡,确保搜索模式为正则表达式。

在查找内容:字段中输入([0-9a-f][0-9a-f])([0-9a-f][0-9a-f])(请注意前导空格)。

用\1 \2填充替换为:字段(注意前导空格),然后单击全部替换。

替换操作可查找数据包的十六进制字节,并在每对字节之间插入空格。正则表达式匹配后跟十六进 制数字对的空格,将其保存在捕获组1中,然后获取相邻十六进制数字对,将其保存在捕获组2中。 替换将打印所需的空间以及每个捕获组的内容。

根据文件的长度,此过程需要几秒钟或几分钟。它在运行时会占用大量RAM。如果文件很大,请耐 心等待。

Replace		×
Find Replace Find in Files Find in Projects Mark		
Find what: ([0-9a-f][0-9a-f])([0-9a-f][0-9a-f])	v III v Find Next v	2
Rep <u>l</u> ace with: \1 \2	v <u>R</u> eplace	
	In selection Replace <u>All</u>	
Backward direction	Replace All in All Opened Documents	
Match <u>w</u> hole word only Match <u>c</u> ase	Close	
✓ Wrap around		
Search Mode	Z Transparency	
<u>N</u> ormal	 On losing focus 	
Extended (\n, \r, \t, \0, \x)	Always	
Regular expression matches newline		<u>^</u>

记事本++"替换"对话框,其中包含用正则表达式填充的查找和用另一个正则表达式填充的"替换"字段。

执行上述操作后,生成的输出文件看起来与此代码片断类似,并可通过Text2pcap进行转换。

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

 000010
 02
 ac
 d4
 bb
 00
 00
 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
 45
 6e
 76
 65
 6c
 6f
 70
 3a
 3a

 000060
 6d
 6c
 67
 3

将文本文件转换为PCAP

通过Wireshark GUI

要将完整的文件转换为pcap,请打开Wireshark并导航到文件>从十六进制转储导入,此时将显示一 个对话框。

Wireshark - In	nport From Hex Dum	ıp			×
File: C:/Users/ma	riomed/Downloads/deb	ug wired sar	nple - Copy.log		Browse
Hex Dump P	legular Expression				
Offsets: O He O De O Co No	xadecimal cimal tal ne				
Direction indicati	on: 🗌				
ASCII identificat	ion:				
Timestamp format:	%H:%M:%S.%f		Oio format will b	e applied)	
Encapsulation					
Encapsulation Ty	pe: Ethernet				~
O No dummy he	eader				
 Ethernet 	Ethertype (hex):				
O IP	Protocol (dec):			IP version:	IPv4 👳
	Source address:				
	Destination address:				
O UDP	Source port:				
O TOP	Destination port:				
○ SCTP	Tag:				
O SCTP (Data)	PPI:				
ExportPDU	Dissector	data			~
Interface name:	Fake IF, Import from H	ex Dump			
Maximum frame ler	ngth:				
			Im	port Cano	el Help

Wireshark导入对话框

单击Browse...按钮并选择转储文本文件。确保所选的偏移量类型为Hexadecimal,Encapsulation type为Ethernet,No dummy header未选中。 单击Import开始转换过程。

通过命令行

要在windows命令行中将文本文件转换为pcap文件,请运行<path to wireshark install folder>\text2pcap.exe <path to text file pcap> <output file path>。

或者,可以将wireshark文件夹添加到PATH中,否则每次转换文件时都需要运行引用 text2pcap.exe的整个路径的text2pcap。Text2pcap.exe位于wireshark安装文件夹内。



成功进行数据包转储转换后的Windows命令行输出

Text2pcap还包含多个用于预处理文本文件的正则表达式选项,有关更多信息,请参阅<u>Text2pcap手</u> 册页。

故障排除

文本文件正确,但Text2pcap无法读取任何数据包

Text2pcap无法读取常用终端仿真程序(Secure CRT、Putty或其他)生成的某些文件编码。 更改为使用记事本++的Text2pcap可读取的编码。转到Encoding>UTF-8并保存文件,然后再次转换 为pcap。

File Edit	Search View	Encoding	Language	Settings	Tools	Macro	Run	Plugin	s Win	dow ?							
0 🖰		ANS	I					⊒ d	, <u>=</u> </td <td>> M (</td> <td>ן ל_β</td> <td>₽⊘</td> <td>\odot</td> <td></td> <td>▶ 🛅 </td> <td>📮 🎫 🖼</td> <td> ▼ ≍ [</td>	> M (ן ל_β	₽⊘	\odot		▶ 🛅	📮 🎫 🖼	 ▼ ≍ [
	San our case of	• UTF-	8				-3 M.A.		يم		p		, _ 				 -
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	Con	vert to ANSI			52	9e	03	a4	a7	fd	dØ	3e	d8	18	27	
197	00014	Con	vert to UTF-8	POM) 3	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	

记事本++编码菜单选项。

不一致的偏移量

当数据包的数据部分的字节未正确分为成对时,会出现此错误,这会导致Text2pcap假设新数据包 启动,并且无法解释。

搜索数据包内容中间没有分隔的任何数据包字节或字符串,如 undebug all 命令。

:\Users\mariomed>text2pcap "C:\Users\mariomed\Downloads\debug wired sample - Copy.log" output.pcap									
nput from: C:\Users\mariomed\Downloads\debug wired sample - Copy.log									
Dutput 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									
1) (tout) an analytic formal According to the formation of the second states states of the second states of the second states states of									

尝试转换无效文件后的Windows命令行输出。不一致的偏移被多次打印到终端。

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言,希望全球的用户都能通过各 自的语言得到支持性的内容。

请注意:即使是最好的机器翻译,其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任,并建议您总是参考英文原始文档(已提供 链接)。