# Wireshark用のアクセスポイントパケットダンプ の変換

内容
<u>はじめに</u>
前提条件
<u>手順</u>
<u>パケットダンプの実行</u>
<u>出力ファイルのクリーンアップ</u>
<u>クリーンアップパケットの概要情報</u>
<u>開始スペースとオフセットコロンを削除する</u>
<u>正しいパケットオフセット</u>
<u>分離パケットバイト</u>
<u>テキストファイルをPCAPに変換する</u>
<u>Wireshark GUIを使用</u>
<u>コマンドラインを使用</u>
<u>トラブルシューティング</u>
<u>テキストファイルは正しいにも関わらず、Text2pcapがパケットを読み取れない</u>
<u>矛盾したオフセット</u>

# はじめに

このドキュメントでは、サイズ制限の回避策として、COSアクセスポイント(AP)で生成されたパ ケットダンプをWireshark用のPCAP形式に変換する方法について説明します。

# 前提条件

- メモ帳++ Windowsでのみ使用可能
- Text2pcapがインストール済み:Wiresharkの通常のインストールに含まれる

## 手順

## パケットダンプの実行

APコマンドラインでdebug traffic wired <multiple options> verbose コマンドを実行して、APパケットダンプをキャプチャします。複数のフィルタとインターフェイスから選択できます。

端末にセッションを記録します。

このとき、最小限のキーストロークを送信するように注意してください。キャプチャ自体に属さないファイル上の印刷可能な文字が多いほど、変換前にクリーンアップが必要になります。

これを行う最も簡単な方法は、パケットダンプ用のコンソールセッションを使用して、問題を複 製し、ダンプを停止して、すぐにセッションを終了することです。

ssh経由でダンプを実行している場合は、フィルタを使用して対象のトラフィックのみをキャプチャします。それ以外の場合、キャプチャにはsshセッションパケットが含まれます。

キャプチャの設定方法の詳細は、『<u>COS APのトラブルシューティング</u>』を参照してください。

作業が終了したら、undebug allコマンドを使用してキャプチャを停止します。結果のファイルは 次のようになります。

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 a]] <truncated> tcpdump: pcap\_loop: error reading dump file: Interrupted system call All possible debugging has been turned off <end of file>

#### 出力ファイルのクリーンアップ

パケットダンプ自体に含まれていない情報があれば削除します。dumpコマンドを含む行、ホスト 名(APname#)を含むプロンプト、およびファイルに存在する他の無関係なsyslogメッセージを削 除します。

undebugコマンドは、上記のようにパケットの内容の前に出力される可能性があるため、特に注 意してください。クリーンアップ後、結果のファイルは次のようになります。

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
ted>								
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		
	0x0000: 0x0010: 0x0020: 0x0030: 0x0040: ted> 0x0070: 0x0080: 0x0090: 0x00a0: 0x00b0:	0x0000: 0100 0x0010: 02ac 0x0020: fffa 0x0030: 7665 0x0040: 636f ted> 0x0070: 444c 0x0080: 7469 0x0090: 414e 0x00a0: 6572 0x00b0: 6c0d	0x0000: 0100 5e7f 0x0010: 02ac d4bb 0x0020: fffa ebc2 0x0030: 7665 7273 0x0040: 636f 6469 ted> 0x0070: 444c 4e41 0x0080: 7469 6e75 0x0090: 414e 3a20 0x00a0: 6572 220d 0x00b0: 6c0d 0a4d	0x0000: 0100 5e7f fffa 0x0010: 02ac d4bb 0000 0x0020: fffa ebc2 0e76 0x0030: 7665 7273 696f 0x0040: 636f 6469 6e67 ted> 0x0070: 444c 4e41 444f 0x0080: 7469 6e75 6d2f 0x0090: 414e 3a20 2273 0x00a0: 6572 220d 0a53 0x00b0: 6c0d 0a4d 583a	0x0000: 0100 5e7f fffa 806d 0x0010: 02ac d4bb 0000 0111 0x0020: fffa ebc2 0e76 0298 0x0030: 7665 7273 696f 6e3d 0x0040: 636f 6469 6e67 3d22 ted> 0x0070: 444c 4e41 444f 432f 0x0080: 7469 6e75 6d2f 312e 0x0090: 414e 3a20 2273 7364 0x00a0: 6572 220d 0a53 543a 0x00b0: 6c0d 0a4d 583a 2033	0x0000:01005e7ffffa806d971d0x0010:02acd4bb00000111cd110x0020:fffaebc20e760298757b0x0030:76657273696f6e3d22310x0040:636f64696e673d227574ted>0x0070:444c4e41444f432f312e0x0070:444c4e41444f432f312e0x0080:74696e756d2f312e302e0x0090:414e3a2022737364703a0x00a0:6572220d0a53543a20730x00b0:6c0d0a4d583a20330d0a	0x0000:01005e7ffffa806d971da0400x0010:02acd4bb00000111cd11c0a80x0020:fffaebc20e760298757b3c3f0x0030:76657273696f6e3d22312e300x0040:636f64696e673d227574662dted>0x0070:444c4e41444f432f312e35300x0080:74696e756d2f312e302e342e0x0090:414e3a2022737364703a64690x00a0:6572220d0a53543a207373640x00b0:6c0d0a4d583a20330d0a0d0a	0x0000:         0100         5e7f         fffa         806d         971d         a040         0800           0x0010:         02ac         d4bb         0000         0111         cd11         c0a8         64d1           0x0020:         fffa         ebc2         0e76         0298         757b         3c3f         786d           0x0030:         7665         7273         696f         6e3d         2231         2e30         2220           0x0040:         636f         6469         6e67         3d22         7574         662d         3822           ted>           y

## クリーンアップパケットの概要情報

新しいオフセット000000が表示されると、新しいパケットの開始が検出されます。Text2pcapは、各パケットの前に出力される要約情報を処理できるため、問題を回避するには、それらを削除するのが最善です。

メモ帳で++Search>Findの順に移動し、Markタブを選択して、Search ModeがExtendedになって いることを確認します。

Find what:フィールドに記号>を入力し、Mark Allをクリックします。このアクションは、>記号を 含むすべての行にブックマークを付けます。

Mark		×
Find Replace Find in Files Find in Projects Mar	k	
Eind what: 🔰	v	Mark All
		Clear al marks
Bookmark line	In selection	Copy Marked Text
Backward direction		Close
Match whole word only		
Match gase		
Wrag around		
Search Mode	S	Transparency
<u>N</u> ormal		On losing focus
Extended (\n, \r, \t, \0, \x)		Always
Regular expression     In matches newline		

Notepad++ markダイアログボックスのFind whatフィールドに山形の文字が表示されます。

ヘッダーにマークを付けた後、メモ帳++は次のようにすべてのドキュメント行を強調表示します

1 22:35:17.1669188 IP CSCO-W-PF320YP6.lan.60354 > 239.255.250.3702: 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

シェブロンを含む強調表示された行を持つパケットダンプスニペット。

Search>Bookmarkに移動し、Remove bookmarked linesをクリックします。その後、ファイルは 次のスニペットのようになります。

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

開始スペースとオフセットコロンを削除する

Search>Findに移動し、Replaceタブを選択して、検索モードがExtendedになっていることを確認します。

Find what:フィールドに、8つの空白を入力します。Replace with:フィールドを空のままにして、 Replace allをクリックします。これにより、すべての行の先頭にある8つの連続する空白が何も置 き換えられ、事実上それらを削除します。置き換えダイアログは次の図のようになります。

Replace			×
Find Replace Find in Files Find in Projects Mark			
Find what:		▲ ▼ Find Next	
Rep <u>l</u> ace 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		Transparency	
<u>N</u> ormal		<ul> <li>On losing focus</li> </ul>	
Extended (\n, \r, \t, \0, \x)		<ul> <li>Always</li> </ul>	
Regular expression     matches newline			<u>^</u>
Replace All: 2935 occurrences were replaced in entire f	ile		.:

メモ帳++検索するフィールドを8つのスペースで置き換えます。

この操作後の結果ファイルは、次のスニペットのようになります。

0x0000:01005e7ffffa806d971da040080045000x0010:02acd4bb00000111cd11c0a864d1efff0x0020:fffaebc20e760298757b3c3f786d6c200x0030:76657273696f6e3d22312e302220656e0x0040:636f64696e673d227574662d38223f3e0x0050:3c736f61703a456e76656c6f706520780x0060:6d6c6e733a736f61703d22687474703a0x0070:2f2f7777772e77332e6f72672f323030

Search>Findに移動し、Replaceタブを選択して、Search ModeがExtendedになっていることを確認します。Find what:フィールドに:(コロンの後の空白に注意してください)を入力します。 Replace with:フィールドを空のままにして、Replace allをクリックします。 これにより、オフセットの後のすべてのコロンと最初のスペースが置き換えられます。

Replace			×
Find Replace Find in Files Find in Projects Mark			
<u>F</u> ind what:		▲ ▼ Find Next	
Replace with:	v let	<u>R</u> eplace	
	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 😒	Transparency	
<u>N</u> ormal		On losing focus	
Extended (\n, \r, \t, \0, \x)		<ul> <li>Always</li> </ul>	
Regular expression matches newline			<u>^</u>
Replace All: 2935 occurrences were replaced in entire f	file		.:

メモ帳++[検索]ダイアログボックスの[検索]フィールドにコロンとスペースを入力します。

#### 前の操作の後、結果の出力ファイルは次のようになります。

0x000001005e7ffffa806d971da040080045000x001002acd4bb00000111cd11c0a864d1efff0x0020fffaebc20e760298757b3c3f786d6c200x003076657273696f6e3d22312e302220656e0x0040636f64696e673d227574662d38223f3e0x00503c736f61703a456e76656c6f706520780x00606d6c6e733a736f61703d22687474703a0x00702f2f7777772e77332e6f72672f323030

## 正しいパケットオフセット

Text2pcapでは、各パケット内のパケットオフセットを6文字の16進数文字列として想定していますが、APパケットダンプではオフセットを示すために0xを使用しています。これを修正するには、Search>Findの順に移動し、Replaceタブを選択して、Search ModeがExtendedになっていることを確認します。

Find what:フィールドにxと入力します。Replace with:フィールドに0を入力し、Replace allをク

リックします。これにより、オフセット内のすべてのxが、Text2pcapの想定されるオフセット形 式に一致するように0に置き換えられます。

Replace		×
Find Replace Find in Files Find in Projects	lark	
Eind what: x		🔺 🔍 🔻 Find Next
Replace with: 0		<u>R</u> eplace
	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
🗹 Wrap around		
Search Mode	🖂 Ti	ransparenc <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 e	ntire file	

メモ帳++[置換]ダイアログボックスの[検索する文字列]フィールドに文字xを入力し、[置換]フィールドに文字0を入力します。

#### 前の操作の後、結果の出力ファイルは次のようになります。

00000001005e7ffffa806d971da0400800450000001002acd4bb00000111cd11c0a864d1efff000020fffaebc20e760298757b3c3f786d6c2000003076657273696f6e3d22312e302220656e000040636f64696e673d227574662d38223f3e0000503c736f61703a456e76656c6f70652078

#### 分離パケットバイト

Text2pcapデータ形式では、16進数値の各ペアをスペースで区切る必要があります。形式が正し くない場合、Text2pcapはパケットデータをオフセットとして読み取り、失敗します。

Search>Findに移動してReplaceタブを選択し、Search ModeがRegular expressionになっている

#### ことを確認します。

Find what:フィールドに、([0-9a-f][0-9a-f])([0-9a-f])(先頭のスペースに注意)と入力します。

Replace with:フィールドに \1 \2(先頭のスペースに注意)を入力し、Replace allをクリックします。

置換操作では、パケットの16進数バイトが検索され、各ペアの間にスペースが挿入されます。正 規表現は、スペースの後に16進数のペアが続く部分と一致する部分をキャプチャグループ1に保 存し、その後に隣接する16進数のペアを取ってキャプチャグループ2に保存します。置き換える と、必要なスペースと各キャプチャグループの内容の両方が印刷されます。

ファイルの長さによっては、数秒または数分かかります。実行中に大量のRAMを使用しますファ イルが大きい場合は、我慢してください。

Replace	×
Find Replace Find in Files Find in Projects M	1ark
Find what: [([0-9a-f][0-9a-f])([0-9a-f][0	9a-f]) v 🖬 V Find Next 🥑
Replace with: \1 \2	v Replace
	In selection Replace <u>All</u>
Backward direction	Replace All in All Opened Doc <u>u</u> ments
Match <u>w</u> hole word only	Close
🖌 Wrap around	
Search Mode	✓ Transparency
<u>N</u> ormal	<ul> <li>On losing focus</li> </ul>
○ Extended (\n, \r, \t, \0, \x)	Always
Regular expression matches newline	

メモ帳++ [置換]ダイアログボックスが開き、正規表現で入力された内容が検索され、 [置換]フィールドに別の正規表現が入力されます。

前の操作の後、結果の出力ファイルは次のスニペットのように表示され、Text2pcapで変換する 準備が整います。

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
 45
 6e
 76
 65
 6c
 6f
 70
 65
 20
 78

 000050
 3c
 73
 6f
 61
 70
 3d
 22
 68
 74
 74
 70
 3a

 000060
 6d
 6c
 67
 33
 26
 6f
 7

## テキストファイルをPCAPに変換する

Wireshark GUIを使用

ファイル全体をpcapに変換するには、Wiresharkを開いてFile > Import from hex dumpの順に選択 すると、ダイアログボックスが表示されます。

Kireshark - Is	mport From Hex Dum	ιp			×
File: C:/Users/ma	riomed/Downloads/deb	rug wired san	ple - Copy.log		Browse
Hex Dump \$	Regular Expression				
Offsets: O He O De O Co O No	ixadecimal icimal ital ine				
Direction indicat	ion: 🗆				
ASCII identificat	tion: 🗆				
Timestamp format	1994:1994:195.196F		(No format will b	e applied)	
Encapsulation					
Encapsulation Ty	pe: Ethernet				~
O No dummy h	eader				
<ul> <li>Ethernet</li> </ul>	Ethertype (hex):				
O IP	Protocol (dec):			IP version:	IPv4 👳
	Source address:				
_	Destination address:				
O NDA	Source port:				
O TOP	Destination port:				
O SCTP	Tag:				
O SCTP (Data)	PPI:				
O ExportPDU	Dissector	data			~
Interface name:	Fake IF, Import from H	ex Dump			
Maximum frame les	ngth:				
			Im	port Cancel	Heb

Wiresharkインポートダイアログボックス

Browse...ボタンをクリックして、ダンプテキストファイルを選択します。選択したオフセットタ イプが16進数、カプセル化タイプがイーサネット、ダミーヘッダーが選択されていないことを確 認します。

Importをクリックして、変換プロセスを開始します。

#### コマンドラインを使用

テキストファイルをWindowsコマンドラインでpcapファイルに変換するには、<path to wireshark install folder>\text2pcap.exe <path to text file pcap> <output file path>を実行します。

オプションで、PATHにwiresharkフォルダを追加できます。追加しない場合は、ファイルを変換 するたびに、text2pcap.exeへのパス全体を参照するtext2pcapを実行する必要があります。 Text2pcap.exeは、wiresharkのインストールフォルダ内にあります。

PS C:\Users\mariomed\Downloads> text2pcap "debug wired sample - Copy.log" final.pcap
Input from: debug wired sample - Copy.log
Output to: final.pcap
Output format: pcapng
Read 147 potential packets, wrote 147 packets (50904 bytes including overhead).

パケットダンプ変換が成功した後のWindowsコマンドライン出力

text2pcapには、テキストファイルを前処理するための複数のregexオプションも含まれています 。詳細は、『<u>Text2pcapのマニュアルページ</u>』を参照してください。

# トラブルシューティング

テキストファイルは正しいにも関わらず、Text2pcapがパケットを読み取れない

Text2pcapは、よく使われるターミナルエミュレータ(Secure CRT, Puttyなど)で生成された特定のファイルエンコーディングを読み込めません。

Text2pcapで読み取り可能なエンコーディングにNotepad++で変更します。Encoding>UTF-8に移動してファイルを保存し、再度pcapに変換します。

File Edit	Search View	Encoding	Language	Settings	Tools	Macro	Run	Plugin	s Wind	dow ?								
ð 6 B		ANS	I					⊒ d	,≡ </td <td>) Ø (</td> <td>₿ fx [</td> <td>₽ ⊘</td> <td><math>\odot</math></td> <td></td> <td>▶ 🛅  </td> <td>📮 🎫 🖼</td> <td>≖ ▲ `</td> <td>▼ <b>≍</b> [</td>	) Ø (	₿ fx [	₽ ⊘	$\odot$		▶ 🛅	📮 🎫 🖼	≖ ▲ `	▼ <b>≍</b> [
	100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100 - 100	• UTF-	8				- 1 M - 1		<u> 24</u>			La serie esta esta esta esta esta esta esta est	, _ <del> </del>				- 12	-
193	000100	UTF-	8-BOM			a7	76	16	77	5b	02	d8	9c	07	f6	b1		
194	000110	UTF-	16 LE BOM			۱b	fd	15	15	6f	74	d7	f5	6c	30	72		
195	000120	Char	acter sets		>	C	46	1a	20	cd	e3	fb	ef	10	3f	5d		
196	000130	Con	vert to ANSI			52	9e	03	a4	a7	fd	dØ	3e	d8	18	27		
197	000140	Con	vert to UTF-8	ROM		<del>)</del> 3	f9	24	16	b3	57	4f	93	e7	20	e2		
198	000150	Con	vert to UTF-1	6 BE BOM		эb	d5	57	64	90	01	72	57	b9	83	6d		
199	000160	Con	vert to UTF-1	6 LE BOM		}a	6f	f8	b5	ce	8a	8e	9a	a0	9c	42		

メモ帳++のエンコーディングメニューオプション

## 矛盾したオフセット

このエラーは、パケットのデータ部分のバイトが正しくペアに分割されていない場合に表示され ます。この場合、Text2pcapは新しいパケットの先頭を引き継ぎ、解釈に失敗します。

undebug all コマンドなど、パケットコンテンツの途中で区切りや文字列が含まれていないパケットバイトがないかどうかを検索し ます。

:\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
utput to: output.pcap
utput format: pcapng
** (text2pcap:81244) 10:30:46.781149 [(none) MESSAGE] Inconsistent offset. Expecting 75, got 80. Ignoring rest of packet
<pre>** (text2pcap:81244) 10:30:46.781712 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
** (text2pcap:81244) 10:30:46.782136 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet
<pre>** (text2pcap:81244) 10:30:46.782446 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
<pre>** (text2pcap:81244) 10:30:46.782599 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
<pre>** (text2pcap:81244) 10:30:46.782748 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
<pre>** (text2pcap:81244) 10:30:46.782891 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
** (text2pcap:81244) 10:30:46.783033 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet
<pre>** (text2pcap:81244) 10:30:46.783169 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
<pre>** (text2pcap:81244) 10:30:46.783319 [(none) MESSAGE] Inconsistent offset. Expecting 10, got 10. Ignoring rest of packet</pre>
** (text2pcap:81244) 10:30:46.783456 [(none) MESSAGE] Inconsistent offset. Expecting 10. got 10. Ignoring rest of packet

無効なファイルを変換しようとした後のWindowsコマンドライン出力。矛盾したオフセットが端子に複数回出力されます。

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人に よる翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっ ても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性につ いて法的責任を負いません。原典である英語版(リンクからアクセス可能)もあわせて参照する ことを推奨します。