Wireshark용 액세스 포인트 패킷 덤프 변환

목차

소개

이 문서에서는 COS 액세스 포인트 생성 패킷 덤프를 크기 제한의 해결 방법으로 Wireshark용 PCAP 형식으로 변환하는 방법에 대해 설명합니다.

사전 요구 사항

- Notepad++ - Windows에서만 사용 가능
- 설치된 Text2pcap - Wireshark의 일반 설치에 포함됨

절차

패킷 덤프 수행

AP 명령줄에서 debug traffic wired <multiple options> verbose 명령을 실행하여 AP 패킷 덤프를 캡처합니다. 여러 필터와 인터페이스 중에서 선택할 수 있습니다.

터미널에 세션을 기록합니다.
이때 키 입력을 가장 적게 보내도록 주의해야 합니다. 캡처에 속하지 않는 파일의 인쇄 가능한 문자가 많을수록 변환하기 전에 더 많은 정리를 수행해야 합니다.
가장 쉬운 방법은 패킷 덤프에 대한 콘솔 세션으로 문제를 복제하고 덤프를 중지한 후 즉시 세션을 종료하는 것입니다.

ssh를 통해 덤프를 수행하는 경우 필터를 사용하여 원하는 트래픽만 캡처합니다. 그렇지 않으면 캡처에 ssh 세션 패킷이 포함됩니다.

캡처 구성 방법에 대한 전체 지침은 COS AP 트러블슈팅을 참조하십시오.

완료되면 undebug all 명령을 사용하여 캡처를 중지합니다. 결과 파일은 다음과 같습니다.

```
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
all     0x00b0:  6c0d 0a4d 583a 2033 0d0a 0d0a
<truncated>
tcpdump: pcap_loop: error reading dump file: Interrupted system call
All possible debugging has been turned off
<end of file>
```

출력 파일 정리

패킷 덤프 자체에 속하지 않은 모든 정보를 제거합니다. dump 명령이 포함된 줄, 호스트 이름(APname#)이 포함된 프롬프트 및 파일에 있는 다른 관련 없는 syslog 메시지를 삭제합니다.

위와 같이 undebug 명령은 패킷 내용 앞에 인쇄할 수 있으므로 특별히 주의하십시오. 정리 후 결과 파일은 다음과 같습니다.

```
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>
        0x0070:  444c 4e41 444f 432f 312e 3530 2050 6c61
```

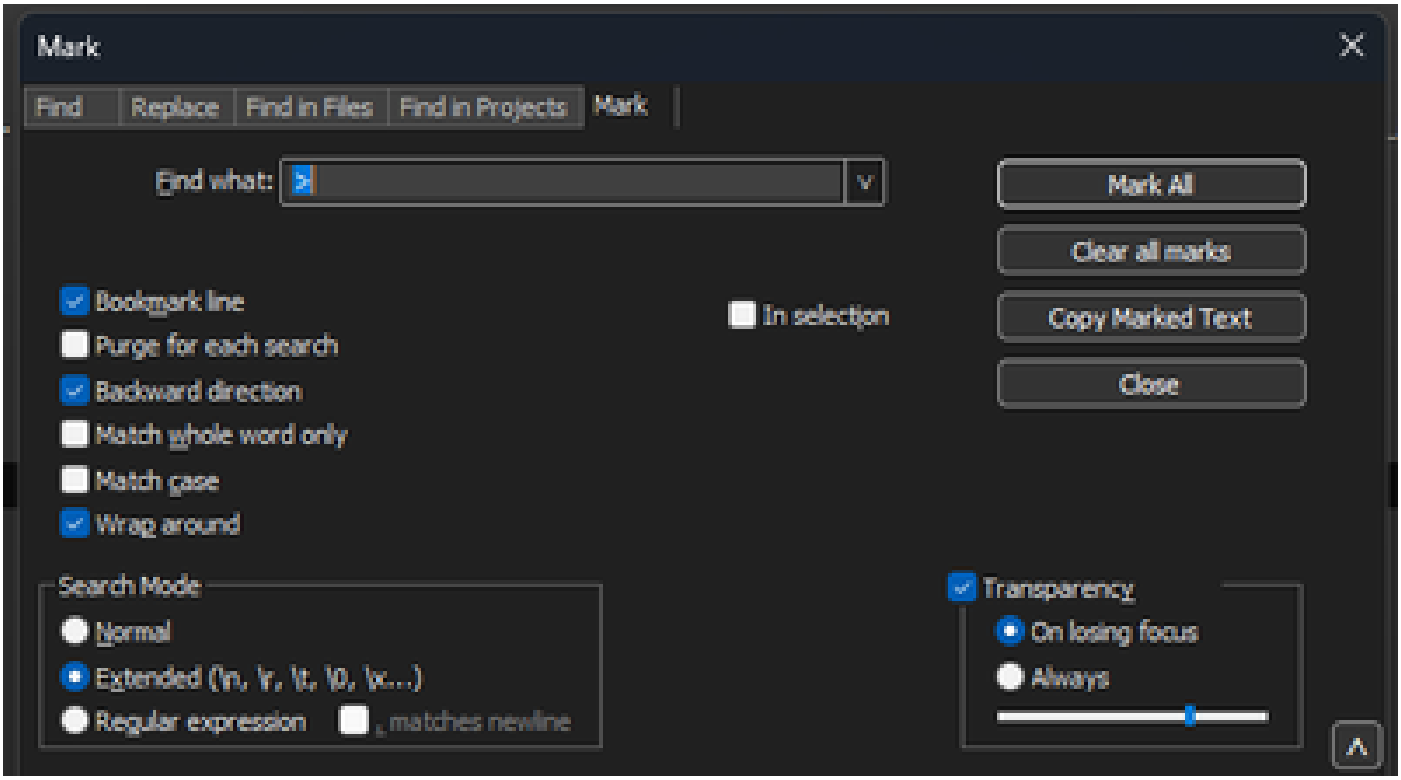
```
0x0080:  7469 6e75 6d2f 312e 302e 342e 320d 0a4d
0x0090:  414e 3a20 2273 7364 703a 6469 7363 6f76
0x00a0:  6572 220d 0a53 543a 2073 7364 703a 616c
0x00b0:  6c0d 0a4d 583a 2033 0d0a 0d0a
```

패킷 정리 요약 정보

새 패킷의 시작은 새 오프셋 값이 나타날 때 000000. Text2pcap는 각 패킷 앞에 인쇄된 요약 정보를 처리할 수 있으므로 문제를 방지하는 것이 가장 좋습니다.

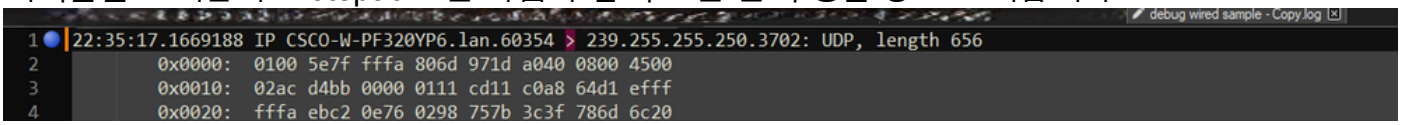
Notepad(메모장)++ Search(검색)>Find(찾기)로 이동하고 Mark(표시) 탭을 선택한 다음 Search Mode(검색 모드)가 Extended(확장)인지 확인합니다.

Find what: 필드에 기호를 >입력하고 Mark All(모두 표시)을 클릭합니다. 이 작업을 수행하면 > 기호가 포함된 모든 행에 책갈피가 지정됩니다.



Chevron 문자가 포함된 Find what 필드를 포함하는 Notepad++ mark 대화 상자

머리글을 표시한 후 Notepad++는 다음과 같이 모든 문서 행을 강조 표시합니다.



펼침 단추를 포함하는 강조 표시된 줄이 있는 패킷 덤프 코드 조각입니다.

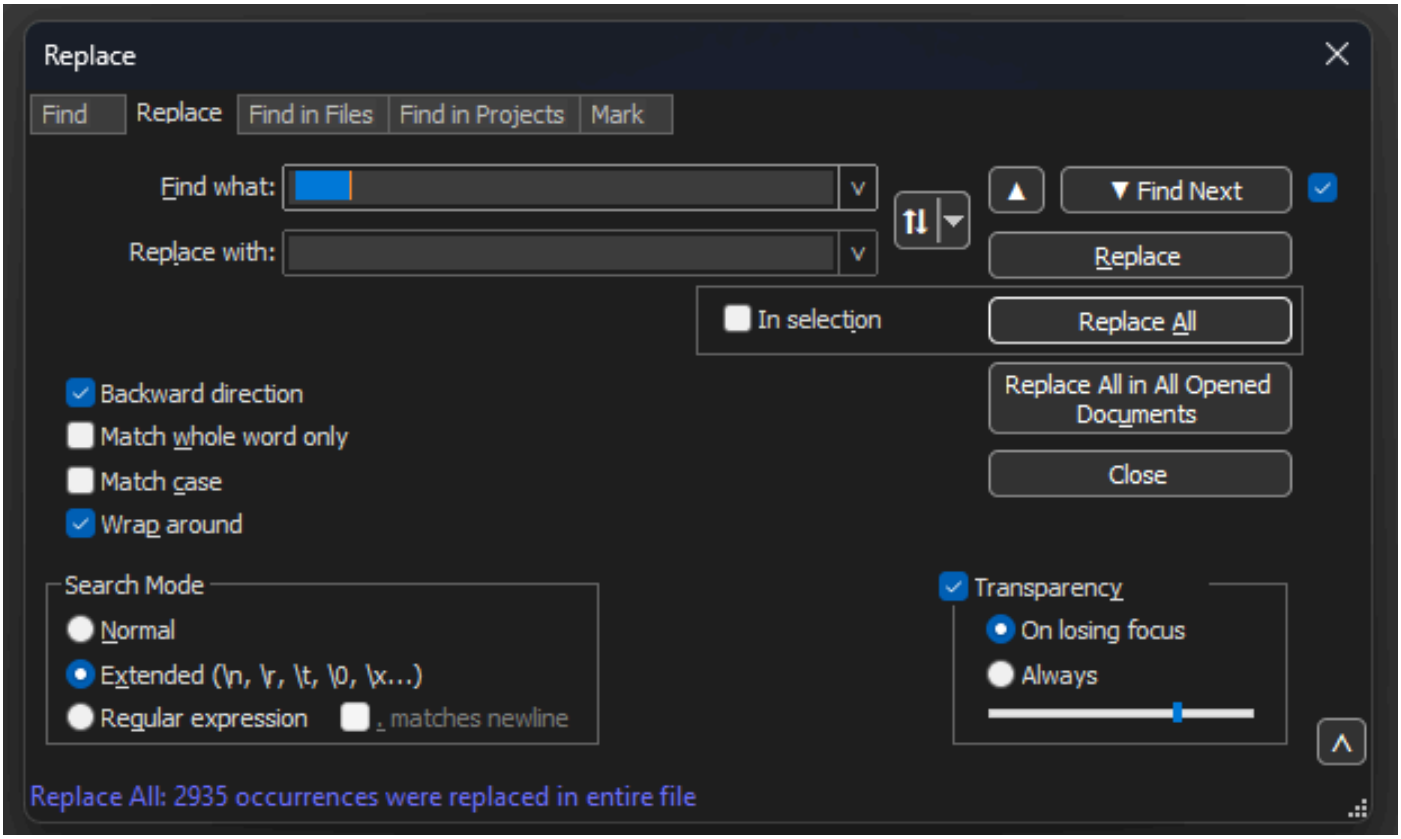
Search(검색)>Bookmark(책갈피)로 이동하고 Remove bookmarked lines(책갈피 라인 제거)를 클릭합니다. 이렇게 하면 파일이 다음과 같이 표시됩니다.

```
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
```

시작 공백 및 오프셋 콜론 제거

검색>찾기로 이동하고 대체 탭을 선택하여 검색 모드가 확장되었는지 확인합니다.

Find what: 필드에 8개의 공백을 입력합니다. Replace with:(바꿀 내용:) 필드를 비워 두고 Replace all(모두 교체)을 클릭합니다. 이렇게 하면 모든 줄의 시작 부분에 있는 8개의 연속 공백이 모두 공백으로 대체되어 효과적으로 삭제됩니다. 바꾸기 대화 상자가 이 이미지와 같습니다.

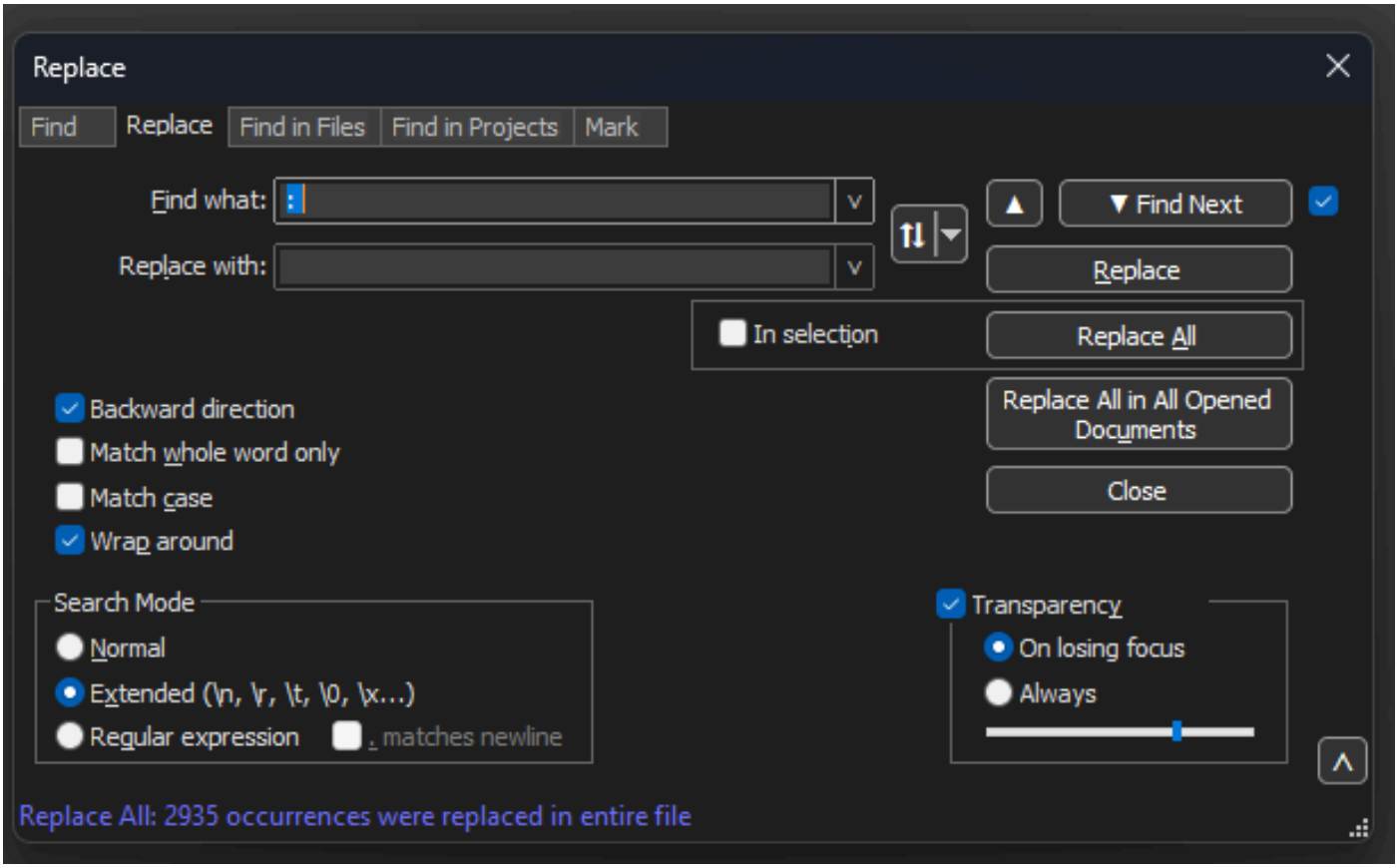


Notepad++ Replace(메모장) 대화 상자를 Find what(찾을 내용) 필드에 공백을 8개 입력합니다.

이 작업 후의 결과 파일은 다음과 같습니다.

```
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
```

검색>찾기로 이동하고 대체 탭을 선택하고 검색 모드가 확장되었는지 확인합니다. Find what: 필드에 다음을 입력합니다(콜론 뒤에 공백이 있음). Replace with: 필드를 비워두고 Replace all(모두 바꾸기)을 클릭합니다.
이렇게 하면 오프셋 뒤의 모든 콜론 및 첫 번째 공백이 교체됩니다.



메모장++ 바꾸기 대화 상자를 콜론과 공백으로 채워진 필드를 찾습니다.

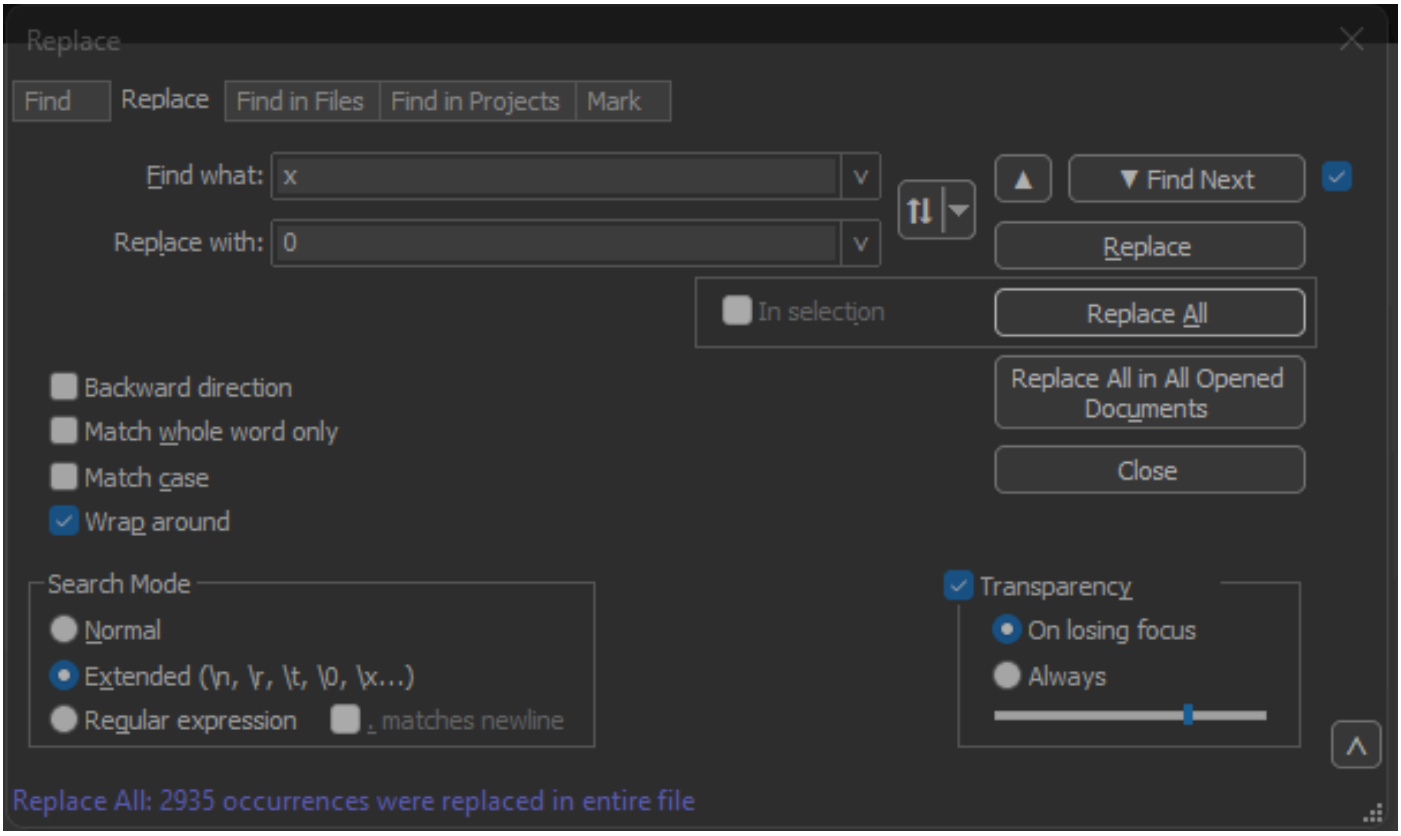
이전 작업 후 결과 출력 파일은 다음과 같이 표시됩니다.

```
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
```

올바른 패킷 오프셋

Text2pcap는 각 패킷 내부에 패킷 오프셋을 6자 16진수 문자열로 예상하지만, AP 패킷 덤프에서는 0x를 사용하여 오프셋을 대신 상징합니다. 수정하려면 검색>찾기로 이동하고 대체 탭을 선택합니다. 검색 모드가 확장되었는지 확인합니다.

Find what: 필드에 x를 입력합니다. Replace with: 필드를 0으로 채우고 Replace all(모두 대체)을 클릭합니다. 이렇게 하면 오프셋 내부의 모든 x가 0으로 대체되어 Text2pcap의 예상 오프셋 형식과 일치합니다.



Notepad++ Replace(메모장) 대화 상자를 Find what field filled with the character x(문자 x로 채워진 필드 찾기) 및 Replace field filled with the character 0(문자 0으로 채워진 필드 바꾸기)로 바꿉니다.

이전 작업 후 결과 출력 파일은 다음과 같이 표시됩니다.

```
000000 0100 5e7f fffa 806d 971d a040 0800 4500
000010 02ac d4bb 0000 0111 cd11 c0a8 64d1 efff
000020 fffa ebc2 0e76 0298 757b 3c3f 786d 6c20
000030 7665 7273 696f 6e3d 2231 2e30 2220 656e
000040 636f 6469 6e67 3d22 7574 662d 3822 3f3e
000050 3c73 6f61 703a 456e 7665 6c6f 7065 2078
```

개별 패킷 바이트

Text2pcap 데이터 형식을 사용하려면 각 16진수 값 쌍을 공백으로 구분해야 합니다. 형식이 잘못되면 Text2pcap가 패킷 데이터를 오프셋으로 읽고 실패합니다.

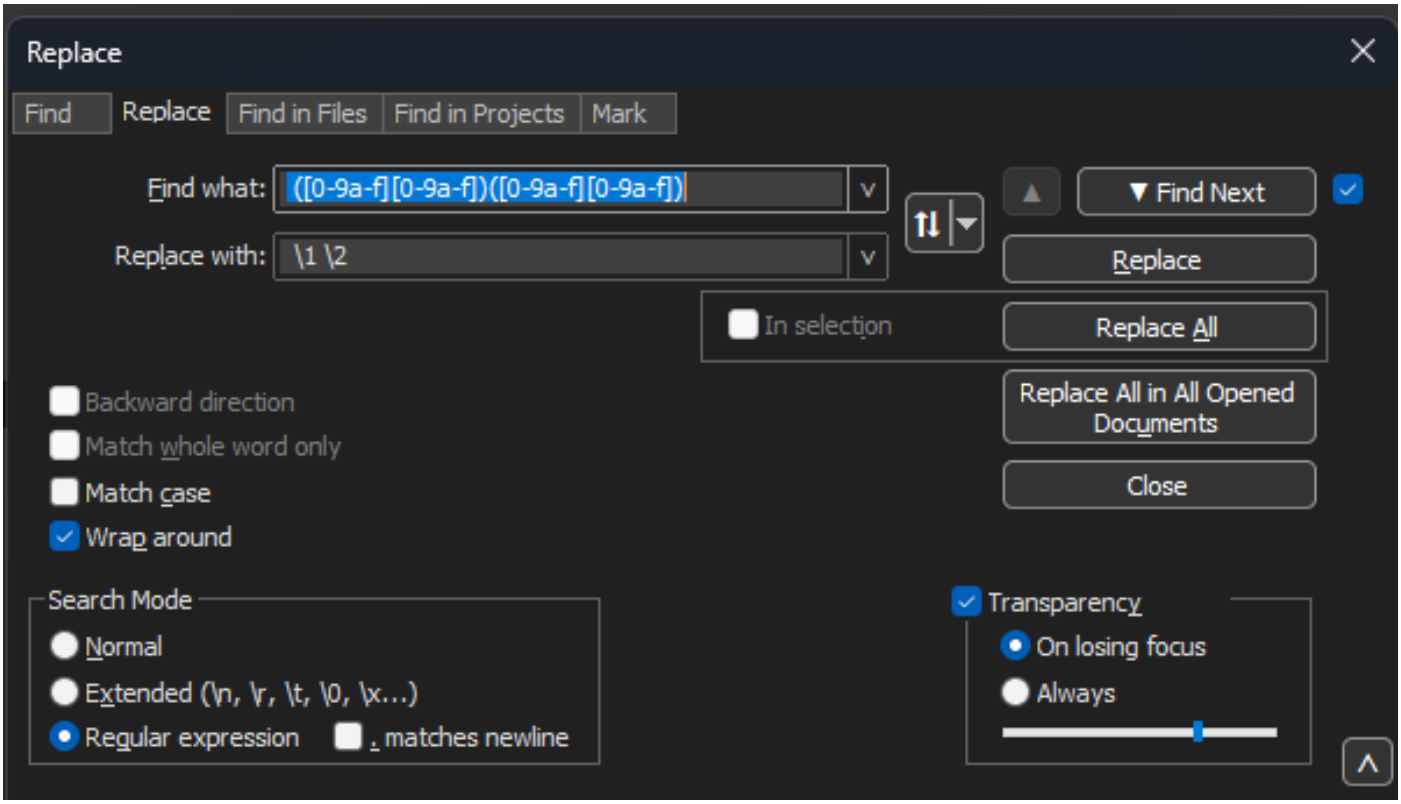
검색>찾기로 이동하고 대체 탭을 선택하고 검색 모드가 정규식인지 확인합니다.

Find what: 필드에 ([0-9a-f][0-9a-f])([0-9a-f][0-9a-f])를 입력합니다(선행 공백 참조).

바꿀 내용: 필드를 \1 \2로 채우고(맨 앞 부분 참조) 모두 바꾸기를 클릭합니다.

교체 작업은 패킷의 16진수 바이트를 찾고 각 쌍 사이에 공백을 삽입합니다. regex는 공백을 매칭하고 그 뒤에 16진수 쌍을 추가하여 캡처 그룹 1에 저장한 다음 인접한 16진수 쌍을 가져와 캡처 그룹 2에 저장합니다. 대체는 각 캡처 그룹의 내용뿐만 아니라 필수 공백을 모두 인쇄합니다.

파일의 길이에 따라 몇 초 또는 몇 분이 소요됩니다. 실행 중 많은 RAM을 사용합니다. 파일이 큰 경우 인내심을 가지십시오.



메모장++ 바꾸기 대화 상자에는 정규식으로 채워진 항목과 다른 정규식으로 채워진 바꾸기 필드가 있습니다.

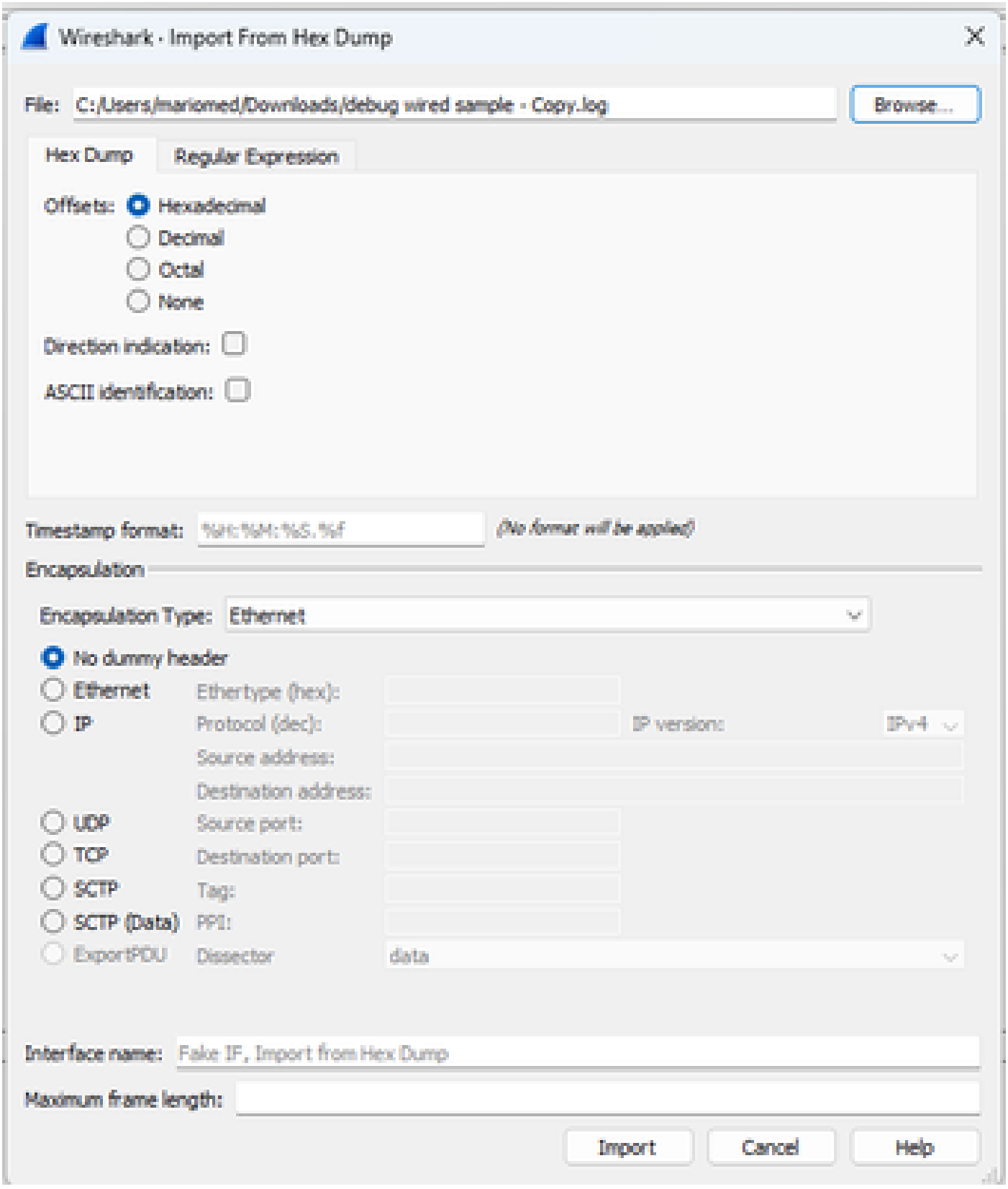
이전 작업 후 결과 출력 파일은 이 코드 조각과 비슷하며 Text2pcap로 변환할 준비가 되었습니다.

```
000000 01 00 5e 7f ff fa 80 6d 97 1d a0 40 08 00 45 00
000010 02 ac d4 bb 00 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
000060 6d 6c 6e 73 3a 73 6f 61 70 3d 22 68 74 74 70 3a
000070 2f 2f 77 77 77 2e 77 33 2e 6f 72 67 2f 32 30 30
000080 33 2f 30 35 2f 73 6f 61 70 2d 65 6e 76 65 6c 6f
000090 70 65 22 20 78 6d 6c 6e 73 3a 77 73 61 3d 22 68
```

텍스트 파일을 PCAP로 변환

Wireshark GUI 사용

전체 파일을 pcap로 변환하려면 Wireshark를 열고 File>Import from hex dump(파일>16진수 덤프에서 가져오기)로 이동하면 대화 상자가 나타납니다.



Wireshark 가져오기 대화 상자

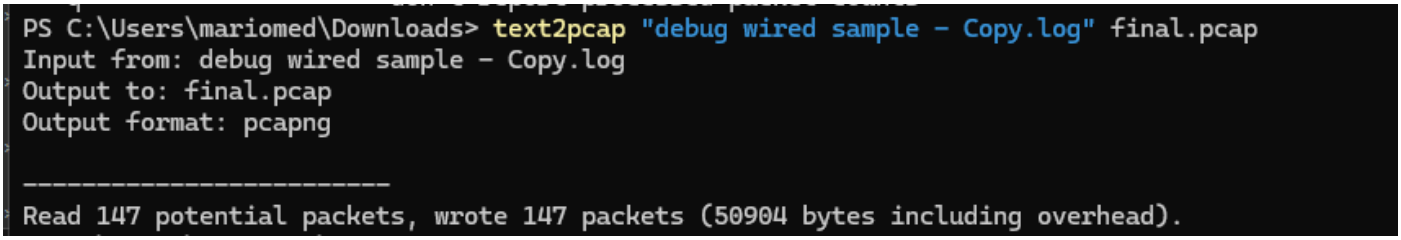
Browse...(찾아보기...) 버튼을 클릭하고 덤프 텍스트 파일을 선택합니다. 선택한 오프셋 유형이 16진수, 캡슐화 유형이 이더넷 및 No dummy header가 선택되었는지 확인합니다.

변환 프로세스를 시작하려면 Import(가져오기)를 클릭합니다.

명령줄을 통해

텍스트 파일을 Windows 명령줄에서 pcap 파일로 변환하려면 <path to wireshark install folder>\text2pcap.exe <path to text file pcap> <output file path>를 실행합니다.

선택적으로 wireshark 폴더를 PATH에 추가할 수 있습니다. 그렇지 않으면 파일을 변환할 때마다 text2pcap.exe에 대한 전체 경로를 참조하는 text2pcap를 실행해야 합니다. Text2pcap.exe는 wireshark 설치 폴더 내에 있습니다.



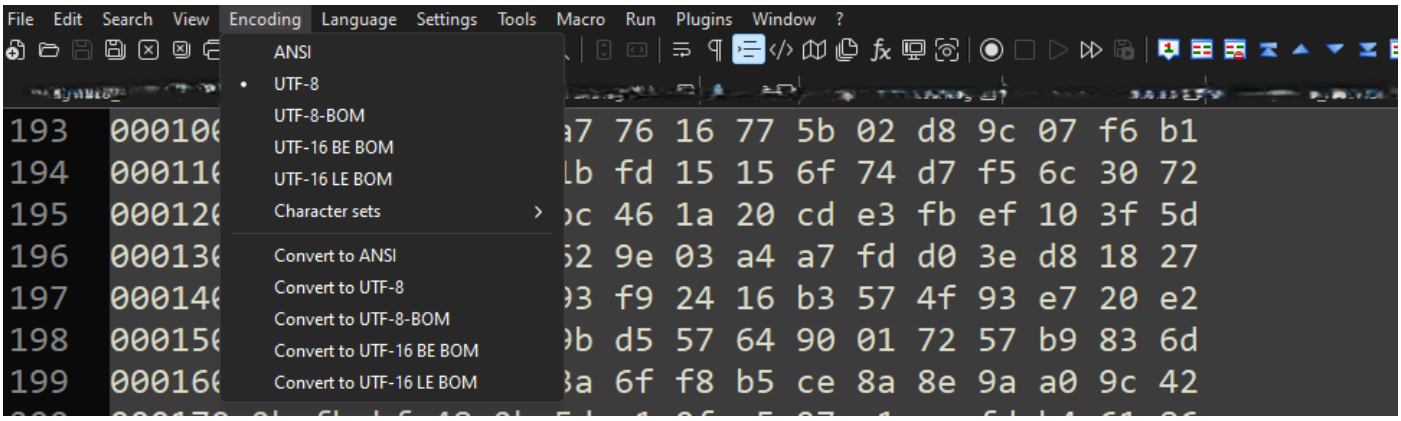
성공적인 패킷 덤프 변환 후 Windows 명령줄 출력

Text2pcap에는 텍스트 파일을 전처리하는 여러 regex 옵션도 포함되어 있습니다. 자세한 내용은 Text2pcap 매뉴얼 페이지를 참조하십시오.

문제 해결

텍스트 파일이 올바르지만 Text2pcap에서 패킷을 읽을 수 없습니다.

Text2pcap는 일반적으로 사용되는 터미널 에뮬레이터에서 생성하는 특정 파일 인코딩을 읽을 수 없습니다(Secure CRT, Putty 등).
Text2pcap with Notepad++에서 읽을 수 있는 인코딩으로 변경합니다. Encoding(인코딩) >UTF-8(UTF-8)로 이동하여 파일을 저장한 다음 pcap로 다시 변환합니다.



메모장++ 인코딩 메뉴 옵션.

불일치 오프셋

이 오류는 패킷에 있는 데이터 부분의 바이트가 쌍으로 올바르게 구분되지 않을 때 나타납니다. 이로 인해 Text2pcap가 새 패킷의 시작을 가정하고 해석하지 못합니다.
명령과 같은 패킷 내용의 중간에 있는 문자열이나 분리 없이 패킷 바이트를 undebug all 검색합니다.

```
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
Output 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
```

잘못된 파일을 변환하려고 하면 *Windows* 명령줄 출력이 표시됩니다. 일관되지 않은 오프셋이 터미널에 여러 번 인쇄됩니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.