Laboratorium - Użycie z programu Wireshark do przechwytywania danych pochodzących z protokołu FTP.

Topologia – FTP

W tym ćwiczeniu omówimy przechwytywanie danych TCP w sesji FTP . Topologia składa się z komputera z dostępem do Internetu.



Cele

Identyfikacja pól w nagłówku TCP oraz operacji przychwytywania przez Wireshark sesji FTP

Scenariusz

Dwa protokoły w warstwie transportowej TCP/IP, to TCP zdefiniowane w RFC 761 oraz UDP zdefiniowane w RFC 768. Oba protokoły obsługują komunikację protokołów wyższych warstw. Na przykład, TCP jest używany do obsługi warstwy transportowej między innymi, dla protokołów Hypertext Transfer Protocol (HTTP) i FTP. Protokół UDP działa w warstwie transportowej i współpracuje między innymi z protokołami Domain Name System (DNS).

Uwaga: Zrozumienie zawartości nagłówków TCP i UDP oraz ich działania jest bardzo istotne dla inżynierów sieciowych.

W laboratorium będziesz używał programu typu Open Source Wireshark w celu przechwytywania i analizowania pól nagłówka protokołu TCP dla sesji FTP przesyłania plików pomiędzy komputerem hosta i anonimowym serwerem FTP. Aby połączyć się z anonimowym serwerem FTP i pobrać plik będzie używany Wiersz poleceń systemu Windows.

Wymagane zasoby - FTP

1 PC (Windows 7, Vista, lub XP z dostępem do wiersza poleceń, dostępem do Internetu i zainstalowanym programem Wireshark)

Część 1: Identyfikacja pól w nagłówku segmentu TCP oraz obserwacja działania w sesji FTP za pomocą programu Wireshark.

W części 1 należy użyć programu Wireshark do przechwytywania sesji FTP i sprawdzenia pól nagłówka TCP.

Krok 1: W programie Wireshark uruchom przechwytywanie.

- a. Zamknij wszystkie zbędne komunikacje w sieci, takie jak na przykład przeglądarki WWW, aby ograniczyć ilość ruch podczas przechwytywania pakietów w programie Wireshark.
- b. W programie Wireshark uruchom przechwytywanie.

Krok 2: Pobierz plik Readme.

- a. W wierszu poleceń wpisz ftp ftp.cdc.gov.
- b. Zaloguj się na stronę FTP Centers for Disease Control and Prevention (CDC) używając konta anonymous bez hasła.
- c. Znajdź i pobierz plik Readme.

```
C:\Users\userl}ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FIP Service
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 Anonymous user logged in.
ftp> 1s
200 PORT command successful.
150 Opening ASCI mode data connection for file list.
aspnet_client
pub
Readme
Siteinfo
up.htm
w3c
web.config
welcome.msg
226 Transfer complete.
ftp: 76 bytes received in 0.00Seconds 19.00Kbytes/sec.
ftp> get Readme
200 PORT command successful.
150 Opening ASCII mode data connection for Readme(1428 bytes).
226 Transfer complete.
ftp: 1428 bytes received in 0.01Seconds 204.00Kbytes/sec.
ftp> guit
```

Krok 3: Zatrzymaj przechwytywanie w programie Wireshark.

Krok 4: Przejdź do okna głównego programu Wireshark.

Wireshark przechwycił wiele pakietów w trakcie sesji FTP do strony ftp.cdc.gov. Aby ograniczyć ilość danych do analizy, w polu **Filter: entry** wpisz **tcp oraz ip.addr == 198.246.112.54** i kliknij **Apply**. Adres IP 198.246.112.54 jest adresem strony ftp.cdc.gov.

<u>F</u> ile	<u>E</u> dit <u></u>	/iew _	<u>G</u> o <u>(</u>	<u>C</u> aptu	re <u>A</u>	nalyze	<u>S</u> tati:	tics	Telep	hony	<u>T</u> o	ols	Interna	als <u>H</u> el	р							
	<u> </u>		× I		2	1	8	Q	\$	i	\$ }	7 1	<u></u> [] €	Q	11	**	¥	1	8 %	
Filter	tcp a	nd ip.ad	ldr ==	: 198.2	46.112	2.54						[▼ Ex	pression	Cl	ear A	Apply	Save				
No.	Time		S	ource			Dest	nation			Pro	tocol	Len	igth In	fo							
5	5 1.13	67160	00 1	92.1	168.1	1.17	198	.246	.112	. 54	TC	P		66 4	9243	> f	tp [[SYN]	Seq=	0 W	/in=8	192 Ler
7	1.22	65020	00 1	.98.2	246.1	L12.5	4 192	.168	.1.1	.7	TC	Р		66 f	tp >	492	43 [SYN,	ACK]	Se	q=0	Ack=1 W
8	31.22	66270	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	TC	Р		54 4	9243	> f	tp [ACK]	Seq=	1 A	ck=1	Win=81
9	91.31	45680	00 1	.98.2	246.1	12.5	4 192	.168	.1.1	.7	FT	Р		81 R	espo	nse:	220) Mic	rosof	t F	TP S	ervice
10) 1.52	33720	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	TC	Р		54 4	9243	> f	tp [ACK]	Seq=	1 A	ck=2	8 Win=8
12	2 4.58	51850	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	FT	Р		70 R	eque	st:	USER	t ano	nymou	s		
13	3 4.67	50400	00 1	.98.2	246.1	12.5	4 192	.168	.1.1	.7	FT	Р		126 R	espo	nse:	331	. Ano	nymou	s a	cces	s allow
14	4.87	72450	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	TC	Р		54 4	9243	> f	tp [ACK]	Seq=	17	Ack=	100 Wir
19	9 5.96	15140	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	FT	Р		61 R	eque	st:	PASS	;				
20	6.04	89290	00 1	.98.2	246.1	12.5	4 192	.168	.1.1	.7	FT	Р		85 R	espo	nse:	230) Ano	nymou	s u	ser	logged
21	6.25	00830	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	TC	Р		54 4	9243	> f	tp [ACK]	Seq=	24	Ack=	131 Wir
25	5 8.85	52250	00 1	.92.1	.68.1	1.17	198	.246	.112	. 54	FT	Р		80 R	eque	st:	PORT	192	,168,	1,1	7,19	2,92
•																						-
Er	amo 5	· 66	hyt.	as 01	n wii	ro (5	28 h	ite)	66	hyt	05	cant	urod	(528	hite	.) 0	n in	torfa				
	horno	+ TT	Sr	с• н/	nHa	i Dr h	a.15	•63	(an.)	1090	.es 5.h	د apt ۵۰15	.ur eu	Det	• Not		r 00	·c5·7	$\frac{1}{2}$ (30		5·0a·	00.05.
	terne	t Pro	toc	1 v	orsi	on 4	Src	103	2 16	90.0 8 1	17	(107	168	1 17		:geai	108	246 1	12 54	1 (1	108 7	46 112
	ansmi	ssion		otro		ot oc o	1 5	C P	nrt.	497	43	(492	243)	Dst F	Port	ft	n (2	1) 5	Seq. (, (-) I	en.	0
	anomi	33101					, 5	C P	<i></i>	492		(432	.+J),	DSCI	- or c		р (<u>с</u> .	1), 3	eq. u	/, I	Len.	•
0000	30 4	16 9a	99	c5 7	2 90) 4c	e5 k	e 15	63	08	00 4	15 0	0	0Fr	·.∟ .	с.	.E.					
0010	00	34 03	d8	40 0	0 80	06	fe ()5 c0	a8	01	11 0	6 f	6	.4@.	·· ·		• • •					
0020	20	36 CU	20	00 1	.5 41	9e	03 0	a 00	00	00	00 8	SU 0	2	po.[0	• • • •	• • •					
0040	04	00 45 02	21	00 0	0 02	. 04	04 6	.c 01	. 05	05	00 (л U	1		•••••		• • •					

Krok 5: Przeanalizuj zawartości pól TCP.

Po zastosowaniu filtru TCP pierwsze trzy ramki w okienku listy pakietów (górna sekcja) wyświetla w warstwie transportowej protokół TCP służący do tworzenia niezawodnej sesji. Sekwencja [SYN], [SYN, ACK], [ACK] ilustruje 3-etapowe uzgodnienie.

5 1.136716000	192.168.1.17	198.246.112.54	ТСР	66 49243 > ftp [SYN] Seq=0 Win=8192 L
7 1.226502000	198.246.112.54	192.168.1.17	TCP	66 ftp > 49243 [SYN, ACK] Seq=0 Ack=1
8 1.226627000	192.168.1.17	198.246.112.54	TCP	54 49243 > ftp [ACK] Seq=1 Ack=1 Win=8

TCP jest rutynowo używany podczas sesji do kontroli dostarczenia datagramu, weryfikacji jego dotarcia i zarządzania rozmiarem okna. Dla każdej wymiany danych pomiędzy klientem a serwerem FTP jest uruchomiana nowa sesja TCP. Na zakończenie wymiany danych sesja TCP jest zamykana. Gdy sesja FTP jest zakończona, to TCP wykonuje procedurę zamknięcia i zakończenia połączenia.

Szczegółowe informacje na temat TCP są dostępne w panelu szczegółów pakietów programu Wireshark (sekcja środkowa). Podświetl pierwszy datagram TCP z komputera hosta i rozwiń rekord TCP. Rozwinięty datagram TCP wydaje się być podobny do okienka szczegółów pakietu (packet detail) widocznego poniżej.

Laboratorium - Przechwytywanie danych pochodzących z FTP programem Wireshark

÷	Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
÷	Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
÷	Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.112.54 (198.246.112.54)
	Transmission Control Protocol, Src Port: 49243 (49243), Dst Port: ftp (21), Seq: 0, Len: 0
	Source port: 49243 (49243)
	Destination port: ftp (21)
	[Stream index: 0]
	Sequence number: 0 (relative sequence number)
	Header length: 32 bytes
	🗆 Flags: 0x002 (SYN)
	000 = Reserved: Not set
	0 = Nonce: Not set
	0 = Congestion window Reduced (CWR): Not set
	0 = ECN-Echo: Not set
	0 = Acknowledgment: Not set
	0 = Push: Not set
	0 = Reset: Not set
	🖬1. = Syn: Set
	0 = Fin: Not set
	Window size value: 8192
	[Calculated window size: 8192]
	B Checksum: 0x4321 [validation disabled]
	B Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No



Powyższy rysunek przedstawia schemat datagramu TCP. Opis każdego pola znajduje się w dokumencie:

- Numer portu źródłowego TCP przypisany jest do sesji TCP hosta, który otworzył połączenie. Liczba ta
 jest zwykle wartością losową powyżej 1023.
- Numer portu docelowego jest używany w celu określenia protokołu warstwy wyższej bądź aplikacji na komputerze docelowym (serwerze). Wartości z zakresu 0-1023 reprezentują "dobrze znane porty" i związane są z popularnymi usługami i aplikacjami (w sposób opisany w dokumencie RFC 1700, takimi jak Telnet, FTP, HTTP oraz innymi). Kombinacja czterech wartości (źródłowego adresu IP, źródłowego numeru portu, docelowego adresu IP, docelowego numeru portu) identyfikuje w sposób unikalny sesję dla obu hostów: klienta i serwera.

Uwaga: W programie Wireshark przechwycony poniżej port docelowy to 21, co oznacza że jest to FTP. Serwery FTP na porcie 21 nasłuchują połączenia od klienta FTP.

- Numer sekwencyjny określa numer ostatniego oktetu w segmencie.
- Numer potwierdzenia określa numer następnego oktetu oczekiwanego przez odbiorcę.
- Bity kontrolne (flagi) mają specjalne znaczenie w zarządzaniu sesją i w określaniu sposobu traktowania segmentów. Wśród nich wyróżniamy:
 - ACK bit/flaga potwierdzenia otrzymania segmentu,

- SYN bit/flaga synchronizacji, ustawiona tylko wtedy, gdy nowa sesja jest negocjowana podczas trójetapowego uzgadniania,
- FIN bit/flaga zakończenia, która oznacza żądanie zamknięcia sesji.
- **Rozmiar okna** to wartość rozmiaru okna przesuwnego, oznaczająca ile oktetów może być przesłanych zanim nadawca będzie musiał czekać na potwierdzenie.
- Wskaźnik Urgent jest używany tylko z flagą Urgent (URG), gdy nadawca musi wysłać pilne dane do odbiornika.
- Options ma obecnie tylko jedną możliwość i jest określona jako maksymalna wielkość segmentu TCP (wartość opcjonalna).

Używając programu Wireshark przechwyć pierwszą fazę w sesji TCP (flaga SYN ustawiona na 1) i wypełnij informację o nagłówku datagramu TCP:

Z komputera PC do serwera CDC (tylko flaga SYN jest ustawiona na 1):

Adres IP nadawcy:	
Adres docelowy IP:	
Numer portu źródłowego:	
Numer portu docelowego	
Numer sekwencyjny:	
Numer potwierdzenia:	
Długość nagłówka:	
Rozmiar okna:	

W drugim filtrowanym przechwytywaniu Wiresharka serwer CDC FTP potwierdza żądanie z komputera PC. Zanotuj wartości bitów SYN i ACK.



Wypełnij następujące informacje dotyczące wiadomości SYN-ACK.

Źródłowy adres IP:	
Adres docelowy IP:	
Numer portu źródłowego:	
Numer portu docelowego:	
Numer sekwencyjny:	
Numer potwierdzenia:	
Długość nagłówka:	
Rozmiar okna:	

W końcowej fazie negocjacji w celu nawiązania komunikacji, komputer wysyła do serwera komunikat potwierdzający. Zauważ, że tylko bit ACK jest ustawiony na 1, a numer sekwencyjny został zwiększony do 1.

⊕ Frame 8: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0 ⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72) ⊕ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.112.54 (198.246.112.54) □ Transmission Control Protocol, Src Port: 49243 (49243), Dst Port: ftp (21), Seq: 1, Ack: 1, Len: 0 Source port: 49243 (49243) Destination port: ftp (21) [Stream index: 0] Sequence number: 1 (relative sequence number) Acknowledgment number: 1 (relative ack number) Header length: 20 bytes □ Flags: 0x010 (ACK) 000. = Reserved: Not set ...0 = Nonce: Not set 0... = Congestion Window Reduced (CWR): Not set0.. = ECN-Echo: Not set0. = Urgent: Not set 0.... = Push: Not set0.. = Reset: Not set0 = Fin: Not set Window size value: 8192 [Calculated window size: 8192] [Window size scaling factor: 1]

Wypełnij następujące informacje dotyczące wiadomości ACK.

Źródłowy adres IP:	
Adres docelowy IP:	
Numer portu źródłowego:	
Numer portu docelowego:	
Numer sekwencyjny:	
Numer potwierdzenia:	
Długość nagłówka:	
Rozmiar okna:	

Ile innych datagramów TCP zawiera bit SYN?

Po ustanowieniu sesji TCP, może wystąpić ruch FTP pomiędzy komputerem PC i serwerem FTP. Klient i serwer FTP komunikują się ze sobą nie wiedząc, że TCP kontroluje i zarządza nawiązaną przez nich sesją. Gdy serwer FTP wysyła odpowiedź: 220 do klienta FTP, to sesja TCP w kliencie FTP wysyła potwierdzenie do sesji TCP na serwerze. Tą sekwencję można przechwycić i obejrzeć w programie Wireshark.

10 1.523372000 192.168.1.17 198,246,112,54 TCP 54 49243 > ftp [ACK] Seq=1 Ack=28 Win= 12 4.585185000 192.168.1.17 198,246,112,54 FTP 70 Request: USER anonymous 13 4.675040000 198.246.112.54 192.168.1.17 126 Response: 331 Anonymous access allo FTP Þ ⊕ Frame 9: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0 ⊞ Ethernet II, Src: Netgear_99:c5:72 (30:46:9a:99:c5:72), Dst: HonHaiPr_be:15:63 (90:4c:e5:be:15:63) Internet Protocol version 4, src: 198.246.112.54 (198.246.112.54), Dst: 192.168.1.17 (192.168.1.17) I Transmission Control Protocol, Src Port: ftp (21), Dst Port: 49243 (49243), Seq: 1, Ack: 1, Len: 27 □ File Transfer Protocol (FTP) 220 Microsoft FTP Service\r\n Response code: Service ready for new user (220) Response arg: Microsoft FTP Service

Gdy połączenie FTP jest zakończone, to klient FTP wysyła komendę "quit". Serwer FTP potwierdza zakończenie połączenia FTP za pomocą odpowiedzi: 221 Goodbye. W tym momencie sesja TCP w serwerze FTP wysyła datagram TCP do klienta FTP, ogłaszający zakończenie sesji TCP. Sesja TCP na kliencie FTP potwierdza otrzymanie datagramu kończącego sesję i wysyła własny datagram TCP kończący sesję. Gdy źródło zakończenia sesji TCP, serwer FTP otrzyma podwójne zakończenie, datagram z ustawionym bitem ACK jest wysyłany aby potwierdzić zakończenie sesji TCP i sesja TCP jest zamknięta. Tą sekwencję można przechwycić i obejrzeć na diagramie.



Dzięki zastosowaniu filtru **ftp**, cała sekwencja ruchu FTP może być badana w programie Wireshark. Zwróć uwagę na kolejność występowania zdarzeń podczas tej sesji FTP. Do pobrania pliku Readme użyto nazwy użytkownika anonymous. Po zakończeniu transferu plików użytkownik zakończył sesję FTP.

Filter:	ftp			▼ Expre	ession C	lear Apply	Save	
No.	Time	Source	Destination	Protocol	Length	Info		
	1.314568000	198.246.112.54	192.168.1.17	FTP	81	Response:	220	Microsoft FTP Service
12	4.585185000	192.168.1.17	198.246.112.54	FTP	70	Request:	USER	anonymous
13	4.675040000	198.246.112.54	192.168.1.17	FTP	126	Response:	331	Anonymous access allowe
19	5.961514000	192.168.1.17	198.246.112.54	FTP	61	Request:	PASS	
20	6.048929000	198.246.112.54	192.168.1.17	FTP	85	Response:	230	Anonymous user logged i
25	8.855225000	192.168.1.17	198.246.112.54	FTP	80	Request:	PORT	192,168,1,17,192,92
26	8.945530000	198.246.112.54	192.168.1.17	FTP	84	Response:	200	PORT command successful
27	8.955549000	192.168.1.17	198.246.112.54	FTP	60	Request:	NLST	
29	9.053034000	198.246.112.54	192.168.1.17	FTP	109	Response:	150	Opening ASCII mode data
39	9.347432000	198.246.112.54	192.168.1.17	FTP	78	Response:	226	Transfer complete.
42	12.621720000	192.168.1.17	198.246.112.54	FTP	80	Request:	PORT	192,168,1,17,192,93
43	12.709658000	198.246.112.54	192.168.1.17	FTP	84	Response:	200	PORT command successful
44	12.722592000	192.168.1.17	198.246.112.54	FTP	67	Request:	RETR	Readme
45	12.811097000	198.246.112.54	192.168.1.17	FTP	118	Response:	150	Opening ASCII mode data
58	13.107294000	198.246.112.54	192.168.1.17	FTP	78	Response:	226	Transfer complete.
61	15.514815000	192.168.1.17	198.246.112.54	FTP	60	Request:	QUIT	
62	15.601920000	198.246.112.54	192.168.1.17	FTP	61	Response:	221	

Zastosuj ponownie filtr TCP w Wireshark aby zbadać zakończenie sesji TCP. Cztery pakiety są transmitowane dla zakończenia sesji TCP. Ponieważ połączenie TCP jest typu full-duplex, to każda strona musi samodzielnie dokonać zakończenia. Sprawdź adresy źródłowe i docelowe.

W tym przykładzie serwer FTP nie ma już danych do wysłania w strumieniu; serwer wysyła segment z ustawioną flagą FIN w ramce 63. PC wysyła ACK, aby potwierdzić otrzymanie FIN w celu zakończenia sesji z serwera do klienta w ramce 64.

W ramce 65 komputer PC wysyła FIN do serwera FTP, aby zakończyć sesję TCP. Serwer FTP odpowiada za pomocą ACK, aby potwierdzić FIN przychodzący z komputera PC w ramce 67. Teraz sesja TCP między serwerem FTP i PC jest zakończona.

	61 15.514815000 192.168.1.17	198.246.112.54	FTP	60 Request: QUIT	
	62 15.601920000 198.246.112.54	192.168.1.17	FTP	61 Response: 221	
	63 15.602245000 198.246.112.54	192.168.1.17	ТСР	54 ftp > 49243 [FIN,	ACK] Seq=365 Ack=
	64 15.602314000 192.168.1.17	198.246.112.54	тср	54 49243 > ftp [ACK]	Seq=101 Ack=366 V
	65 15.605832000 192.168.1.17	198.246.112.54	ТСР	54 49243 > ftp [FIN,	ACK] Seq=101 Ack=
	67 15.696497000 198.246.112.54	192.168.1.17	TCP	54 ftp > 49243 [ACK]	Seq=366 Ack=102 V
		III			Þ
Đ	Frame 63: 54 bytes on wire (43	2 bits), 54 bytes	captured (432	bits) on interface ()
			capearea (ise	- breby on meeriace	*
Œ	Ethernet II, Src: Netgear_99:c	5:72 (30:46:9a:99	:c5:72), Dst:	HonHaiPr_be:15:63 (90):4c:e5:be:15:63)
Œ	Ethernet II, Src: Netgear_99:c Internet Protocol Version 4, 5	5:72 (30:46:9a:99 rc: 198.246.112.5	:c5:72), Dst: 4 (198.246.112	HonHaiPr_be:15:63 (90 2.54), Dst: 192.168.1):4c:e5:be:15:63) .17 (192.168.1.17)
± ±	Ethernet II, Src: Netgear_99:c Internet Protocol Version 4, S Transmission Control Protocol,	5:72 (30:46:9a:99 rc: 198.246.112.5 Src Port: ftp (2	:c5:72), Dst: 4 (198.246.112 1), Dst Port:	HonHaiPr_be:15:63 (90 2.54), Dst: 192.168.1 49243 (49243), Seq:):4c:e5:be:15:63) .17 (192.168.1.17) 365, Ack: 101, Len

Wyzwanie

Ponieważ FTP nie są bezpiecznym protokołem, to wszystkie dane przesyłane za jego pomocą wysyłane są otwartym tekstem. Dotyczy to także ID użytkowników, haseł i zawartości plików tekstowych nieszyfrowanych. Analizując sesję FTP szybko odnajdziemy ID użytkownika, hasło a także hasła w plikach konfiguracyjnych.