IMMERSIVE LEARNING ENVIRONMENT

LAB: THREE-WAY HANDSHAKE LAB

INSTRUCTIONS





Client Machine

Remote Server

Three-way handshake is a method, which is used to establish a connection between local host/client and server. It is a three-step method in which both client and server exchange SYN and ACK packets before actual data communication begins.

STEP 1: Establish a TCP session between client and remote server

Open a web browser on your Linux virtual machine. Select the Firefox web browser icon from quick lunch bar on left side of your screen.



NOTE: Make sure your virtual machine is connected to an internet.

Now let's open Wireshark on our virtual machine from quick lunch bar or open terminal and type wireshark and hit enter on your terminal.



0	vmwaretool			
5				
	Wireshark			

OR

Vmwaretool	
Terminator	
🐵 🗇 🗇 /bin/bash	
[07/20/10] cood@\/Mt_f wireshark	/bin/bash 80x24
[0// SU/ Tal seen@nu: ~2 MTLESUGLK	

Select your virtual machine physical interface from Wireshark home screen. Most of the case it is usually first interface option on your Wireshark home screen.

2. click on blue shark icon to start capturing the packets Apply a display filter <ctrl-></ctrl->
Welcome to Wireshark Captureusing this filter: Letter a capture filter
1. Select Interface ens33 any A_A_A_ Loopback: lo A_A_ nflog
Page

After selecting the interface form the Wireshark home screen click on blue shark icon on your Wireshark home screen.

Now let's enter web address on our Firefox web browser web address field and open the website.



STEP 2: Analyze the captured packet on Wireshark

Once we have successfully opened the web address on our Firefox web browser. Let get back to our Wireshark active packet capture windows and stop capturing the packets.

😣 🔿 🗊 *ens33							
Stop capturing packets	(< > € ⊨ →						
Apply a display filter <ctrl-></ctrl->							
1 click on stop capturing packets No.	Source	Destination	Protocol	Length Info			
1 2019-07-30 13:26:18.7296375	192.168.198.1	192.168.198.255	BROWSER	243 Host Announcement DESKTOP-8IDTRHD,			
2 2019-07-30 13:26:28.1626462	192.168.198.135	192.168.198.2	DNS	84 Standard query 0x78e6 A detectport			
3 2019-07-30 13:26:28.1628064	192.168.198.135	192.168.198.2	DNS	84 Standard query 0x8aff AAAA detectp			
4 2019-07-30 13:26:28.1792406	Vmware_fd:d5:5c	Broadcast	ARP	60 Who has 192.168.198.135? Tell 192.			
5 2019-07-30 13:26:28.1792699	Vmware_ff:7c:01	Vmware_fd:d5:5c	ARP	42 192.168.198.135 is at 00:0c:29:ff:			
6 2019-07-30 13:26:28.1796755	192.168.198.2	192.168.198.135	DNS	242 Standard query response 0x78e6 A d			
7 2010-07-30 13:26:28 1707051	102 168 108 2	102 168 108 135	DNS	266 Standard query response Av8aff AAA			

Click on stop capturing packets icon on top of your Wireshark menu icon next to blue icon.

let's analyze the captured packets on Wireshark. You will notice a lot of packets.

Enter TCP on filter column below the menu icon and hit enter. The packets will be selected based on the filter option which makes easy to search capture packets. Lets focused on TCP protocol packets and observe the TCP packets follow. You will notice TCP SYN , SYN ACK and ACK packets stream with identical source and destination IP address associated with specific Destination port request. In our case it web request port 80. Find the detail on screenshot below.

	, t	🗘 🗘 1. Enter top)						X 🗆
	No.	Time		Source	Destination	Protoc *	Length Info		
		7053 2019-07-30	13:26:45.3253901	192.168.198.135	172.217.6.3	OCSP	495 Request		
		7063 2019-07-30	13:26:45.5045125	172.217.6.3	192.168.198.135	OCSP	755 Response 3	. TCP T	'hree-Way Hankshake Packets
		7199 2019-07-30	13:26:45.8190319	192.168.198.135	172.217.6.3	OCSP	495 Request	л	
	1	7212 2019-07-30	13:26:45.9969881	1/2.21/.6.3	192.168.198.135	OCSP	755 Response	<u> </u>	
2. Observer the TCP packets		0 2019-07-30	13:20:20.104//9/	140 165 180 17	149.103.100.17	TCP	74 37360 → 60 60 80 ÷ 27586	LOTH	Seq=721173754 WIII-29200 Lell=0 M55=1400 SACK_PERM-
		10 2019-07-30	13:20:20.1930030	149.105.100.17	149 165 180 17	TCP	54 37586 → 80	Lack]	Ack] Seq=1050556004 Ack=721175755 Win=04240 Len=0
		12 2019-07-36	13:26:28 1942933	149 165 180 17	192 168 198 135	TCP	60 80 → 37586	[ACK]	Seg=1650358605 Ack=721174049 Win=64240 Len=0
		14 2019-07-30	13:26:28.2028198	192.168.198.135	149.165.180.17	TCP	54 37586 → 80	[ACK]	Seg=721174049 Ack=1650358989 Win=30016 Len=0
		19 2019-07-30	13:26:31.1932455	192.168.198.135	128.230.247.70	TCP	74 33306 → 80	[SYN]	Seg=771871999 Win=29200 Len=0 MSS=1460 SACK_PERM=
		20 2019-07-30	13:26:31.2231481	128.230.247.70	192.168.198.135	TCP	60 80 → 33306	[SYN,	ACK] Seq=1289766779 Ack=771872000 Win=64240 Len=0
		21 2019-07-30	13:26:31.2232265	192.168.198.135	128.230.247.70	TCP	54 33306 → 80	[ACK]	Seq=771872000 Ack=1289766780 Win=29200 Len=0
		23 2019-07-30	13:26:31 7959602	128 230 247 70	192 168 198 135	TCP	60 80 → 33306	FACK1	Seg=1289766780 Ack=771872332 Win=64240 Len=0
	▼ T	ransmission Cont	trol Protocol, Src	Port: 37586, Dst Port	: 80, Seq: 72117375	54, Len: 0			
		Source Port: 3	7586						
		Destination Por	rt: 80						
		[Stream index:	0]						
		EICP Segment Le	en: 0]						
		Acknowledgment	number: 0						
		Header Length:	40 bytes						
		Flags: 0x002 ()	SYN)						
		Window size va	lue: 29200						

WHAT TO SUBMIT

Submit you work with detailed screenshots.



This document is licensed with a <u>Creative Commons Attribution 4.0 International License</u> ©2017 <u>Catalyzing Computing and Cybersecurity in</u>