Peyrin & Ryan Summer 2020

# CS 161 Computer Security

# Network Security I

## Question 1 Introduction to Networking

- (a) **TCP and UDP** The transmission control protocol (TCP) and user datagram protocol (UDP) are two of the primary protocols of the Internet protocol suite.
  - i. How do TCP and UDP relate to IP (Internet protocol)? Which of these protocols are encapsulated within (or layered atop) one another? Could all three be used simultaneously?

ii. What are the differences between TCP and UDP? Which is considered "best effort"? What does that mean?

()

### Question 2 Attack on TCP

()

Suppose that a client connects to a server, and then performs the following TCP hand-shake and initial data transfer:

Client (port $P_C$ )	Server	(port $P_S$ )
	SYN	1. Client sends initial SYN with sequence number $A$ (usually random).
	SYN-ACK	2. Server sends SYN-ACK with sequence number $B$ (also usually random) and ACK $A+1.$
	ACK	<ol> <li>Client sends ACK with sequence number A+1 and ACK B+1.</li> <li>Client sends DATA A of length L<sub>A</sub> with sequence number A+1 and ACK B+1.</li> </ol>
	DATA A	
<	DATA B	5. Server sends DATA B of length $L_B$ with sequence number $B + 1$ and ACK $A + 1 + L_A$ .
	DATA C	6. Client sends DATA C of length $L_C$ with sequence number $A + 1 + L_A$ and ACK $B + 1 + L_B$ .
<b>.</b>		7. Data exchange continues until both sides are done send- ing data.

(a) Assume that the next transmission in this connection will be DATA D from the server to the client. What will this packet look like?

Sequence number:		ACK:	
Source port:	$P_S$	Destination port:	$P_C$
Length:	$L_D$	Flags:	ACK

- (b) You should be familiar with the concept and capabilities of a *man-in-the-middle* as an attacker who **can observe** and **can modify** traffic. There are two other types of relevant attackers in this scenario:
  - 1. On-path attacker: can observe traffic but cannot modify it.
  - 2. Off-path attacker: cannot observe traffic and cannot modify it.

Carol is an *on-path* attacker. Can Carol do anything malicious to the connection? If so, what can she do?

(c) David is an *off-path* attacker. Can David do anything malicious to the connection? If so, what can he do?

(d) The client starts getting responses from the server that don't make any sense. Inferring that David is attempting to hijack the connection, the client then immediately sends the server a **RST** packet, which terminates the ongoing connection. David wants to impersonate the client by establishing a new connection. How would he go about doing this?

#### Question 3 TLS protocol details

Depicted below is a typical instance of a TLS handshake.

Client	Serv	er
	1. ClientHello	1. Client sends 256-bit random number $R_b$ and supported ciphers
<del>.</del>		2. Server sends 256-bit random number $R_s$ and chosen cipher
	3. Certificate	3. Server sends certificate
	4. ServerKeyExchange	4. DH: Server sends $\{g, p, g^a \mod p\}_{K_{\text{server}}^{-1}}$
	5. ServerHelloDone	5. Server signals end of handshake
	6. ClientKeyExchange	6. DH: Client sends $g^b \mod p$ RSA: Client sends $\{PS\}_{K_{server}}$
7. Chai	7. ChangeCipherSpec, Finished	Client and server derive cipher keys $C_b, C_s$ and integrity keys $I_b, I_s$ from $R_b, R_s, PS$
	, moneu ,	7. Client sends MAC(dialog, $I_b$ )
8	3. ChangeCipherSpec, Finished	8. Server sends MAC(dialog, $I_s$ )
	9. Application Data	
*	10. Application Data	9. Client data takes the form $\{M_1, \text{MAC}(M_1, I_b)\}_{C_b}$ 10. Server data takes the form $\{M_2, \text{MAC}(M_2, I_s)\}_{C_s}$



- (a) What is the purpose of the *client random* and *server random* fields?
- (b) ClientHello and ServerHello are not encrypted or authenticated. Explain why a man-in-the-middle cannot exploit this. (Consider both the Diffie-Hellman and RSA case.)

(c) Note that in the TLS protocol presented above, there are two cipher keys  $C_b$  and  $C_s$ . One key is used only by the client, and the other is used only by the server. Likewise, there are two integrity keys  $I_b$  and  $I_s$ . Alice proposes that both the server and the client should simply share one cipher key C and one integrity key I. Why might this be a bad idea?

(d) The protocol given above is a simplified form of what actually happens. After step 8 (ChangeCipherSpec), the protocol as described above is still vulnerable. What is the vulnerability and how could you fix this?