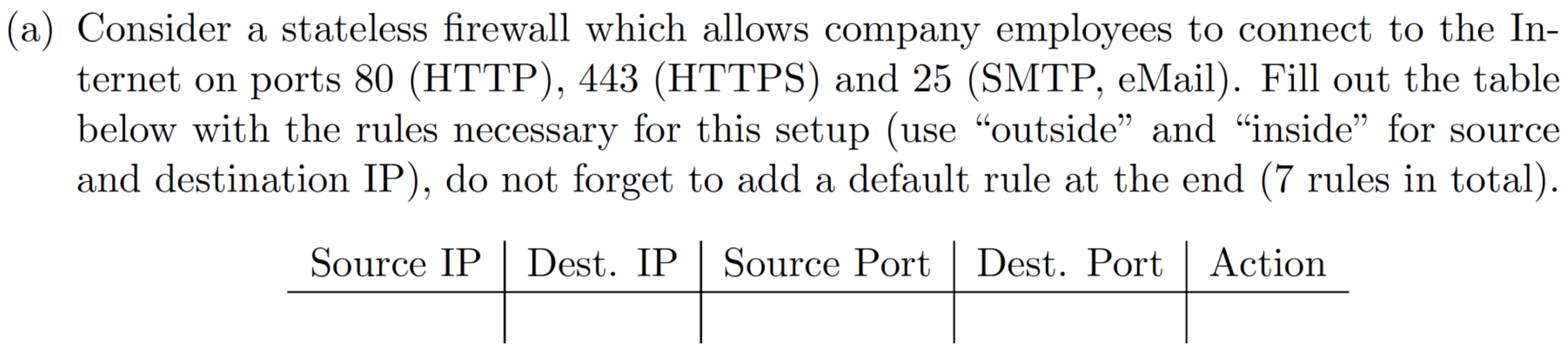
Tutorial 10 Solutions

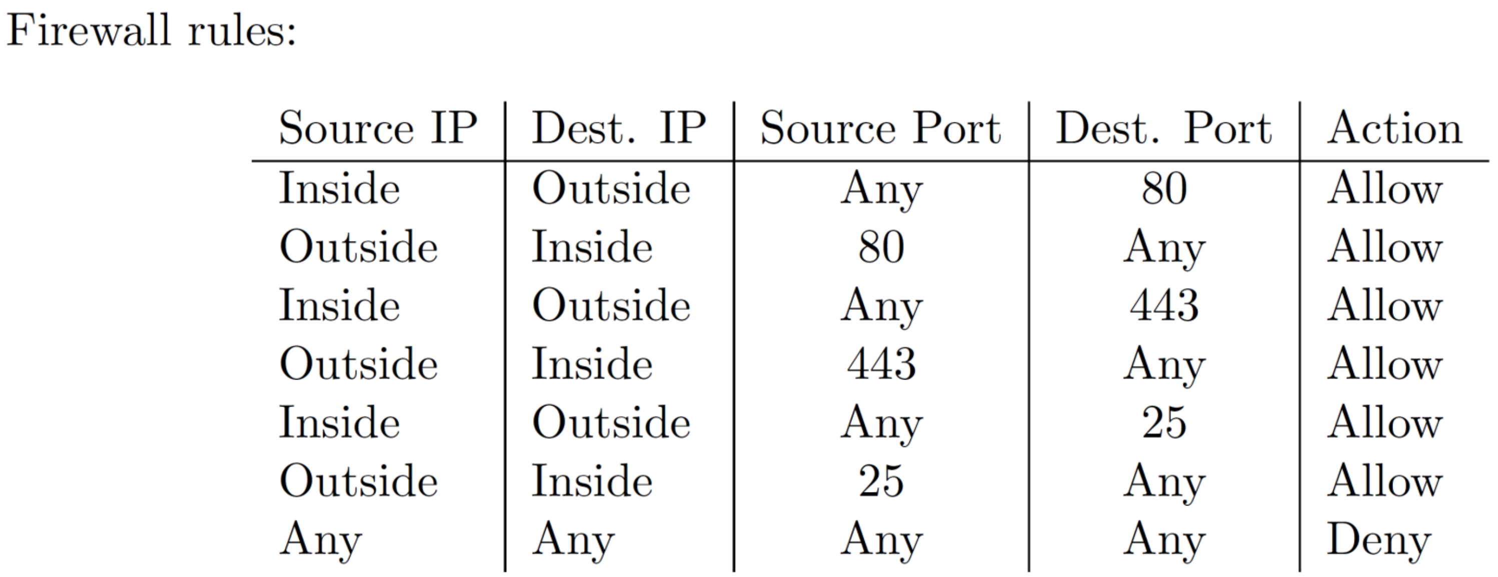
CS5285

Question 1



(a) Consider a stateless firewall which allows company employees to connect to the Internet on ports 80 (HTTP), 443 (HTTPS) and 25 (SMTP, eMail). Fill out the table below with the rules necessary for this setup (use “outside” and “inside” for source and destination IP), do not forget to add a default rule at the end (7 rules in total).

Source IP	Dest. IP	Source Port	Dest. Port	Action



Firewall rules:

Source IP	Dest. IP	Source Port	Dest. Port	Action
Inside	Outside	Any	80	Allow
Outside	Inside	80	Any	Allow
Inside	Outside	Any	443	Allow
Outside	Inside	443	Any	Allow
Inside	Outside	Any	25	Allow
Outside	Inside	25	Any	Allow
Any	Any	Any	Any	Deny

Question 2

a) In each of the following scenarios, identify the type of malicious program that a host is being attacked by.

(1) A program replicates itself in a very fast pace and severely slows down the host.

Bacteria

(2) A program starts erasing data in a hard drive when the date becomes April 1st while the program does not infect any executable files.

Logic bomb

(3) A program emails a copy of itself to a subset of email addresses obtained from an address book stored in the host and starts erasing data in a hard drive when the date becomes April 1st.

Worm

(4) A program monitors all HTTP GET messages sent out from the host and emails a copy of the messages to trapdoor@whatever.email.net.

Trojan Horse

Question 3

a) What is the difference between a virus and a worm?

- A worm can spread over a network with goal to infect as may hosts as possible, a virus is looking to infect files (it also wants to move around) but is looking for a suitable ``carrier'' to travel from one host to another (e.g., an email message, a USB stick, etc.).

Question 3

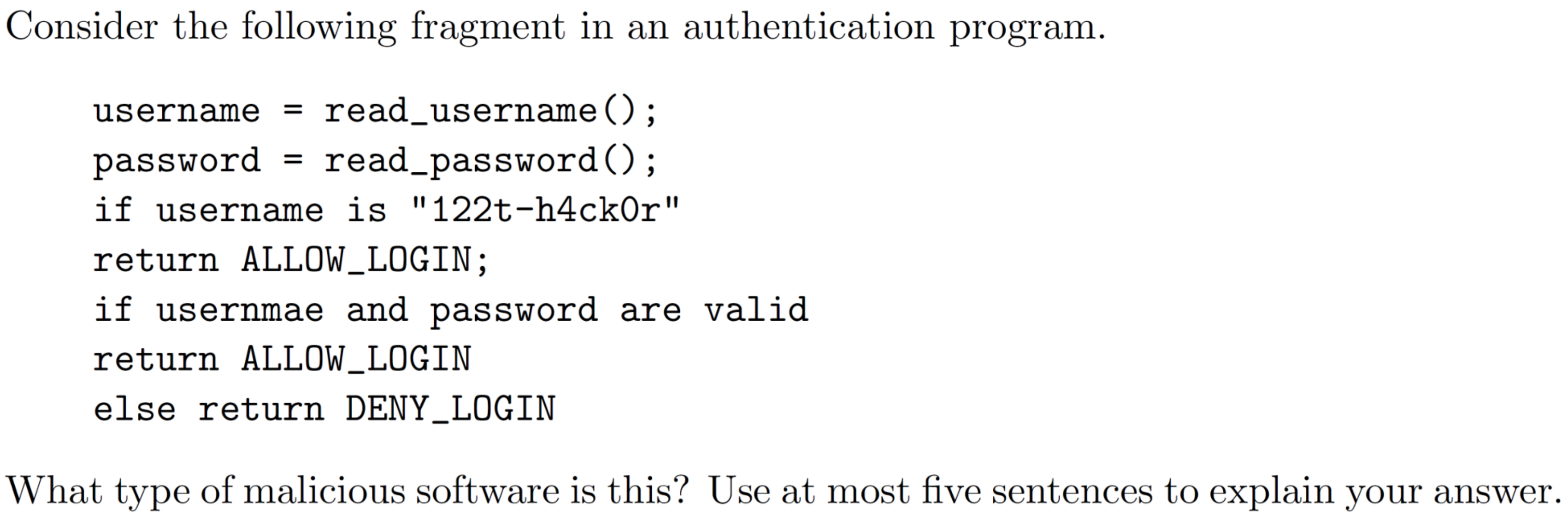
b) Which kind of firewall could stop a virus from spreading?

- Only an application proxy can detect a virus, because only an application proxy looks at the application data of a communication.

c) What are common defenses against malware?

- There are several good practices:
  - use a firewall
  - keep your operating system up-to-date with patches
  - use an antivirus software
  - treat all attachments in email messages from unknown people as suspicious

Question 3 (d)



Consider the following fragment in an authentication program.

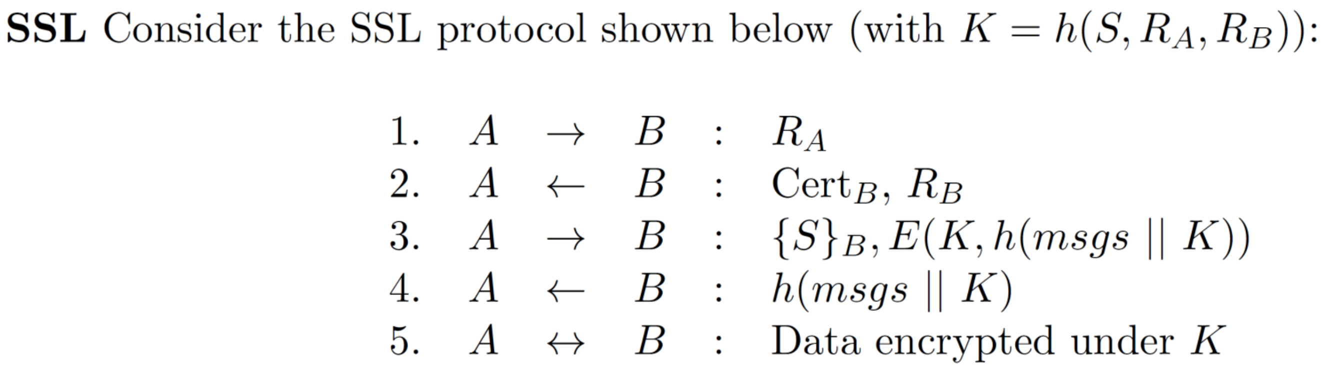
```
username = read_username();
password = read_password();
if username is "122t-h4ck0r"
return ALLOW_LOGIN;
if usernmae and password are valid
return ALLOW_LOGIN
else return DENY_LOGIN
```

What type of malicious software is this? Use at most five sentences to explain your answer.

(1) Trapdoor / Backdoor. If the login username is "122t-h4ck0r", the computer system allows the user to log onto the system regardless what password the user enters, while all other users have to enter their passwords correctly.

(2) So, the system does require valid passwords for all users except the user "133t-h4ck0r". Hence it is reasonable to conclude that this username is a secret entry point for the ones who know this secret username to access the system.

Question 4 (a)

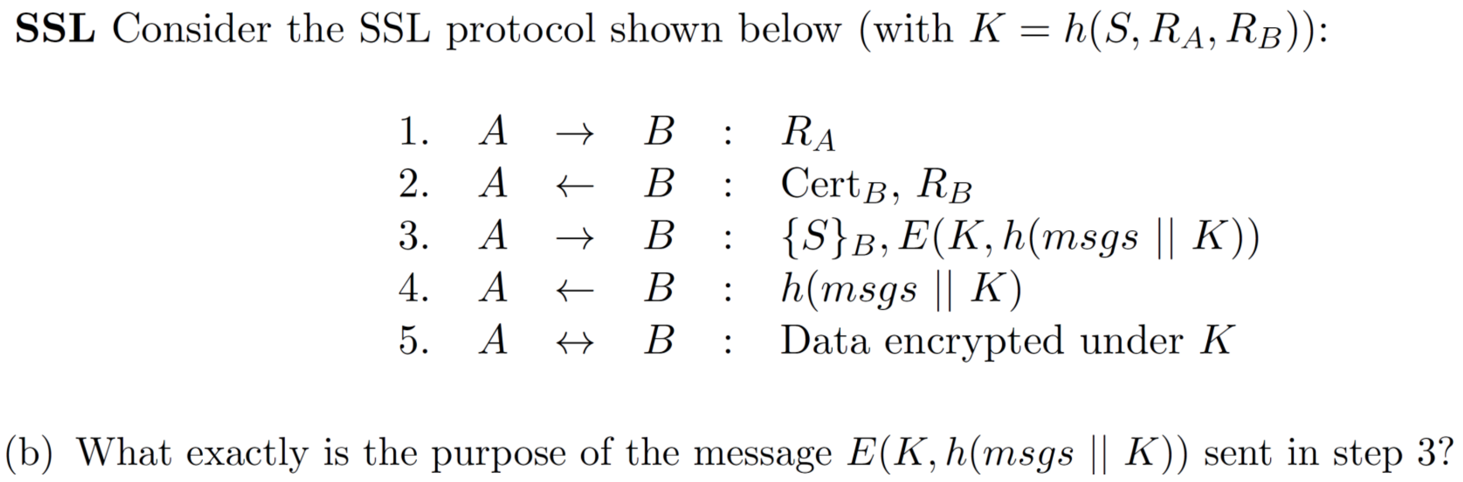


SSL Consider the SSL protocol shown below (with $K = h(S, R_A, R_B)$):

$$\begin{array}{llcll}
1. & A & \rightarrow & B & : & R_A \\
2. & A & \leftarrow & B & : & \mathrm{Cert}_B,\ R_B \\
3. & A & \rightarrow & B & : & \{S\}_B, E(K, h(msgs \mid\mid K)) \\
4. & A & \leftarrow & B & : & h(msgs \mid\mid K) \\
5. & A & \leftrightarrow & B & : & \text{Data encrypted under } K
\end{array}$$

- What would TLS look like if it was based on symmetric cryptography only? Would it be practical?
- No $\{S\}_B$ and key based on long term symmetric key K_{AB} .
- K= $h(K_{AB},R_A,R_B)$
- This would require every client to somehow share a symmetric key with every server on the Internet….not practical

Question 4 (b)



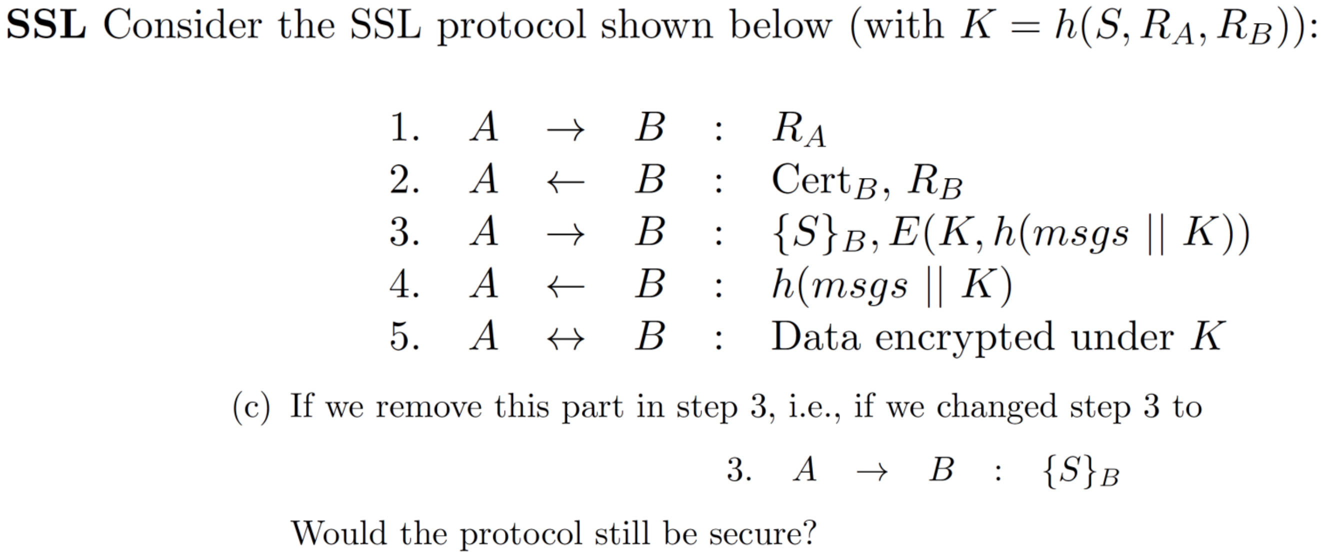
SSL Consider the SSL protocol shown below (with $K = h(S, R_A, R_B)$):

$$\begin{array}{lllll}
1. & A & \rightarrow & B & : & R_A \\
2. & A & \leftarrow & B & : & \text{Cert}_B,\ R_B \\
3. & A & \rightarrow & B & : & \{S\}_B, E(K, h(msgs \mid\mid K)) \\
4. & A & \leftarrow & B & : & h(msgs \mid\mid K) \\
5. & A & \leftrightarrow & B & : & \text{Data encrypted under } K
\end{array}$$

(b) What exactly is the purpose of the message $E(K, h(msgs \mid\mid K))$ sent in step 3?

- The message E(K, h(msgs || K)) sent in step 3 confirms that A actually knows K (explicit key authentication).
- Only Bob can decrypt {S}_B and generate the correct h(msgs || K), thus Alice can still be certain she is talking to Bob.

Question 4 (c)



SSL Consider the SSL protocol shown below (with $K = h(S, R_A, R_B)$):

$$\begin{array}{lllll}
1. & A & \rightarrow & B & : & R_A \\
2. & A & \leftarrow & B & : & \text{Cert}_B,\ R_B \\
3. & A & \rightarrow & B & : & \{S\}_B, E(K, h(msgs \parallel K)) \\
4. & A & \leftarrow & B & : & h(msgs \parallel K) \\
5. & A & \leftrightarrow & B & : & \text{Data encrypted under } K
\end{array}$$

(c) If we remove this part in step 3, i.e., if we changed step 3 to

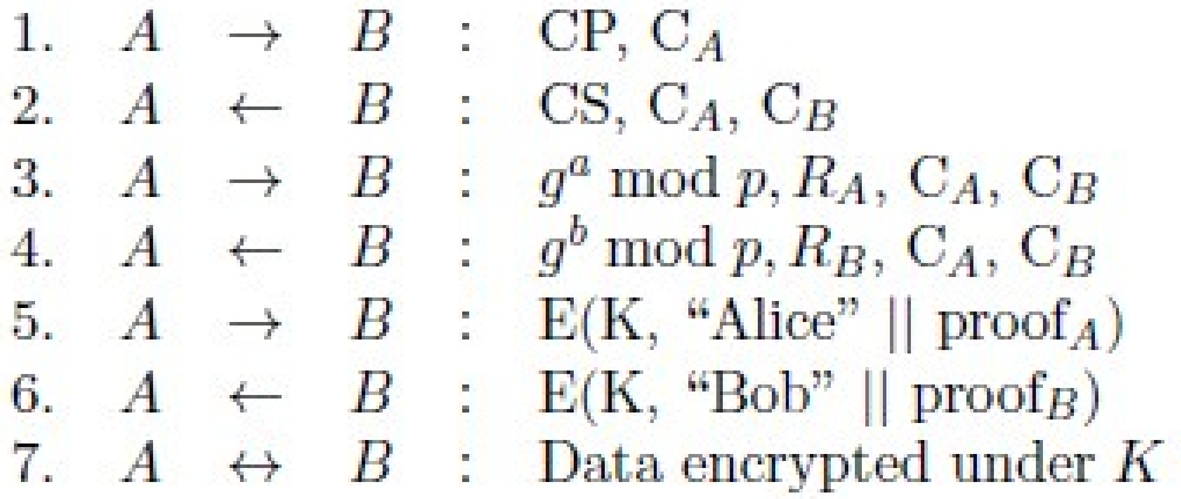
$$3. \quad A \quad \rightarrow \quad B \quad : \quad \{S\}_B$$

Would the protocol still be secure?

- Depends what we mean by ‘secure’…the protocol would still be secure from authentication/key establishment viewpoint.
- It could be more vulnerable to denial-of-service attacks. If this message is removed, an attacker can simply send a random number to Bob in step 3 and then abandoning the connection, forcing Bob to keep it open until it times out, wasting resources on Bob's side. If the attacker repeats this many times from different sources until a limit is reached, Bob will stop accepting new connections and the DoS attack is successful.

Question 5

- Imagine you have a key exchange protocol similar to main mode in IKE Phase 1, but adding an additional piece of data ("cookies", C_A and C_B) to the message flow:



1.	A	$\rightarrow$	B	:	CP, C_A
2.	A	$\leftarrow$	B	:	CS, C_A, C_B
3.	A	$\rightarrow$	B	:	$g^a \bmod p, R_A$, C_A, C_B
4.	A	$\leftarrow$	B	:	$g^b \bmod p, R_B$, C_A, C_B
5.	A	$\rightarrow$	B	:	E(K, "Alice" || proof_A)
6.	A	$\leftarrow$	B	:	E(K, "Bob" || proof_B)
7.	A	$\leftrightarrow$	B	:	Data encrypted under K

Question 5

The cookies are in the form $C_x = h(K_x, \mathrm{IP}_{peer}, \mathrm{timestamp})$ where K_x is a secret key only known to the party creating the cookie and IP_{peer} is the IP address of the peer (i.e., Alice would put Bob’s IP and vice versa).

(a) What are the reasons for including such cookies in the exchange?

Question 5

- **Solution to (a)**

These cookies can help identifying spoofed packets (i.e., packets containing a fake sender address). This ties the messages to a specific IP address of A and B. If an attacker wants to initiate communication with the other party he needs a cookie from the other party, valid for the IP address he is using (so a cookie is something that says you can talk to me, but only from this IP). It stops an attacker from making many fake IP addresses and initiating session with A/B.

Question 5

- (b) The function of these cookies have be effective before the exchange reaches step 5, otherwise B could be in trouble. Can you explain why?

Question 5

- **Solution**: As indicated in (a), the cookies help prevent spoofed exchanges, that means if the cookies are correct, B assumes that A is a legitimate user and that the exchange should go ahead. B has to be sure that this is the case, because steps 5 and 6 involve actually calculating K which requires an exponentiation ($g^{ab} \bmod p$). This is a quite expensive operation. If an attacker could generate thousands of spoofed exchanges and B would need to do an exponentiation for each exchange, the attacker could quickly exhaust the computational resources of B, effectively launching a denial-of-service attack.