

Problem 1 – 30 points

Problem 2 – 20 points

Problem 3 – 20 points

Problem 4 – 30 points

Problem 1 (Short Answer) (30 points)

Give brief answers (one or two sentences) to each of the following.

- a. What is the principle of least privilege? Why is it important?
- b. Is a TCP connection secure against eavesdropping? Why or why not?
- c. You have a copy of Anthony Joseph's certificate chain: his certificate is signed by the EECS department; the EECS department's certificate is signed by UC Berkeley; UC Berkeley's certificate is signed by Verisign. Whose public keys do you need to know in advance in order to obtain the correct public key for Anthony?

Problem 2 (Packet Filters) (20 points)

This problem explores some uses and limitations of (stateless) packet filtering firewalls.

You may use the syntax for packet filtering rules from Sept. 12 lecture. A ruleset is a list of rules, and the first matching rule determines the action taken. A rule is an action followed by a specification of which packets match: e.g., drop tcp 1.2.3.4:* -> *:25.

a. We have an internal web server, used only for testing purposes, at IP address 5.6.7.8 on our internal corporate network. The packet filter is situated at a choke point between our internal network and the rest of the Internet. Can such a packet filter block all attempts by outside hosts to initiate a direct TCP connection to this internal web server? If yes, show a packet filtering ruleset that provides this functionality; if no, explain why a (stateless) packet filter cannot do it.

b. Can a packet filter block all incoming email containing the phrase "Make money fast"? If yes, show a packet filtering rule set that provides this functionality; if no, explain why a (stateless) packet filter can not do it.

Problem 3 (Source Code Analysis) (20 points)

```
/* Escapes all new lines in the input string, replacing them with '\n'. */  
/* Requires: p != NULL; p is a valid '\0'-terminated string */
```

```
void escape(char *p)  
{  
    while (*p)  
    {
```

```

if (*p == '\n')
{
    memcpy(p+2,p+1, strlen(p));
    *p++ = '\\'; *p++ = '\n';
}
else p++;
}
}

```

You may assume that `escape()`'s argument is always non-null and points to a `'\0'` terminated string. What's wrong with this code from a security point of view?

Problem 4 (Crypto) (30 points)

Alice wants to send a cellphone text message to Bob securely, over an insecure communication network. Alice's cellphone has a RSA public key K_a and matching private key V_a ; likewise, Bob's cellphone has K_b and V_b . Let's design a cryptographic protocol for doing this, assuming both know each other's public keys.

Here is what Alice's cellphone will do to send the text message m :

1. Alice's phone randomly picks a new AES session key k and computes $c = \text{RSA-Encrypt}(K_b, k)$, $c' = \text{AES-CBC-Encrypt}(k, m)$ and $t = \text{RSA-Sign}(V_a, (c, c'))$.
2. Alice's phone sends (c, c', t) to Bob's phone.

Here is what Bob's cell phone will do upon receiving (c, c', t) :

1. Bob's phone checks that t is a valid RSA signature on (c, c') under public key K_a . If not, abort.
2. Bob's phone computes $k' = \text{RSA-Decrypt}(V_b, C)$ and $m' = \text{AES-CBC-Decrypt}(k', c')$.
3. Bob's phone informs Bob that Alice sent message m' .
 - a. Does this protocol ensure the confidentiality of Alice's messages? Why or why not?
 - b. Does the protocol ensure authentication and data integrity for every text message Bob receives? Why or why not?
 - c. Suppose that Bob is Alice's stockbroker. Bob hooks up the output of this protocol to an automatic stock-trading service, so if Alice sends a text message "Sell 100 shares MSFT" using the above protocol, then this trade will be immediately and automatically executed from Alice's account. Suggest one reason why this might be a bad idea from a security point of view.