**CS155: Computer Security** **Spring 2002**

# Final Exam

Name: ______________________________

Student ID #: ________________________

**Instructions:**

- The exam is two and a half hours long.
- The exam is open book and open notes.
- You must answer all 6 questions.
- Don't forget to sign the honor code.
- Good luck!

Do not write in the box below

| 1 | 2 | 3 | 4 | 5 | 6 | Total |
|---|---|---|---|---|---|---|
| | | | | | | |

## The Stanford Honor Code

In accordance with both the letter and spirit of the **Honor Code**, I did not cheat on this exam.

Signature: ____________________________________

**Problem 1.** Problems from all over.

**a.** Suppose there are two versions of a network daemon, one that stores its input buffer on the program run-time stack and one that stores its buffer on the heap. Which implementation is more likely to be easy to exploit using a buffer overflow attack? Briefly justify your answer.

**b.** Assume a client machine authenticates a user to a remote server over a cleartext channel. The following table refers to four authentication methods. Write 'yes' or 'no' in each of the following table cells.

| | Vulnerable to network sniffing | Network data vulnerable to dictionary attack | Client can change his secret in case of a compromise |
|---|---|---|---|
| Plaintext passwords | | | |
| Plaintext biometrics | | XXXXXXXXXXXXXX XXXXXXXXXXXXXX | |
| Challenge response with user secret on smartcard | | | yes |
| Challenge response using human passwords | | | yes |

**c.** Software fault isolation enables several applications to run in a single process without interfering with one another. Briefly explain the potential benefit of software fault isolation over running these applications in separate processes.

**d.** Briefly describe one way in which the lack of security in the DNS protocol can lead to a serious security vulnerability.

**e.** Some host based intrusion detection systems run in user space as a user process (such as Tripwire) while other host based intrusion detection systems run in kernel space (such as LIDS).
Which approach is harder to circumvent? Why?
Which approach is harder to upgrade and maintain? Why?

**Problem 2.** Principle of least privilege.

In this question we will design a mail server daemon broken into components based on the principle of least privilege. The mail server should support the following functionality:

1. The mail server daemon should accept mail from remote hosts and from users on the local host. To receive mail from a remote host the mail server listens on network port 25. To receive mail from a local user the mail server listens on a local (unix domain) socket.
2. All mail given to the mail server is queued in a mail queue.
3. Mail that should be delivered to a remote host is sent by connecting to port 25 on the remote host. Mail that should be delivered to a local user is appended to that user's mailbox file.

The mail server will consist of a number of processes (components), where each process is given exactly those privileges needed to perform its duties.

Five components are sufficient for implementing proper separation of duties on the mail server. List these components and for each component provide the following information:

1. what does the component do,
2. what are the privileges needed for the component to carry out its task,
3. what damage can an attacker do if the component is compromised.

Please also draw a box diagram of the five components and the basic message flow between them.

**Problem 3.** Kerberos.

- **a.** Draw the message flow in the Kerberos authentication protocol when a user wants to access some network service. Assume the user already has a TGT. Write the main information exchanged in each flow.
- **b.** Suppose the secret key stored on a network server is exposed. Explain how a hacker can eavesdrop on all past and future communication to that network server by sniffing (and possibly recording) network traffic.
- **c.** Suppose the secret key stored on the KDC is exposed. Explain how a hacker can eavesdrop on all past and future communication in the network by sniffing (and possibly recording) network traffic.
- **d.** If the KDC crashes then users can no longer authenticate themselves to network services. Describe in detail how you would design a Kerberos domain that does not have a single point of failure. What are the security implications of your design (is your network more vulnerable to attack)?

**Problem 4.** Firewalls

In this question we explore some applications and limitations of a stateless packet filtering firewall. For each question, briefly explain how the firewall should be configured to defend against the attack, or why the firewall cannot defend against the attack.

1. Can the firewall prevent a SYN flood attack from the external network?
2. Can the firewall prevent a Smurf attack from the external network?
3. Can the firewall prevent external users from exploiting a security bug in a CGI script on an internal web server (the web server is serving requests from the Internet)?
4. Can the firewall prevent an online password dictionary attack from the external network on the telnet port of an internal machine?
5. Can the firewall prevent a user on the external network from opening a window on an X server in the internal network? Recall that by default an X server listens for connections on port 6000.
6. Can the firewall block a virus embedded in an incoming encrypted email?
7. Can the firewall be used to block users on the internal network from browsing a specific external IP address?

**Problem 5.** Attacks on bad code

**a.** Consider the following code snippet:

```
filename = mktemp();        // Make a unique non-existent temporary filename.
sleep(10);                  // Sleep for 10 seconds.
fp = fopen(filename, ``w'');
fprintf(fp, ``Hello world'');     // Write to file.
close(fp);
```

Suppose this code is part of some setuid root program. Give an example of how this code can be exploited by an unprivileged user to do some unauthorized operation. Assume the code is running on Linux and that the attacker knows the value returned by `mktemp`.

**b.** Suppose the `sleep(10)` is removed from the code above. Can you still mount the attack from part (a)? Explain.

**c.** How would you change the `mktemp` library function to prevent the attack from part (a)?

**d.** Suppose an administrator installs a Linux patch that marks the stack segment as non-execute. Give an example of how a buffer overflow in a local buffer on the stack can still result in a root shell. Assume the code is running in a web server running as root. The attacker is attempting to get a root shell from the network.

**e.** Suggest one practical way in which your attack from part (d) can be prevented (without changing the server source code).

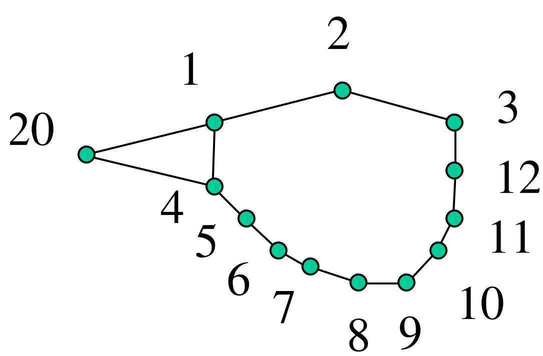
**Problem 6.** BGP and S-BGP

The BGP protocol is used to select routes on the Internet. For this question, the Internet is a graph whose nodes are administrative systems (ASes), each with a unique number. Each AS is responsible for some set of IP addresses. To advertise a destination IP address that it owns, a node sends each of its neighbors an UPDATE message containing the IP address and its AS number. The neighbors may then send this information on to their own neighbors in another UPDATE message containing their own AS number and the number of the AS that started the message. Similarly, their neighbors may propagate the information, and so on. Eventually each AS should have a table of routes leading to each other AS in the network, together with a list of IP addresses owned by each AS in the table. If an AS receives two or more advertised routes to the same destination, it must choose at most one to advertise. For example, AS 5 may receive two advertisements for reaching IP address 42.42.42.42 at AS 10, such as route $6, 7, 3, 10$ from AS 6 and route $2, 10$ from AS 2. It can advertise (i.e. send to its neighbors) either $5, 6, 7, 3, 10$ or $5, 2, 10$ but not both.

ASes can also withdraw previously-advertised routes, but we will not consider that here. All these questions are about what happens when nodes start with empty routing tables and only receive route advertisements.

BGP has no security—there is nothing to stop a malicious intruder from altering or removing the messages sent between ASes, or from inserting its own bogus messages. There is also nothing to stop a malicious AS from lying about the routes available or advertising IP addresses that it doesn't own.

**S-BGP:** S-BGP is a security addition to BGP that addresses many of these problems. It includes a public key infrastructure that allows any AS to identify another's public key and to check that a destination AS truly owns the IP addresses it is advertising. Another important feature is the *route attestation:* when AS $a$ sends to AS $b$ an advertisement containing route $P$, it signs $P$ together with the statement "$b$ is allowed to advertise this route." This implies that $b$ really is a neighbor of $a$. All relevant route attestations are sent along with a route advertisement. Hence a recipient of an advertisement for route $\mathrm{AS}_n, \ldots, \mathrm{AS}_1$ can check that for all $i$ between 1 and $n-1$, there is a valid route attestation signed by $\mathrm{AS}_i$ that includes route $\mathrm{AS}_i, \ldots, \mathrm{AS}_1$ and "$\mathrm{AS}_{i+1}$ is allowed to advertise this route."



**Questions.** Consider the network shown in the Figure. We will examine the behavior of AS 1 and AS 4 sending AS 20 a route advertisement for reaching AS 3. Suppose that all (non-compromised) ASes prefer to choose the shortest route. In a correct execution of the protocol:

- AS 1 advertises route $1, 2, 3$.
- AS 4 advertises route $4, 1, 2, 3$.
- AS 20 chooses route $1, 2, 3$ because it's shorter.

**a.** (Attack A) Suppose AS 1 is compromised from the beginning. It advertises to AS 20 and AS 4 the route $1, 4, 5, 6, \ldots, 12, 3$ instead of $1, 2, 3$. AS 4 and all the other nodes remain secure. What route will AS 4 advertise? What route will AS 20 choose? Is this likely to reduce the speed of traffic traveling from 20 to 3?

**b.** (Attack B) Suppose AS 1 remains secure (and advertises route $1, 2, 3$) but AS 4 is compromised from the beginning. It advertises route $4, 3$, though it will actually route any packets it receives to 5. All the other nodes remain secure. Now what route will AS 20 choose? Compared to attack A, is the network transit time of packets from 20 to 3 probably much greater, much less, or about the same? Justify your answer.

**c.** If everyone in the network were running S-BGP, would that prevent attack A? Justify your answer.

**d.** If everyone in the network were running S-BGP, would that prevent attack B? Justify your answer.

**e.** The BGP protocol allows each AS a great deal of freedom in selecting what route to advertise (and use). Is it possible for a signature-based extension of BGP to ensure that each AS implements its own route selection policy correctly? Explain why or why not.