CS 161 Computer Security
Fall 2005 Joseph/Tygar/Vazirani/Wagner **Final Soln**

Problem 1. [Defaults] (9 points)

(a) Which is generally safer (from a security point of view), a firewall with a "default deny" policy or a firewall with a "default allow" policy? Why?

Default deny. Inadvertently omitting an item from the list causes a loss of service but does not compromise security.

(b) Many spam filters can be configured to use either a whitelist or a blacklist. Name one advantage of using a whitelist (instead of a blacklist) for your spam filter.

You won't receive spam from anyone who isn't on the whitelist (unless, of course, the spammer forges the From: address so it looks like it came from someone on your whitelist). These are the same advantages as "default deny."

(c) Name one disadvantage of using a whitelist (compared to a blacklist) for your spam filter.

Legitimate email from people who you haven't entered onto your whitelist won't get through, so you might not see some email you'd like to see.

Problem 2. [Authentication] (8 points)

Describe two fundamentally different conceptual approaches that can be used for user authentication. Be concise: One sentence should suffice.

A few sample solutions (listing any two is acceptable):

- *Passwords. The user types in her secret password to prove her identity.*
- *Cryptographic authentication protocols. The user has a secret key and authenticates herself using a crypto protocol (e.g., SSL).*
- *Biometrics. The user displays her fingerprint, hand geometry, iris scan, etc., so that the computer can verify that it is really her.*
- *Hardware tokens. The user owns a hardware device which displays a new one-time password every minute, and the user types in the current one-time password so that the server can verify its correctness.*

Problem 3. [Intrusion Response] (6 points)

The software company Snoracle (slogan: "Unwakeable") is selling a new defense against DDoS attacks. Their software looks at the source IP address on all incoming packets, and if it finds any IP address that accounts for more than 1% of traffic over the last hour, it installs an entry in the router that blocks all packets from that address for the next 24 hours. Their marketing folks are claiming that this will stop all DDoS attacks cold in the water. Is this a good solution to the problem? Give one reason why or why not.

No. It's a poor solution.

- *It's too easy to break: with more than 100 zombies, you can flood the victim's network link without any zombie consuming more than 1% of traffic.*
- *It's too easy to evade detection with forged source addresses. You just use a different (forged) IP address on every packet.*
- *It doesn't protect against attacks that overwhelm resources at the end host (e.g., CPU, memory) without filling the network pipe.*
- *Attackers could exploit this to cause collateral damage to innocent third parties. If CNN is using this, an attacker could prevent Joe from being able to reach CNN by sending a large number of packets whose IP address has been forged to look like they came from Joe.*

Any one of these was an acceptable reason.

Problem 4. [Hardware Support for Dual-Mode Operation] (6 points)

Early Intel processors (e.g., the 8086) did not provide hardware support for dual-mode operation (i.e., support for a separate user mode and kernel mode). As a result, most of the systems implemented on these processors did not support multi-user operation. List and explain one potential problem associated with supporting multi-user operation without hardware support for dual-mode operation. Be concise: one or two sentences should suffice.

A few sample solutions (any one is acceptable):

- *A rogue process could modify the address space of other processes or of the kernel.*
- *A rogue process could disable interrupts and avoid getting re-scheduled, so that the rogue hogs all the CPU and other process don't get a chance to run.*
- *There's no distinction between privileged vs unprivileged instructions, so a rogue process could send I/O commands directly to attached peripherals. For instance, a rogue process could trash the hard disk or snoop on network packets.*

Problem 5. [Shamir Secret Sharing] (10 points)

In this question, you have a chance to explain your understanding of the Shamir Secret Sharing system. Let s be a secret that is supposed to be shared. Let p be a large prime, let n be the number of share holders, and let q be a quorum of share holders (i.e., a threshold so that it will take at least q shares to recover the secret s). Assume that $1 < q < n < p$.

(a) Explain how Shamir creates a function $f(x)$ such that: $f(0) = s \pmod{p}$; and $f(1) \bmod p$ equals the first share; and $f(2) \bmod p$ equals the second share; etc. Be explicit.

Let $f(x) = s + a_1 x + \cdots + a_{q-1} x^{q-1}$, where $f_1, \ldots, f_{q-1}$ are chosen at random $\bmod p$. The ith share is then computed as $f(i) \bmod p$.

(b) Explain briefly (maximum of 50 words!) why one can recover s when one has values of $f(x) \bmod p$ for at least q distinct values of x (where $0 < x < p$).

Solution #1: Lagrange interpolation. Suppose we know $f(a_i) = b_i \pmod{p}$. Define the function δ_i by $\delta_i(x) = \prod_{j \neq i}(x - a_j)$, and let $f(x) = \sum_i b_i \delta_i(a_i)^{-1} \delta_i(x)$. Then $f(0) \bmod p$ is the secret s.

Solution #2: Linear algebra. Equate $f(x) = f_0 + \cdots + f_{q-1} x^{q-1}$, where $f_0, \ldots, f_{q-1}$ are q formal unknowns. Each equation of the form $f(a) = b \pmod{p}$, for some known a, b, yields one linear equation on the unknowns $f_0, \ldots, f_{q-1}$, namely, $b = f_0 + \cdots + f_{q-1} a^{q-1} \pmod{p}$. With q values of $f(x)$, we get q linear equations in q unknowns, and we can solve for $f_0, \ldots, f_{q-1}$. Then f_0 is the secret s.

Problem 6. [Key Distribution with Mutual Authentication] (10 points)

Here is a version of an authentication protocol discussed in class. In this protocol Alice and Bob wish to authenticate. There is a trusted authority T which generates a fresh random session key K and distributes it to Alice and Bob. Alice has a symmetric key K_A that is shared with T. Bob shares symmetric key K_B with T.

In the notation below $x \rightarrow y : m$ means that x sends message m to y. $\{m\}_k$ means that message m is encrypted with symmetric key k. Messages in quotes are literals that are transmitted. For example, the first line means Alice sends the trusted authority T a message saying “I want to authenticate with Bob” and that message is encrypted with key K_A.

$$\begin{aligned}
\text{Alice} \rightarrow \text{T}: &\quad \{\text{“I want to authenticate with Bob”}\}_{K_A} \\
\text{T} \rightarrow \text{Alice}: &\quad \{\text{“Use session key”}, K, \text{“and send Bob this message”}, \\
&\quad\quad \{\text{“This is Alice using session key”}, K\}_{K_B}\}_{K_A} \\
\text{Alice} \rightarrow \text{Bob}: &\quad \{\text{“This is Alice using session key”}, K\}_{K_B}
\end{aligned}$$

Alice and Bob now share key K and can use K to secure all future messages between them.

(a) This authentication and key-exchange protocol is subject to a replay attack. Explain how the replay attack would work.

Attack #1: An eavesdropper can observe the third message (from Alice to Bob) and any subsequent traffic that Alice sends encrypted under K to Bob. Later, the eavesdropper can replay the third message and subsequent traffic to Bob, and Bob will think that the replay came from Alice.

Attack #2: An eavesdropper can observe the entire three-message exchange and all subsequent traffic sent by Alice or Bob. Later, if Alice begins to request another session to Bob, the attacker replace T’s response with the second message of the prior session, and can then replay any of the traffic from the prior session (since Alice and Bob will then re-use the same key K for both sessions).

(b) Assume Alice, Bob, and T have synchronized clocks. Show how to modify the messages of the above protocol to defend against replay attacks. The only changes to the protocol permitted are to add additional values to the three messages in the protocol (but you may not delete any values or otherwise change the structure of the message flow). Make the minimum number of changes to the protocol necessary for security, and be precise about exactly where your new added values will go.

Below, t represents a timestamp chosen by Alice. New additions are underlined.

$$\begin{aligned}
Alice \rightarrow T: &\quad \{\text{“I want to authenticate with Bob}\underline{\text{ at time”}, t}\}_{K_A}\\
T \rightarrow Alice: &\quad \{\text{“Use session key”}, K, \underline{\text{“at time ”}, t,}\text{“and send Bob this message”},\\
&\quad\quad \{\text{“This is Alice using session key”}, K, \underline{\text{“at time”}, t}\}_{K_B}\}_{K_A}\\
Alice \rightarrow Bob: &\quad \{\text{“This is Alice using session key”}, K, \underline{\text{“at time”}, t}\}_{K_B}
\end{aligned}$$

Each party that receives a timestamp should check that it is current, and if not, terminate the session.

Problem 7. [Gesundheit] (12 points)

Kachoo!, Inc. has just released a new web service that allows people to sign their web pages. The service does this by appending, hidden inside a special HTML tag at the bottom of an otherwise normal web page, the author's name, the date, and a signature (which contains the author's name and date signed by the author's RSA private key). The web page itself is unencrypted, but the signature can be validated by downloading `http://www.kachoo.com/pubkeys.html` (which contains a list of all registered Kachoo! users and each user's public key) to retrieve the author's public key.

Explain why this gives a completely false sense of security, by outlining two different ways that you could make it appear that Linus Torvalds has posted a web page saying "Open source is for losers; I've decided to go work for SCO". The definition of "different" is that each attack has a unique fix. For each of the attacks you list, give a countermeasure that the author/viewer could take to protect themselves against that attack.

Attack #1: Wait for Linus to post some other message on his web site. Copy the name, date, and signature, but modify the contents of the message. The viewer will still receive a valid signature and be fooled.

Countermeasure #1: The contents of the web page should also be included in the input to the signature.

Attack #2: When the viewer downloads `http://www.kachoo.com/pubkeys.html`, corrupt the response (e.g., send a spoofed response packet) so that it contains a listing for Linus with a public key that is not his. This corruption is possible, since the `pubkeys.html` is downloaded over insecure HTTP. The attacker can generate his own keypair and list Linus's name next to the attacker's public key. Then, the attacker can create a web page that is validly signed using this keypair, fooling readers.

Countermeasure #2: Secure distribution of `pubkeys.html`. For instance, it might be distributed over SSL. Or, it might be signed with Kachoo!'s private key, and a copy of Kachoo!'s public key might be embedded in every web browser so that the browser can check that this page has not been corrupted.

Problem 8. [One is the Loneliest Number] (10 points)

In this class, we have seen several different mechanisms for isolating untrusted programs, including virtual memory, system call interposition, and virtual machines.

(a) Name one threat that system call interposition protects against but virtual memory does not.

Opening a network connection (e.g., to attack other machines).

Opening files (e.g., to read secret files or modify the user's data).

Sending signals to other processes (e.g., to kill them).

(b) The military runs a multi-user computer that all government employees can log into; programs that require access to top-secret data are run inside a virtual machine. Richard Stallman is given an account on this computer so that he can install emacs. Colonel Greene runs a copy of Stallman's emacs program inside a virtual machine and uses it to edit the top-secret list of UFOs stored in Area 51's warehouses. (Only Greene has an account on the guest OS running inside the virtual machine.) If Richard Stallman were malicious, could he arrange to learn the contents of this list? If yes, explain how; if no, say why not.

Yes. He could embed a Trojan horse in emacs that uses a covert channel to leak out the contents of the UFO list. For instance, emacs might module the system load to communicate the contents of the file it is editing (1 = do heavy computation for one second, 0 = do nothing for one second). Richard could use his account to monitor the system load and thus receive the secret information that is being leaked by his Trojan'ed emacs.

Problem 9. [And if you believe that, ...] (11 points)

Consider the zero-knowledge protocol presented in lecture that gives a proof of knowledge of a square root x of y mod N, where $N = PQ$ is the product of two large primes:

1. The prover sends random $s = r^2 \bmod N$.
2. The verifier sends either challenge I) $s \bmod N$ or II) $sy \bmod N$.
3. The prover answers the challenge with either I) $r \bmod N$ or II) $rx \bmod N$.

(a) Suppose that the verifier is later talking to Christina and wishes to convince her that the prover really does know a square root of y mod N. So he sends Christina the transcript of his conversation with the prover. Should Christina be convinced? Why or why not?

No. Anyone can generate a fake (but valid-looking) transcript just by running the simulator; they don't need to know a square root to come up with a valid-looking transcript.

(b) Suppose that the verifier cannot directly communicate with the prover in the above protocol, but instead all the messages between the prover and verifier are relayed through Danielle. Assume that the prover and the verifier sign every message they send, and they know each other's public key. Should the verifier be convinced at the end of the protocol that the prover knows a square root of y mod N? Justify your answer.

Yes. The verifier knows that the messages are coming from the prover (Danielle cannot tamper with them without detection), and anyone who can answer the challenge must know a square root of y.

(c) Under the same assumptions as part (b), should Danielle be convinced at the end of the protocol that the prover knows a square root of y mod N? Justify your answer.

No. The two endpoints might have arranged to conspire to fool Danielle by generating a fake (but valid-looking) transcript in advance, and then relaying each successive message in the fake transcript through Danielle. They can do this even without knowing a square root of y (see part (a)).

Problem 10. [True/False, and Why?] (18 points)

Answer each question true or false, and then say why in at most one sentence. *Don't forget to provide a one-sentence justification for each part!*

(a) True or [✓ False]: "Security through obscurity" refers to the practice of obscuring a user's password when the user types it in, so that no one else can see it on the screen.

Justification: Actually, security through obscurity refers to keeping the code of your system secret, and hoping this will keep you secure (bad idea!).

(b) [✓ True] or False: Let E_k be a secure block cipher. Then the following encryption scheme is secure against chosen-plaintext attack: to send a message m, the sender picks a random r and sends the two strings r and $E_k(r) \oplus m$.

Justification: The value $E_k(r)$ is indistinguishable from a uniformly random value, and xor-ing m with a uniform random secret value doesn't reveal m (due to the security of the one-time pad).

(c) True or [✓ False]: Suppose I generate a RSA public and private key pair, and I publish the public key. Then that's all I need to be able to send you a securely encrypted email.

Justification: You need to have a public key of your own, if I want to encrypt messages to you.

(d) True or [✓ False]: If Jimmy Neutron tomorrow discovers an efficient algorithm for computing the greatest common divisor of two extremely large numbers, this will make it possible to break RSA.

Justification: Actually, it is already known how to compute the gcd of large numbers efficiently, but RSA still seems secure.

(e) True or [✓ False]: Consider data that is stored over time in a mandatory access control based system. The contents of files containing highly classified ("top secret") information are necessarily more trustworthy than material stored in files marked unclassified.

Justification: In a mandatory access control system, anyone can write to a file marked classified, while only some users (i.e., those who are not cleared for classified data) can write to unclassified files.

(f) [✓ True] or False: Access control matrices can represent anything that is represented by access control lists.

Justification: ACLs are just one representation of an access control matrix—namely, the rows of the matrix.