CS 161 Computer Security

Fall 2006 Joseph/Tygar

MT 2 Solutions

Problem 1. [Covert Channels] (30 points)

(a) (5 points) Write down the Fiat-Shamir zero-knowledge protocol (as presented in class) where Alice proves her identity to Bob with probability 50% each iteration.

Care needs to be taken that one is having Alice authenticate, as the problem requests, and not Bob. Given a fixed pq, product of large primes, all values are taken modulo pq. A has previously published a^2*, a known only to A. For a random r, A sends* r^2*, B replies with a random bit, and depending on the bit, A sends r or ar. This is iterated n times, to establish A's identity with probability* $1-2^{(-n)}$.

(b) (5 points) Identify all the covert channels in the Fiat-Shamir protocol that **Alice** can use to leak information to **Bob**.

Methods include timing of messages, chosen values of r, and the actual final message sent by A, regardless of the demands of the protocol (e.g., deliberate failure).

(c) (5 points) Identify all the covert channels in the Fiat-Shamir protocol that **Bob** can use to leak information to **Alice**.

Methods include timing of message and chosen values of bits.

(d) (5 points) In view of the above covert channels, in what sense is it fair to call Fiat-Shamir a zero-knowledge protocol — since it leaks information how can it be "zero knowledge"?

The question of covert channels is orthogonal to the question of zero-knowledge. A zero-knowledge protocol has no side-channels.

(e) (5 points) How do nonces present an opportunity for covert channels?

The bits of the nonce itself may contain covert messages.

(f) (5 points) How can we limit leakage of covert channel information via nonces?

This is a bit of a trick question: it is quite difficult, and indeed could be said to be practically impossible. While a scheme could be proposed using a trusted third party to generate signed nonces, the details of such a scheme would likely introduce new security flaws.

Problem 2. [Isolation Techniques] (26 points)

(a) (9 points) Briefly describe the type of isolation used by qmail for security purposes.

Qmail uses three types of isolation (3 points each):

1. *The e-mail functions are divided into separate processes using OS mechanisms.*
2. *Each process is given only the privileges that it needs to accomplish its tasks. This "Least Privileges" model is implemented using different user accounts and is enforced by OS mechanisms.*
3. *The amount of code in each process is limited to the bare minimum. By limiting the amount of code, it is easier to verify the correctness of the code.*

We subtracted one point for saying that qmail uses separate modules, instead of separate processes.

(b) (9 points) Briefly explain why the process isolation used by qmail is insufficient from a security point-of-view.

If one of qmail's processes is compromised, it can be used to attack other processes on the same machine, to attack other machines on the network, or to send spam. A compromised process can do any actions permitted by the user account privileges.

We subtracted 3 to 6 points if you did not clearly explain what a compromised program could do.

(c) (8 points) Briefly explain what system call interposition is.

System call interposition is a Reference Monitor for all system calls that is interposed between an application and the operating system (3 points). The reference monitor checks all calls and their arguments against a list of access control rules to determine whether the call is permitted or not (3 points).

We gave an extra two point bonus for Problem 2 if you explained the potential TOCTTOU vulnerability for a naive reference monitor implementation. The bonus was only used to offset any points lost on the problem (the maximum point value for Problem 2 is 26 points).

Problem 3. [Random Number Generation] (24 points)

(a) (8 points) Bob brilliant has come up with a cryptographically secure pseudorandom bit generator that takes a seed of 128 bits. Show how to turn this into a symmetric cryptosystem with a key of 128 bits.

If Bob shares a 128-bit key with Alice, he can seed his PRBG with the key and produce a stream of output bits. He can generate a stream of bits as long as the message he wants to send and XOR the stream with the message to produce a ciphertext. Alice can seed her copy of the PRBG with the same key to produce the same string of bits, which she then XORs with the ciphertext to recover the message. This is a one-time pad with pseudorandom bits, so it is important not to re-use the same part of the stream for more than one message.

The most common mistake was to suggest using the PRBG to generate a key for AES or some other symmetric cipher. Not only does that not answer the problem (because we wanted you to turn the PRBG into a cryptosystem, not use it as a small part of a larger cryptosystem), but if we have 128 truly random bits to seed the generator with, we could use that as a key and wouldn't need the generator at all. This answer received 0–2 points depending on what exactly it said.

(b) (8 points) Given a 128-bit seed, what is the maximum number of pseudorandom bits Bob can expect to get from his generator?

There is no absolute limit on the number of bits Bob can expect to get from his generator.

Just writing 2^{128} *was worth 2 points; that is the upper limit on the number of unique sequences of bits, but each sequence has no* a priori *length limit. Explaining why the particular PRNG we saw in class (using cipher block chaining as in AES-CBC(seed,* 0^n*)) has a maximum length of* 2^{135} *bits before it repeats was worth 4 points.*

(c) (8 points) If a pseudorandom number generator passes statistical tests for randomness, what additional property does it need to be cryptographically secure?

The property is that an attacker should not be able to guess any number/bit with much more than uniform probability even if he/she has access to the PRNG's entire past history.

Just saying "unpredictable" or "indistinguishable from truly random" without mentioning previous bits/numbers was 6 points.

4 points for saying each bit shouldn't be generated *from the previous bits without mentioning unpredictability.*

-1 *for saying "if an attacker can* guess *some bits…" This misses the point that PRNG output is not supposed to be secret, and you should expect an attacker to know some of it.*

-2 *for saying statistical randomness means there is no detectable pattern.*

$+2$ *lost points back for saying the seed needs to be chosen well and unguessable/secret (though these points won't bring you above 8 on this part).*

Problem 4. [Firewalls] (20 points)

(a) (10 points) You are given a firewall that can examine the contents of packets, including reconstructing connection streams. What types of buffer overflow attacks can it protect against, if any? What types of buffer overflow attacks can it not protect against, if any? Explain your answers briefly.

There are many correct answers. For full credit, give enough of these to add up to 10 points.

3 points: *Defends against known buffer overflow vulnerabilities in known services.*

3 points: *Defends against all attacks against blocked services.*

4 points: *Does not defend against unknown attacks on open services.*

3 points: *Does not protect against buffer overflows in the firewall itself.*

3 points: *Does not find buffer overflows triggered by encrypted packets.*

1 point: *Does not defend against overflows triggered by packets that do not traverse the firewall.*

3 points: *Does prevent malicious code injection by matching signatures of known malicious code.*

(b) (10 points) Explain how you could use such a firewall to protect against transmitting unencrypted credit card numbers over the network.

10 points: *Compare the packets against a pattern that matches credit card numbers.*