Wagner
Spring 2014

CS 161 Computer Security Final Exam Practice

This is a review worksheet with sample exam problems, to help you study for the final exam.

Problem 1 ***True or False*** **(0 points)**

(a) True or false: Javascript runs with the origin of the page that loaded it.

(b) True or false: Cross Site Request Forgery occurs when an attacker manages to leave malicious Javascript on a server and have the server run the malicious Javascript when the victim loads the page.

(c) True or false: All versions of TLS provide perfect forward secrecy.

(d) True or false: TLS usually provides confidentiality, integrity, and mutual authentication between the client and server.

(e) True or false: If CBC is used properly with a truly random IV, it provides protection against tampering with the encrypted messages.

(f) True or false: DNSSEC uses TCP.

(g) True or false: TLS only works if we trust certificate authorities.

(h) True or false: ECB mode is generally preferable to CBC mode, when we need symmetric-key encryption.

(i) True or false: CBC is widely used because both encryption and decryption can be done efficiently in parallel.

(j) True or false: A major reason that SSL/TLS is not used everywhere is because of the high cost of public-key cryptography.

(k) True or false: It is believed that Stuxnet was created by a government entity.

(l) True or false: It usually takes thousands or millions of spam emails to get one person to make a purchase.

(m) True or false: Whether you use Tor or not can not be detected by your ISP.

(n) True or false: Tor protects against disclosure of your communication contents.

Problem 2 ***Short answer*** **(0 points)**

(a) What are two security properties that TLS provides?

(b) Name one security property that TLS does not provide.

(c) Which protocol is more reliable in delivering packets, TCP or UDP?

(d) What are the main security requirements for an e-voting system?

(e) List two reasons why cryptographic modes of operation (e.g., CBC mode) are used, instead of just using the block cipher directly to encrypt.

(f) List two ways that a certificate authority can deal with bogus certificates (e.g., a certificate that was issued erroneously).

(g) In TLS, at the end of the handshake each party transmits a MAC over the previous messages to the other party. Describe an attack that could be launched against TLS if the MACs were not included in the protocol.

(h) How is a stream cipher different from a one-time pad?

(i) What is the principle of least privilege?

(j) You are the software developer of a web forum. You hire a security auditor, and she reports two vulnerabilities in your code. One is a reflected XSS, and the other is a SQL injection. You plan to fix both, but you must decide which one to fix first. All else being equal, which of these two vulnerabilities is more serious? Why?

(k) Consider a hierarchical PKI (certificate authority) model, and a certificate chain of length 3 (i.e., consisting of three certificates). Imagine that the attacker might be able to bribe some certificate authorities or signers to issuing malicious/bogus certificates; of course, this is expensive, so the attacker wants to minimize the number of bribes. How many of the certificates in the chain must be bogus, if the attacker wants to fool a legitimate user into accepting this certificate chain as valid?

(l) Fill in the blank: In the 'Web of Trust' model, you 'trust' all of your friends and, implicitly, you also trust _____.

Problem 3 ***Software security*** **(0 points)**

Consider the following vulnerable C function:

```
int f() {
    char name[80];
    int ch, i = 0;
    printf("Your first name?\n");
    while ((ch = getchar()) != EOF) {
        name[i] = ch; i = i+1;
    }
    name[i] = '\0';
    printf("Nice to meet you, %s!\n", name);
    return 0;
}
```

(a) What precisely is the vulnerability in this code? How can it be exploited?

(b) You start looking at the assembly code that the compiler has generated for this function, and you notice that the function epilogue looks like this:

```
movl %eax, 0x1eaff00d // an address in the .data section of memory
movl %eax, 0(%eax)
```

```
movl %esp, %ebp       // restore the stack pointer to the top of the frame
cmp %eax, -4(%esp)    // check if %eax equals value at address %esp-4
jne <stack_smashed>   // if they are different, jump to code that kills the program
ret
```

Apparently, the compiler has inserted this epilogue. Looking further, it appears that before `main()` runs, some compiler-inserted initialization code generates a random 32-bit number and stores it at address `0x1eaff00d`; every function prologue pushes this onto the stack, and every function epilogue has the check shown above.

What buffer overflow defense has the compiler implemented?

(c) What kinds of buffer overflows will this detect? What kind won't it detect?

(d) Will the buffer overflow defense that the compiler inserted detect your exploit from part (a)?

(e) Name one method that an attacker might be able to use to bypass the compiler's defense. What conditions does the code need to satisfy, to make this method possible?

Problem 4 ***Cryptography*** **(0 points)**

Assume Alice and Bob have securely/privately exchanged two random secret values (one from Alice and one from Bob). They concatenate these and use them as the seed for a cryptographically secure pseudo random number generator. They will use the pseudo random number generator to produce two secret keys, K_E for encryption and K_I for integrity. From there, they communicate using symmetric cryptography with these keys.

For example:
Alice sends to Bob: $\text{AES-CBC}_{K_E}(M), \text{MAC}_{K_I}(\text{AES-CBC}_{K_E}(M))$
Bob sends to Alice: $\text{AES-CBC}_{K_E}(M'), \text{MAC}_{K_I}(\text{AES-CBC}_{K_E}(M'))$.

(a) Name one problem with this scheme. What kind of attack can Malice, an active attacker, make?

(b) Does this scheme guarantee confidentiality from a passive eavesdropper? Yes or no.

(c) How can Alice and Bob make this scheme secure?

Problem 5 ***Certificate chains*** **(0 points)**

(a) Explain the concept of a certificate chain, and give a concrete example of one.

(b) In 2011, a bug was found in the iPhone Safari's implementation of verifying certificates. It would verify as good all certificate chains with a valid certification path to a trusted CA. In other words, if the signer of the topmost certificate was a trusted CA, the certificate chain was accepted by iPhone Safari as valid.

With this knowledge, explain how an attacker could mount an attack on `www.amazon.com` and all users attempting to access that website, even when using TLS/SSL.

Problem 6 ***Covert channels*** **(0 points)**

You are the adminstrator for a school computer network. You give students remote access to the same computer using different student accounts. However, you don't want students who are logged in to the same computer to communicate with each other because they could cheat together. You have taken great measures to block any covert channels, but you may have forgotten about physical devices...

Two students are taking a test in a different room, and they are both remotely logged in to the same testing computer. The testing computer has access to a microphone and a speaker system.

(a) Explain how the students can use the microphone and speaker system as a covert channel.

(b) You attempt to block the attack by not allowing access to the microphone while the speaker is being used, and vice versa. Explain how the students could still communicate through a covert channel.

Problem 7 ***Two-factor authentication*** **(0 points)**

A two-factor authentication scheme might be implemented by requiring users to enter in a 4-digit temporary code along with their more permanent normal password. The idea is that an attacker might learn the normal password, but the attacker would not be able to generate the temporary code.

Your friends are creating a new startup called 2Fac2Cool which they claim will disrupt the entire industry with its awesome scheme.

Their scheme: You download an app on your phone. You enter in your normal password to set up the app. The two-factor code is the last 4 digits of SHA256($p \,||\, h$) (viewed as an integer in decimal), where p is your normal password and h is the number of hours since January 1, 1970.

Your friends claim that the server can verify this because it knows the time and the normal password. The temporary code would change every hour.

(a) 2Fac2Cool claims that even if an attacker learns your normal password, they still wouldn't be able to generate the second-factor temporary code. Are they right?

(b) 2Fac2Cool hired a security auditor and they decided to add one extra step to the setup. When you set up the app, 2Fac2Cool will text you a random 2048-bit secret s which the app automatically detects and stores. The new scheme is:

The second-factor temporary code is the last 4 digits of SHA256($p \,||\, h \,||\, s$). The server knows the time, the secret s, and your normal password p, so it can verify your code.

In this revised scheme, if an attacker learns your normal password, can they generate or predict the second-factor temporary code?

(c) Suppose Mallory knows your normal password. At lunch time you write down your second-factor temporary code on a slip of paper because your phone ran out of

battery. You inadvertently leave the slip of paper behind and Mallory picks it up a few days later. 2Fac2Cool says you are safe (with the revised scheme) because the temporary codes only last one hour. Are they right?

Problem 8 ***Broken hash functions*** **(0 points)**

Recall that we used a cryptographic hash function in RSA digital signatures and for storing passwords in a database. Recall that a digital signature scheme is secure if an attacker cannot create a valid message-signature pair, and if the signer cannot deny that (s)he signed the message.

Now suppose we find out that our "cryptographic hash function" actually has a flaw. Explain whether the RSA digital signature is still secure when using the broken cryptographic hash function. If you say that it is insecure, describe an attack that shows that it is insecure.

(a) The hash function does not have preimage resistance. In other words, given a hash value y, it is possible to find x such that $H(x) = y$.

(b) The hash function is not collision resistant. So, it is possible to find an x and a y such that $H(x) = H(y)$. (However, the hash function *is* preimage resistant and second-preimage resistant.)

Another use of cryptographic hash functions is to help us construct a way to securely store passwords in a database. Recall that a password hashing scheme is secure if an attacker cannot recover the passwords from the information stored in the database. The password hashing scheme we recommended is to append a salt to the password, then iteratively hash that a bunch of times (say, n times).

Do the same as parts (a) and (b), but now for this password hashing scheme. In other words, explain whether this password hashing scheme is still secure when using the broken cryptographic hash function. If you say that it is insecure, describe an attack that shows that it is insecure.

(c) The hash function does not have preimage resistance. In other words, given a hash value y, it is possible to find x such that $H(x) = y$.

(d) The hash function is not collision resistant. So, it is possible to find an x and a y such that $H(x) = H(y)$. (However, the hash function *is* preimage resistant and second-preimage resistant.)

Problem 9 ***Modulus re-use in RSA*** **(0 points)**

Alice, Bob, Charlie, and Mallory use RSA to communicate with one another on a regular basis. You may assume that everyone knows everyone's public keys. One day Mallory notices that Alice and Bob have chosen the same RSA modulus N but different exponents e_a, e_b. Mallory knows that the next day Charlie will be sending Alice and Bob a message M containing the location of a secret meeting among the three of them. In particular, the same message will be sent to both Alice and to Bob, but the version sent to Alice will be encrypted with Alice's public key and the one sent to Bob will be encrypted with Bob's public key. If Mallory can intercept both ciphertexts C_a and C_b, can Mallory

uncover the location of the secret meeting? Explain.