Homework Assignment #1 ECE 548/488 Answer Key Cyber Threats and Security Management Fall 2017

1

1) (1.5) Consider the following general code for allowing access to a resource:

DWORD dwRet = IsAccessAllowed(...);

If (dwRet == ERROR\_ACCESS\_DENIED) {

// Security check failed.

// Inform user that access is denied.

} else {

// Security check OK.

}

a. Explain the security flaw in this program.

b. Rewrite the code to avoid the flaw.

*Hint:* Consider the design principle of sail-safe defaults.

**Answer:**

a. This program will always authenticate unless the value of dwRet is not equal to ERROR\_ACCESS\_DENIED. The flaw lies when there is error for any reason and results in dwRet not being a value equivalent to ERROR\_ACCESS\_DENIED, then the user will be authenticated.

b. A simple solution to this problem is to check if there is not an error and an authenticated message is returned. Otherwise, the security check fails and you inform the user that access is denied as follows:

DWORD dwRet = IsAccessAllowed(...);

If (dwRet == ACCESS\_GRANTED) {

// Security check OK.

} else {

// Security check failed.

// Inform user that access is denied.

}

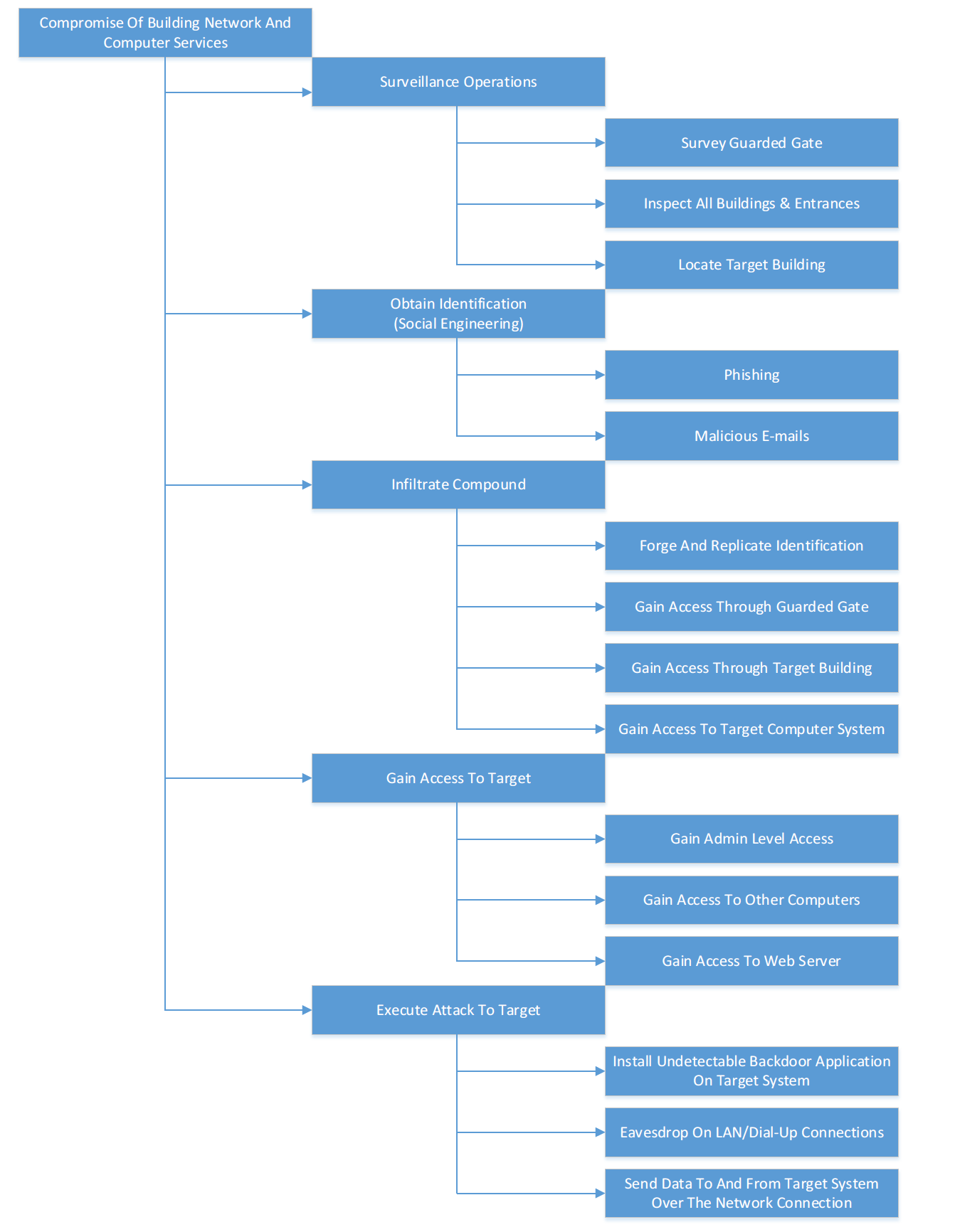
Homework Assignment #1 ECE 548/488 Answer Key Cyber Threats and Security Management Fall 2017

2

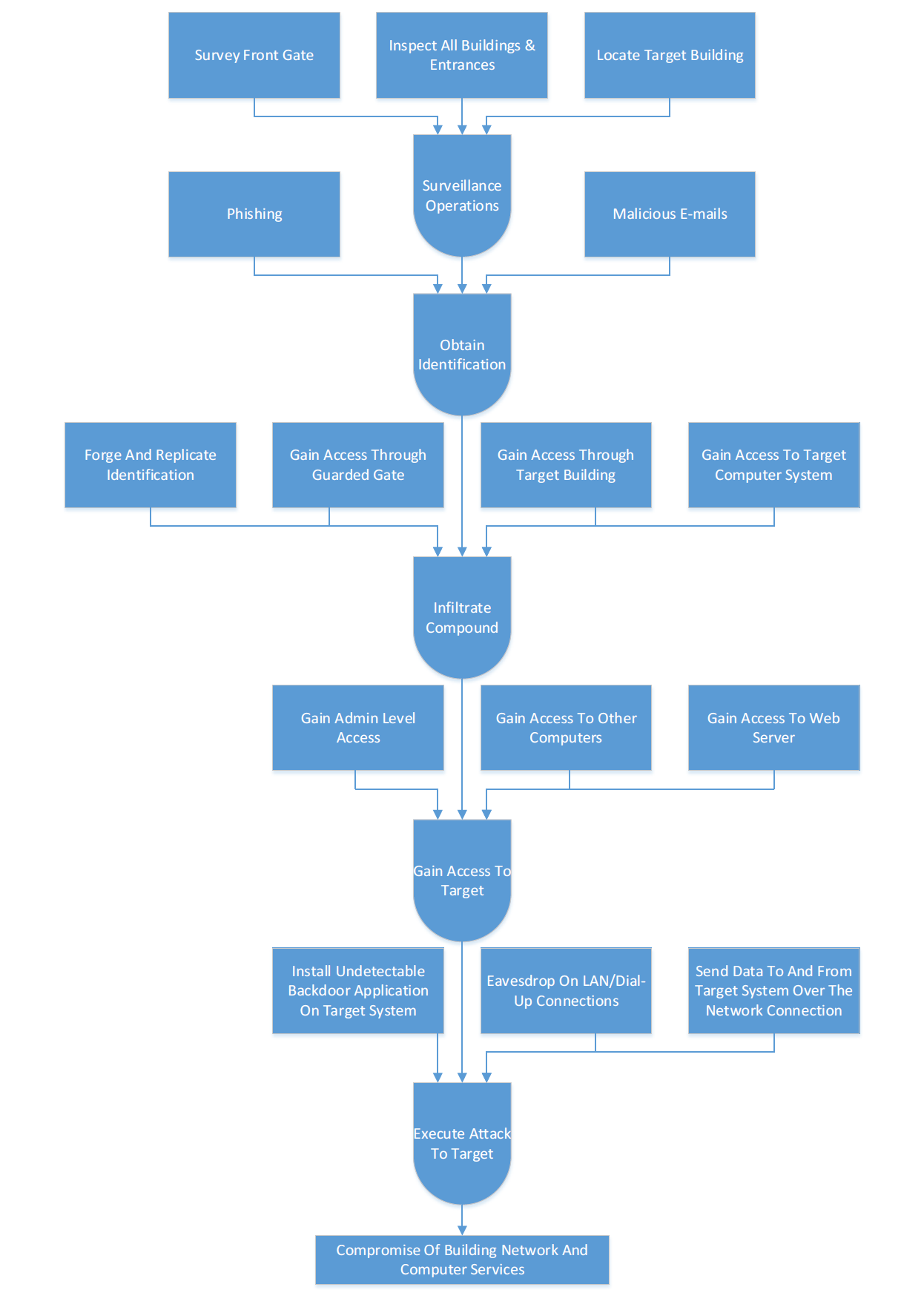
2) (1.7) Consider a company whose operations are housed in two buildings on the same property, one building is headquarters, the other building contains network and computer services. The property is physically protected by a fence around the perimeter. The only entrance to the property is through the fenced perimeter. In addition to the perimeter fence, physical security consists of a guarded front gate. The local networks are split between the Headquarters’ LAN and the Network Services’ LAN. Internet users connect to the Web server through a firewall. Dial-up users get access to a particular server on the Network Services’ LAN.

Develop an attack tree in which the root node represents disclosure of proprietary secrets. Include physical, social engineering, and technical attacks. The tree may contain both AND and OR nodes. Develop a tree that has at least 15 leaf nodes.

**Answer (Textbook Style):**



**Answer (Tanner Studley Style):**



Vanessa Soares

Professor Hong Liu

ECE 488: Cyber Threats and Security Management

Homework 2

15 September 2017

Due: 21 September 2017

2.1

Suppose that someone suggests the following way to confirm that the two of you are both

in possession of the same secret key. You create a random bit string the length of the key, XOR

it with the key, and send the result over the channel. Your partner XORs the incoming block with

the key (which should be the same as your key) and sends it back. You check, and if what

you

receive is your original random string, you have verified that your partner has the same secret

key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme?

In order to determine if there is a flaw in this scheme, we must ana

lyze it in order to determine

if the sender and receiver use the same secret key.

To confirm that the receiver is using the same secret key, the sender transmits a random bit

string that is the same length as the key. The sender XORs the key with the rand

om bit string,

and sends it to the receiver. The receiver XORs the string from the sender with their secret key

and transmits it back to the sender. If the sender receives the same string as their random bit

string, then it is determined that they are usin

g the same secret key.

The main flaw in this scheme arises if someone is listening to the communication between the

sender and the receiver. By listening to the communication, they are able to obtain the secret

key by performing an XOR operation using the

two transmitted strings. I will prove this using

an example.

Let’s say that the random bit string sent from the sender to the receiver is 101101 and the

secret key is 101001.

When I perform the XOR operation for these two strings, I get 100 as the stri

ng.

Random bit string → 101101

Key → 101001

Transmitted String 1 → 100

The receiver uses their secret key of 101101 and XORs it with the received 100 key.

Received string → 100

Key → 101001

Transmitted String 2 → 101101

The receiver will send the string 101101 back to the sender. The sender checks whether the

101101 is the same as the random bit string. The two strings match and it is therefore

determined that the sender and receiver are both using the same secret key.

The person listening to the communications will have obtained the two transmitted strings of

100 and 101101. By applying the XOR operation to the strings:

Transmitted String 1 → 100

Transmitted String 2 → 101101

Observed Key → 101001 (secret key)

The observed result when applying XO

R to the two transmitted strings is the secret key that

was kept by the sender and the receiver. The person listening to the communications can

therefore obtain the secret key by XORing the two transmitted keys. Hence, there is a flaw in

this scheme.

Yes.

The eavesdropper is left with two strings, one sent in each direction, and their XOR is the

secret key.

2.5

In this problem we will compare the security services that are provided by digital

signatures (DS) and message authentication codes (MAC). We

assume that Oscar is able to

observe all messages sent from Alice to Bob and vice versa. Oscar has no knowledge of any keys

but the public one in case of DS. State whether and how (i) DS and (ii) MAC protect against

each attack. The value auth(x) is comput

ed with a DS or a MAC algorithm, respectively.

a.

(Message integrity) Alice sends a message x = “Transfer $1000 to Mark” in the clear and

also sends auth(x) to Bob. Oscar intercepts the message and replaces “Mark” with “Oscar.” Will

Bob detect this?

Alice sends a message to Bob and Oscar intercepts the message, replaces some information,

and sends it to Bob.

Digital Signatures (DS) Protection Against This Attack

→ The digital code is generated using the hash of the message. Then, it is encrypted usi

ng the

sender’s private key and it is added as a signature.

→ DS assures that the message is from the authorized party who it claims to be and guarantees

that the message wasn’t changed in between.

Message Authentication Codes (MAC) Protection Against T

his Attack

→ The MAC is generated based on the message and the secret key using the hash of the

message. Then, it’s encrypted using the sender’s private key and added as a signature.

→ The MAC is attached to the message at the sender’s side while transmit

ting.

→ On the receiver’s end, they receive the message along with the MAC. The message alone

with the secret key is inputted to the MAC algorithm that generates the MAC.

→ The obtained and received MAC are compared to verify message authentication.

→ If

they match, then the receiver can be assured that the message was received from an

authenticated sender and wasn’t modified along the way.

Therefore, both DS and MAC will detect this change of information/message integrity attack

and Bob will detect this

as well.

This will be detected with both DS and MAC.

b.

(Replay) Alice sends a message x = “Transfer $1000 to Oscar” in clear and also sends

auth(x) to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will

bob detect this

?

→ Alice sends a message to Bob and while it’s being sent, Oscar sees the message and

signature and sends it to Bob 100 times.

→ There’s no actual modification to the message. Therefore, sending the same message

repeatedly can’t be identified by Bob.

→

Both DS and MAC only assure that the message was received by the authenticated sender

without any changes.

→ Therefore, repeatedly sending the same message won’t be identified by the sender when

using either DS or MAC.

This won’t be detected by either. Re

mark (use time

-stamps)

c.

(Sender authentication with cheating third party

, i.e., Oscar is bad) Oscar claims that he

sent some message x with a valid auth(x) to Bob but Alice claims the same. Can Bob clear the

question in either case? How can Bob prove

the third party Oscar is cheating in DS and in MAC?

→ Bob receives a message. Oscar claims that he sent the message and Alice claims that she

sent the message.

→ DS allows Bob to verify the signature of the sender by decrypting the signature with both

public keys of the senders that result in the hash message.

→ If the generated hash message matches the received hash message, then the message was

received from the authenticated sender.

→ Bob does that to both public keys to verify the authenticated sen

der.

→ With MAC, Bob can ask for the secret key from both of the users. He has the secret key

already in this case.

→ The one who reveals the right secret key is obviously the authenticated sender.

→ Therefore, Bob can use both DS and MAC to identify

whether the message was received

from the authenticated sender.

→ DS: Bob simply has to verify the message with the public key from both. Obviously, only

Alice’s public key results in a successful verification.

→ MAC: Bob has to challenge both Oscar and A

lice to reveal their secret key to him (he

already knows it anyway). Only Bob can do that.

d.

(Authentication with Bob cheating) Bob claims that he received a message x with a valid

signature auth(x) from Alice (e.g., “Transfer $1000 from Alice to Bob”)

but Alice claims she has

never sent it. Can Alice clear this question in either case?

→ Bob claims that he got a message from Alice. Alice says that she didn’t send the message.

→ With DS, Alice can make Bob forward the received message and its signatur

e. By doing

this, Alice can prove that she wasn’t the sender of the message if the signature doesn’t match.

→ With MAC, Alice can’t prove that she didn’t send the message. Therefore, Bob can claim

that Alice was the sender of the message.

→ If the receiv

er claims that a message was sent by a sender only using DS, the sender can

prove that they didn’t send the message. This isn’t possible when using MAC.

→ DS: Alice has to force Bob to prove his claim by sending her copy if the message in the

question wit

h the signature. Then, Alice can show that message and signature can be verified

with Bob’s public key. Bob has to have generated the message.

→ MAC: No, Bob can claim that Alice generated this message

Daniel Garcia

ECE 548

Homework 3

3.7 Because of the known risks of the UNIX password system, the SunOS-4.0 documentation recommends that the password file be removed and replaced with a publicly readable file called /etc/publickey. An entry in the file for user A consists of a user’s identifier *IDA*, the user’s public key, *PUa*, and the corresponding private key *PRa*. This private key is encrypted using DES with a key derived from the user’s login password *Pa*. When A logs in, the system decrypts E(*Pa*, *PRa*) to obtain *PRa*.

1. The system then verifies that *Pa* was correctly supplied. How?
2. How can an opponent attack this system?

**Answer**

a. Public and private keys have an important relationship. When something is encrypted with one key, and can be decrypted by the other. If for example, the decrypted public key was used to encrypt the ID of the user and then the ciphertetxt was decrypted by the public key, then the resulting plaintext would be the ID of the user if and only if the public key was decrypted properly by the correct password.

b. The biggest flaw in the system is the use of the DES algorithm. Though fast, the DES algorithm is very simple to decrypt and in today’s security landscape is considered very vulnerable. In addition, by cracking DES, not only does the attacker gain access to any communications that the user makes, but additionally obtains the password to the account which can be used to escalate privileges and even possibly gain access to the root account in the system.

3.8 It was stated that the inclusion of the salt in the UNIX password scheme increases the difficulty of guessing by a factor of 4096. But the salt is stored in the plaintext in the same entry as the corresponding ciphertext password. Therefore, those two characters are known to the attacker and need not be guessed. Why is it asserted that the salt increases security?

**Answer**

Yes. Added a salt to a password even in plaintext in the same file still adds extra security that does not need to be protected. The reason for this is that often times attackers will hash common passwords and look them up in a database. If there is no way for common passwords to be found, the attacker must resort to cracking the hashes manually. Even with the salts, attackers still need to guess the entire password correctly and waste time hashing the data in order to guess it.

Daniel Garcia

ECE 548

Homework 4

3.10 Consider the Bloom filter discussed in Section 3.3 Token-Based Authentication.

Define *k* = number of hash functions;

*N* = number of bits in hash table; and

*D* = number of words in dictionary.

1. Show that the expected number of bits in the hash table that are equal to zero is expressed as

=

**Answer**

The probability that for any one position in the hash table, the value is 1 is equal to k/N. This is because there are k hash functions which have an equal probability of setting a value to 1 in the set of N bits. Therefore, the probability of any value being 0 is (1 - k/N). Finally, to find the expected value of the number of zeros in the table, each probability is multiplied to itself D times, thus: =

1. Show that the probability that an input word, not in the dictionary, will be falsely accepted as being in the dictionary is

*P* =

**Answer**

Given that is the probability of bits equal to zero in the table, (1 - ) is the probability that any value is equal to one. Given any input word, the probability that the word will falsely be accepted as in the dictionary is

1. Show that the preceding expression can be approximated as

*P* ≈

**Answer**

Given the following equation from part b

*P* =

And the following equation from part a

=

Therefore

3.11 For the biometric authentication protocols illustrated in Figure 3.12, note that the biometric capture device is authenticated in the case of a static biometric but not authenticated for a dynamic biometric. Explain why authentication is useful in the case of a stable biometric but not needed in the case of dynamic biometric.

**Answer**

In the case of the stable biometric, it is important to identify which biometric reading the resulting template represents. This allows the server to authenticate the biometric device in case the client attempts to authenticate using a different device. This also solves the issue of many devices generating the same biometric template for different users. In the case of dynamic biometric authentication, since a challenge is used as well as a biometric reading, the user must know the answer. If the user knows the answer, and generates the correct biometric signal, then the user must be who they say they are and is therefore authenticated. The device must therefore be correct since the challenge is answered.

Daniel Garcia

ECE 548

Homework 5

4.5 UNIX treats file directories in the same fashion as files; that is, both are defined by the same type of data structure, called an **inode**. As with files, directories include a nine-bit protection string. If care is not taken, this can create access control problems. For example, consider a file with protection mode 644 (octal) contained in a directory with protection mode 730. How might the file be compromised in this case?

**Answer:**

In the Unix filesystem, directory permissions are very important. Since members of the group can access the directory and can modify the directory, even though they cannot list the directory, it means that if a member of the group knows the name of the file, that person can also remove it because removing a file requires permission to write to the directory. The file permissions in this case do not really matter because the directory grants the owner and group enough access to the file that the file can be compromised.

In addition, a member of the owners group can remove the file if they know the name. They can also read the file if they know its name. These two facts make it very clear that the file is not being protected by the 644 ACL the way it was intended to be

4.9 The NIST RBAC standard defines a limited role hierarchy as one in which a role may have one or more immediate ascendants but is restricted to a single immediate descendant. What inheritance relationships in Figure 4.10 (Simple ABAC Scenario) are prohibited by the NIST standard for a limited role hierarchy?

**Answer**

Since the NIST hierarchy defines having only one descendant explicitly, that means the in the graph shown on page 132, directory and the project lead 1 & 2 roles, are invalid. This is they have more than one descendant. In order to restructure the graph to abide by NIST, the biggest change would be to put the production engineer and the quality engineers on different levels in the hierarchy, which would challenge those in charge of the hierarchy to decide which privileges are needed. If the same privileges are needed for both jobs, then a more generic name should be given to the same role to prevent the RBAC from snowballing and creating a laundry list of permissions many of which are duplicates of others.