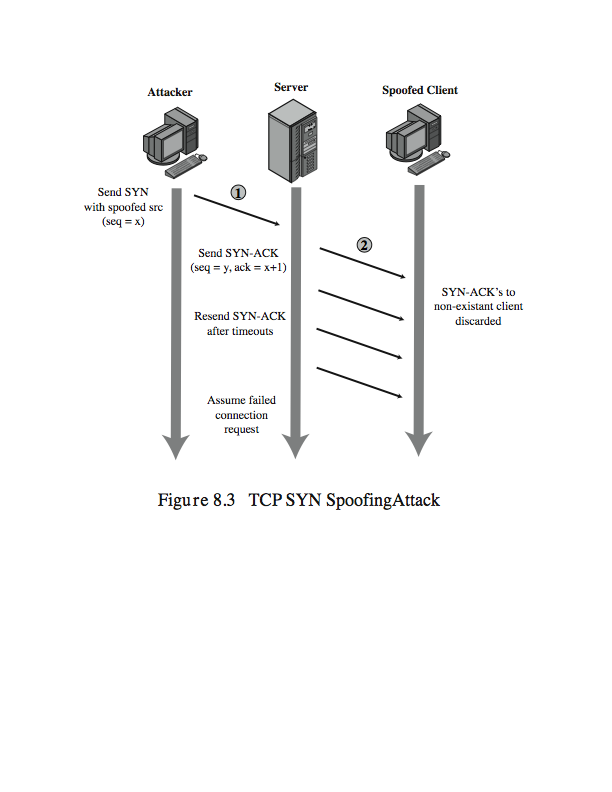
**Homework 04**

1. (6 pts) Explain how TCP SYN flooding attacks work? (with a figure)



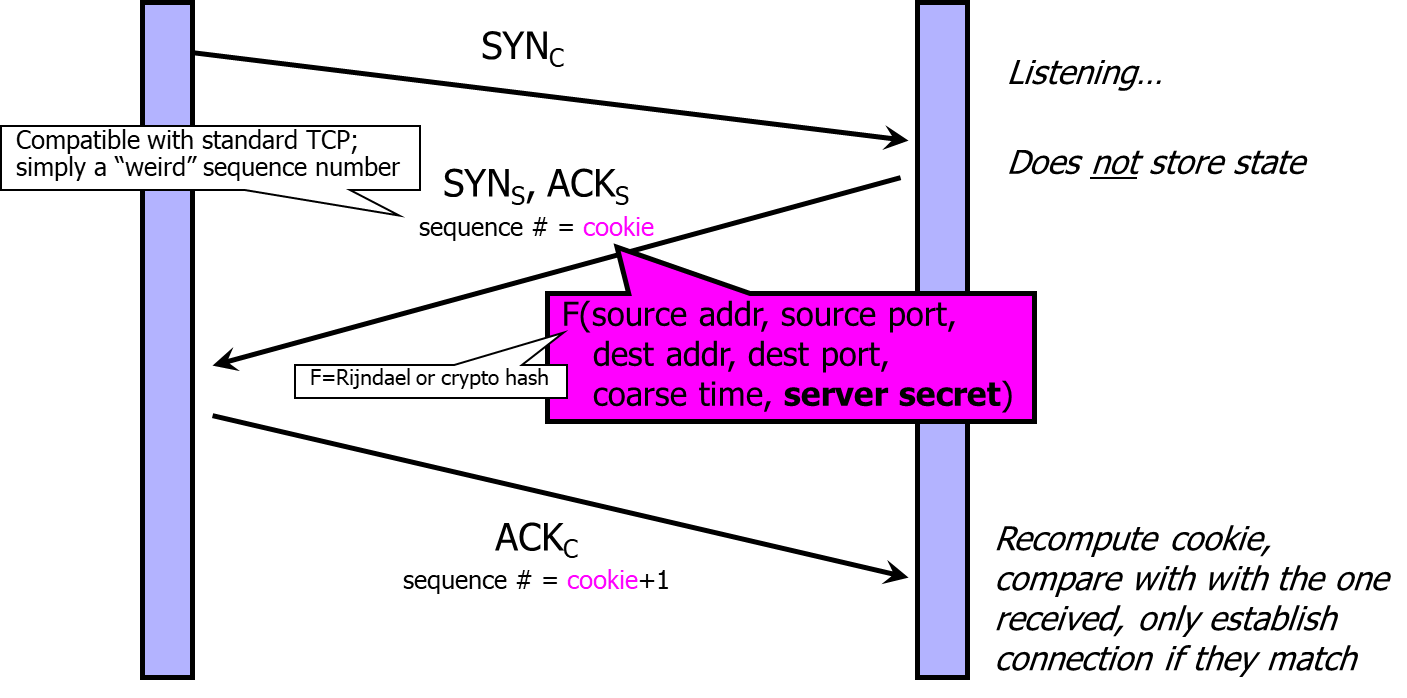
1. (6 pts) Describe how reflection attacks and DNS amplification DoS attacks work

（you may read this article to get a deeper understanding of reflection DDoS attacks： https://www.akamai.com/us/en/about/news/press/2015-press/akamai-warns-of-3-new-reflection-ddos-attack-vectors.jsp）

**Reflection attacks:** attackers send lots of spoofed packets while using the victim host’s IP address as the source IP address to reflectors (e.g., web servers, DNS servers, NTP servers). The reflectors, without knowing the attack respond to the victim, potentially causing the victim denial-of-service. This attack hides the real attack source IP addresses.

**DNS amplification attacks:** it is a type of reflection attack which uses DNS servers as the reflectors. The attacker sends small size DNS request packets to the DNS server, which responds to the victim machine with DNS response packets. Each of the DNS response packet is 50x larger than the request packet, hence achieving the effect of amplification.

1. (5pts) Explain what is SYN Cookie and what it is for? You may use a figure to help illustrate.



SYN Cookies ensure that the responder is stateless until initiator produced at least two messages

* + Responder’s state (IP addresses and ports of the connection) is stored in a cookie and sent to initiator
  + After initiator responds, cookie is regenerated and compared with the cookie returned by the initiator

Here figure is not required for grading.

1. (5 pts) What is DNS spoofing attack?

In a DNS request, there is a 16-bit transaction id (TXID) as a weak authentication of DNS response, which must carry the same TXID. When a client issues a DNS request (i.e., query the IP address given a host name), an attacker may forge DNS responses (which offers the IP address of the host under the control of the attacker and probably doing phishing or malicious services) with guessed TXID to deceive the client.

1. (6 pts) What is proof of work (POW) and why is it deployed in bitcoin?

A **Proof-of-Work** (**PoW**) system (or protocol, or function) is a consensus mechanism. It deters denial of service attacks and other service abuses such as spam on a network by requiring some **work** from the service requester, usually meaning processing time by a computer. (if here you say something else directly regarding blockchains, it is also fine).

In Bitcoin: In Blockchain, PoW is used to confirm transactions and produce new blocks to the blockchain. With PoW, miners (who contribute computing resources to solve cryptography hash puzzles) compete against each other to complete transactions on the network and get rewarded.

1. (6 pts)Assume that passwords are selected from four-character combinations of 26 alphabetic characters. Assume that an adversary is able to attempt passwords at a rate of one per second.
   1. Assuming no feedback to the adversary until each attempt has been completed, what is the expected time to discover the correct password?
   2. Assuming feedback to the adversary flagging an error as each incorrect character is entered, what is the expected time to discover the correct password?

>>>a. there are 26^4 total passwords. On average, the adversary has to try half of them to succeed, so the expected time will be 26^4/2 =…

b. in this case every character may be guessed correctly in 26/2 = 13 times. So totally 13x4=52 times guessing. The time needed is 52 seconds.

1. (8pts) 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 , the user’s public key , and the corresponding private key . This private key is encrypted using DES with a key derived from the user’s login password . When A logs in, the system decrypts E(PRa, Pa) to obtain PRa.



1. The system then verifies that was correctly supplied. How?



1. How can an opponent attack this system?



1. (8 pts) Problem 7.26.

a. You would expect to find 105 \_ 107=1010 = 100 false matches.

b. The probability is only 107=1010 = 1=1000.

1. (9pts) Problem 7.37 .

a. We find d(A;B) = 29=64 = 0:453125, d(A;C) = 39=64 = 0:609375, and d(B;C) =

34=64 = 0:531250.

b. User W is Alice (distance 0.156250), user X is Bob (distance 0.156250), and user

U is Charlie (distance 0.171875). All other distances are near 0.5, so V and Y are

none of the above.

1. (9pts) Construct an access matrix for the following case. There are three users (Cartman, Butters, and Public) that own the files (o1, o2, and o3, respectively). An owner can read and write his own file. Cartman and Butters do not want Public or each other to read anything that they write, whereas Public allows everyone to read his file.
2. Draw the access matrix. Fill in the access matrix with the maximal number of read permissions possible for the three files.
3. Suppose we want to implement this model using (1) an ACL, (2) a capability list (C-list). Show what the lists look like.
4. List two advantages of ACLs over C-lists and two advantages of C-lists over ACL.

**》》》**

1. Answer: (1) Fill the access matrix with the maximal number of read permissions possible for the three files.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *O1* | *O2* | *O3* |
| *Cartman* | *R* |  | *R* |
| *Butters* |  | *R* | *R* |
| *Public* |  |  | *R* |

Note that it is okay (no points taken off) to show all the permissions in the table, but this question asks about read permission only.

**b.**

ACL:

*O1 O2 O3*

*Cartman: R Butters: R Cartman: R*

*Cartman: W Butters: W Butters : R*

*Cartman: Own Butters: Own Public: R*

*Public: W*

*Public: Own*

*C-list:*

*Cartman: O1/R, O1/W, O1/Own, O3/R*

*Butters: O2/R, O2/W, O2/Own, O3/R*

*Public: O3/R, O3/W, O3/Own.*

**c.**

*ACL's provide for superior access review/revocation on a per-object basis*

*C-list provides for superior access review/revocation facilities on a per-subject basis*

1. (4 pts) Assume a system with job positions (i.e., roles). For job position , the number of individual users in that position is and the number of permissions required for the job position is .



1. For a traditional DAC scheme, how many relationships between users and permissions must be defined?
2. For a RBAC scheme, how many relationships between users and permissions must be defined?

>>>Answer:

* 1. In DAC, you can draw N tables (each table corresponds to one job position), where rows stand for users and columns stand for permissions. Each cell in a table represents a relationship between a user and a permission. The question is: totally how many cells do we need? The answer is



* 1. In RBAC, you can draw two tables. The first table T1 shows the relationships between users (rows) and job positions (columns), and the second table T2 shows that between job positions(rows) and permissions (columns). Now for T1, for each job position i, Ui users are defined, which mean Ui cells are not empty. Hence, the total number of relationships (i.e., non-empty cells) will be Sum (Ui) for i from 1 to N from T1. Similarly, for T2, one can derive the total number of non-empty cells are Sum (Pi) for i from 1 to N. Combining both tables, we get the result



1. (8 pts) The high water mark principle and low water mark principle both apply in the realm of multilevel security.
   1. What is MLS?
   2. Define the high water mark principle and the low water marker principle in the context of MLS.
   3. Is BLP consistent with a high water mark principle or a low water mark principle, both, or neither? Justify your answer.
   4. Is Biba’s Model consistent with a high water mark principle or a low water mark principle, both, or neither? Justify your answer.

MLS: an access control system where both system resources and subjects are assigned certain security levels, and subjects are denied/approved the access based on the security levels. (you can also use the definition in slide #27)



1. textbook 11.6 (6pts)

a. If the canary value is overwritten, then there is likely a problem with the return

address.

b. The Microsoft implementation allowed the user to define a handler function to be

called when the canary died. It was claimed that the handler function could be

specified by the attacker. This would make all buffer overflows exploitable, even

those that were not exploitable when the canary was not used.

1. 11.15 (8pts).

a. The results are buf2 = 22222222 and buf2 = 11122222.

b. Apparently, bu\_er 1 has overwritten the start of bu\_er 2.

c. Trudy might be able to overwrite some important data to, say, change a failed

authentication into a successful authentication.

1. 11.16 (6 pts)

a. If len is negative there is a problem. The test in the if will be passed, but then

memcpy assumes that len is unsigned. So, a negative value for len is interpreted

as a very large number, which would lead to a buffer overflow.

b. Explain how an integer overflow might be exploited by Trudy. Trudy can overflow

an array and cause problems.