Daniel Willard

Winter 2024

433 Network Security

1/29/2024

1. **Security concepts.** Textbook, Section 1.8 Exercises, Problem 12. (3 points) Preserving confidentiality, integrity, and availability of data is a restatement of the concern over the three harms of interception, modification, and interruption. How do the first three concepts relate to the second three? That is, is any of the three harms equivalent to one or more of the first three concepts? Is one of the three concepts encompassed by one or more of the three harms??
    a. Interception is a breach of confidentiality.
    b. Modification is a breach of integrity.
    c. Interpretation is a breach of availability.
    d. Each harm directly affects one of the principles of security.
2. **Secure voting.** Textbook, Section 1.8 Exercises, Problem 22. (3 points) Consider a program to accept and tabulate votes in an election. Who might want to attack the program? What types of harm might they want to cause? What kinds of vulnerabilities might they exploit to cause harm?
    a. Attackers: political opponents, hacktivists, hackers, nation-states, foreign nationals, criminal groups, corporations, disgruntled insiders, ect.
    b. They could cause harm by disclosing confidential information, they could undermine public trust, they could delay the count or election. They could change the data and elect a different candidate.
    c. They might exploit software and network vulnerabilities, they could use an insider threat, they could attack the physical system at the vote machine and other physical security weaknesses, and they could attack the supply chain need to run the machine like the power grid. There are many other options.
3. **Smart lock.** Design a "smart" lock for your home that does a 3-factor authentication that checks something the user knows AND something the user has AND something the user is. (3 points)
    a. User must enter a pin to access the key hole.
    b. The user must use their unique key to access the user id authentication.
    c. The look the scans the user’s eye to compare and valid the user before entry is allowed
4. **Authentication that learns.** Textbook, Section 2.5 Exercises, Problem 11. (5 points) 11.Outline the design of an authentication scheme that “learns.” The authentication scheme would start with certain primitive information about a user, such as name and password. As the use of the computing system continued, the authentication system would gather such information as commonly used programming languages; dates,

times, and lengths of computing sessions; and use of distinctive resources. The authentication challenges would become more individualized as the system learned more information about the user. Your design should include a list of many pieces of information about a user that the system could collect. It is permissible for the system to ask an authenticated user for certain additional information, such as a favorite book, to use in subsequent challenges. Your design should also consider the problem of presenting and validating these challenges: Does the would-be user answer a true-false or a multiple-choice question? Does the system interpret natural language prose?

a. Initial data collection: system starts with collecting basic information (user name, password, 12 security question, email, phone number, device being used to access system. Location)
b. Ongoing Data Accumulation: as the user interacts with the system it collects additional and ongoing data: (device being used to access, location, ip address, prompt user to answer one of the 12 security question and if answered wrong prompt user to change the answer on login, log common used programs and programming languages use, average session time, frequently accessed resources, patterns in typing, interactions with other users, preferences in settings, answers to previous login challenges)
c. User Provided information: periodically the system will ask the user to provide data.(prompt user to answer one of the 12 security question and if answered wrong prompt user to change the answer on login, recent significant life events, hobbies and interests)
d. Challenge design: Multiple choice with one random question needing to be typed out as another option to prevent guessing correct in. Captcha that incorporates user specific knowledge, then behavioral biometrics and patterns will be monitored and flag admin if there is a large enough change once log in.
e. Natural Language Processing: while this allows a user friendly experience it introduces many vulnerabilities so it would not be implemented. If it was it would be used to interpret simple structed responses to minimize risk of misinterpretation or manipulation.
f. Validation: system would validate responses securely and accurately. (security question are not easy guessable or researchable, implement a number of failed attempts before locking and escalating the account, use encryption and secure channels to protect the collected data.)
g. Privacy Considerations: the system must ensure privacy of the use so the data would not be tied to a token instead of a name for monitoring proposes. And stored securely.
h. Learning: the system would have some kind of algorithm to analyze the collected data and adapt authentication accordingly like <u>user never capitalized the letter ‘I’.</u>
i. Summary: so the system would learn form data collection both over time and initially, then the changes would become more personal over time to prevent system breaches. The system must balance security and convenience and security and privacy. Like using multiply choice but having one question where

the user must provide ethe correct answer to. This also is better then t/f question or reeling on a NLP that could be modified or easily guessed. The validation must improve with the data collect but prioritize privacy to improve the authentication process

5. **Synchronous password.** Textbook, Section 2.5 Exercises, Problem 17. (3 points) A synchronous password token has to operate at the same pace as the receiver. That is, the token has to advance to the next random number at the same time the receiver advances. Because of clock imprecision, the two units will not always be perfectly together; for example, the token's clock might run 1 second per day slower than the receiver's. Over time, the accumulated difference can be significant. Suggest a means by which the receiver can detect and compensate for clock drift on the part of the token.
    a. Timestamp Exchange: receiver periodically requests timestamps from the token
    b. Calculation of drift: By comparing the timestamp received from the token with the receiver's local clock time when it was received, the receiver can calculate the drift rate between the two clocks.
    c. Adjustment Algorithm: Using the drift rate, the receiver can adjust its expectations for when the token's next output should arrive
    d. Feedback mechanism: The receiver can provide feedback to the token about the observed drift, allowing the token to adjust its clock rate accordingly
    e. Periodic Reevaluation: Periodically, the receiver can reevaluate the drift rate to account for any changes over time.
6. **Access control mechanisms.** Textbook, Section 2.5 Exercises, Problem 1. (4 points) Describe each of the following four kinds of access control mechanisms in terms of (a) ease of determining authorized access during execution, (b) ease of adding access for a new subject, (c) ease of deleting access by a subject, and (d) ease of creating a new object to which all subjects by default have access. per-subject access control list (that is, one list for each subject tells all the objects to which that subject has access) per-object access control list (that is, one list for each object tells all the subjects who have access to that object) access control matrix capability.
    a. per-subject access control list (that is, one list for each subject tells all the objects to which that subject has access):
        i. Ease of determining authorized access during execution (a): relatively easy to determine authorized access during execution because each subject has its own list specifying which objects it can access. simply consulting the list associated with the subject provides clear information about authorized access.
        ii. Ease of adding access for a new subject (b): new subject involves creating a new list for that subject and specifying the objects to which it has access. This process can be straightforward since it only requires modifying the ACL associated with the new subject.
        iii. Ease of deleting access by a subject (c): Deleting access for a subject entail removing or modifying entries in the subject's ACL. This action can be done relatively easily by editing the relevant ACL.

    iv. Ease of creating a new object with default access (d): Creating a new object with default access for all subjects may require updating each subject's ACL to include the new object. This process can be manageable but may become cumbersome as the number of subjects increases.

b. per-object access control list (that is, one list for each object tells all the subjects who have access to that object):

    i. Ease of determining authorized access during execution (a): it's relatively easy to determine authorized access during execution by consulting the ACL associated with the object being accessed.

    ii. Ease of adding access for a new subject (b): Adding access for a new subject involves updating the ACL associated with the object by adding the new subject to the list of authorized users. This process is straightforward and only requires modifying the ACL of the object in question.

    iii. Ease of deleting access by a subject (c): Deleting access for a subject requires removing or modifying entries in the ACL associated with the object. This action can be done relatively easily by editing the ACL of the object.

    iv. Ease of creating a new object with default access (d):): Creating a new object with default access for all subjects may involve creating a new ACL for the object and specifying default access permissions. This action can be done relatively easily by editing the ACL of the object.

c. access control matrix:

    i. Ease of determining authorized access during execution (a): he access control matrix provides a comprehensive overview of all access permissions between subjects and objects, making it relatively easy to determine authorized access during execution. However, searching the matrix for specific permissions may require additional computational effort. A sparse matrix may need to be used.

    ii. Ease of adding access for a new subject (b): Adding access for a new subject involves adding a row to the matrix corresponding to the new subject and specifying the access permissions for that subject across all objects. This process can be straightforward but may require updating multiple entries in the matrix

    iii. Ease of deleting access by a subject (c): Deleting access for a subject requires removing the corresponding row from the matrix. This action can be relatively simple, but it may also necessitate updating multiple entries in the matrix.

    iv. Ease of creating a new object with default access (d): Creating a new object with default access for all subjects may involve adding a new column to the matrix and specifying default access permissions for all subjects. This process can be manageable but may require updating multiple entries in the matrix. All of these sum up make the matrix very reliable but also the most computationally heavy and time consuming to modify

d. capability:
    i. Ease of determining authorized access during execution (a): Determining authorized access during execution with capabilities involves presenting the appropriate capability (token) associated with the desired object. This process can be straightforward, as possession of the capability inherently grants access.
    ii. Ease of adding access for a new subject (b): Adding access for a new subject involves issuing a new capability (token) granting access to the desired object. This process can be relatively easy, as it only requires creating and distributing the capability to the new subject.
    iii. Ease of deleting access by a subject (c): Deleting access for a subject involves revoking or invalidating the corresponding capability associated with that subject. This action can be done relatively easily by removing or invalidating the capability. Though token management.
    iv. Ease of creating a new object with default access (d): Creating a new object with default access for all subjects may involve generating a capability that grants access to the new object and distributing it to all subjects. This process can be straightforward and efficient, as capabilities can be created and distributed as needed.

7. Traffic access control. **You are charged to protect the machine of Alice from receiving unwanted traffic from machines on a blacklist. (6 points)**

a. (1) If the access control is to be enforced by the machine of Alice, will the implementation of access control be an ACL, or a directory? Explain briefly.
    i. the implementation of access control would likely be an ACL (Access Control List). An ACL allows Alice's machine to specify which machines or IP addresses are allowed or denied access to her machine. It provides a straightforward and efficient way to manage access control directly on the machine itself. A directory-based approach, which typically involves storing access control information in a centralized directory service, might introduce unnecessary complexity and overhead for managing access control on a single machine.

b. (2) If the access control is to be enforced by a firewall machine right in front of each sender, will the implementation of access control be an ACL, or a directory? Explain briefly.
    i. firewall acts as a barrier between the sender and Alice's machine, filtering traffic based on predefined rules in the ACL. This setup enables centralized control and management of access control policies at the network perimeter. A directory-based approach would be less suitable in this scenario because it may not provide the level of control and granularity needed to effectively filter traffic at the network perimeter.

c. (3) If the access control is to be enforced by routers *en route* from every sender to the machine of Alice, which will inspect the traffic to either block or pass the traffic, what will be the access control implementation you would choose? Explain briefly.

i. I would opt for an Access Control List (ACL) implementation. Theis would allow each router along the communication path would maintain a list of allowed and denied IP addresses or network ranges. As packets traverse these routers, they would be inspected against the ACL to determine whether they should be forwarded to Alice's machine or dropped. This approach offers more granular control over traffic flow compared to a directory-based implementation and aligns well with the router's capabilities to filter traffic based on specific criteria. Additionally, ACLs are commonly used in network devices for access control purposes, making them a suitable choice for enforcing access control at the router level.

8. **AES lifetime.** Textbook, Section 2.5 Exercises, Problem 21. (2 points) If the useful life of DES was about 20 years (1977–1999), how long do you predict the useful life of AES will be? Justify your answer.

   a. My prediction is that 20 year about with computation advancement. But based on the security strength and Adoption and standardization of AES it is possible that it could possibly go beyond 20 years be that depends on technology advancement. Much like des I believe it will remain relevant for longer but I doubt it will be standard best practical much after 20 years.

9. **RSA.** With n=10, show the procedure how you may choose a corresponding public and private key, and verify it works for a plain text P=2. (3 points)

   a. two distinct prime numbers, p, and q. p*q=n tus, p=2 n=5

   b. (p-1)*(q-1) = (2-1)(5-1) = 4

   c. Choose e and d. where 1<e<4 we will choose 3.

   d. For d we need a e*d mod 4 = 1 where d is a prime thus equals 11 since 3 is already use the next prime is 11

   e. Encryption : C= P^e mond n = 2^3mod10 = 8

   f. Decryption: P` = C^dmond n = P` = 8^11 mod 10 = 2

   g. P =P` proven procedure

10. **Diffie-Hellman key exchange.** Describe how Diffie-Hellman key exchange protocol works, and why it is resilient against eavesdroppers. (4 points)

    a. **Initialization**:

       i. P and G prime number are chosen and exchanged securely person to person

       ii. Then person picks a private key. A and B for person 1(p1 and person 2 (p2)

    b. Key Exchange:

       i. P1 computes A=g^a mod p and sends to p2

       ii. P2 computes B=g^b mod p and send to p1

    c. Secret key Caluations:

       i. P1 K =B^a mod p

       ii. P2 K =A^b mod p

d. Summary: The Diffie-Hellman key exchange protocol is resilient against eavesdroppers due to the computational difficulty of solving the discrete logarithm problem. If A of B values where intercepted it is computationally feasible to determine K without a and b values of the secret key. But it is computational expensive and harder for larger numbers thus making it resilient not proofed.

11. **Alice and her friends.** Alice has 50 friends, and she wishes to send all of them a common message secretly. That is, each friend will receive the same message, and only these 50 friends can decrypt it. Alice knows the public key of each of them. How would you design the message for Alice? (4 points)
    a. Generate a symmetric key
    b. Encrypt the Message with the Symmetric Key
    c. Encrypt the Symmetric Key with Each Friend's Public Key
    d. Compose the Final Message: The final message composed by Alice consists of: The encrypted message CC (encrypted using symmetric encryption). 50 copies of the symmetric key K, each encrypted with the public key of one of Alice's friends and tag with each friends name.
    e. Send the Final Message to Each Friend: all they have to do is decrypt the key with they name marker and then use the key to decrypt the message