Table 11-2 Common sequence of password attack tools

Order	Password attack	Explanation
1	Custom wordlist	Download a stolen password collection
2	Custom wordlist using rule attack	Generate password statistics using a rule attack to create specialized masks
3	Dictionary attack	Perform a dictionary attack on passwords
4	Dictionary attack using rules	Conduct a refined dictionary attack using results from a rule attack
5	Updated custom wordlist using rules	Input any cracked passwords from previous steps to create more refined rules
6	Hybrid attack	Perform a focused dictionary attack with a mask attack
7	Mask attack	Conduct a mask attack on harder passwords that have not already been cracked
8	Brute force attack	Last resort effort on any remaining passwords

Table 12-1 Basic steps in access control

Action	Description	Scenario example	Computer process
Identification	Review of credentials	Delivery person shows employee badge	User enters user name
Authentication	Validate credentials as genuine	Gabe reads badge to determine it is real	User provides password
Authorization	Permission granted for admittance	Gabe opens door to allow delivery person in	User authorized to log in
Access	Right given to access specific resources	Delivery person can only retrieve box by door	User allowed to access only specific data
Accounting	Record of user actions	Gabe signs to confirm the package was picked up	Information recorded in log file

Table 12-3 Access control models

Name	Explanation	Description
Mandatory Access Control (MAC)	End user cannot set controls	Most restrictive model
Discretionary Access Control (DAC)	Subject has total control over objects	Least restrictive model
Role-Based Access Control (RBAC)	Assigns permissions to particular roles in the organization and then users are assigned to roles	Considered a more "real-world" approach
Rule-Based Access Control	Dynamically assigns roles to subjects based on a set of rules defined by a custodian	Used for managing user access to one or more systems
Attribute-Based Access Control (ABAC)	Uses policies that can combine attributes	Most flexible model

⇨ Vulnerability assessment is a systematic and methodical evaluation of the security posture of the enterprise.

Table 13-3 Vulnerability assessment actions and steps

Vulnerability assessment action	Steps
1. Asset identification	a. Inventory the assets b. Determine the assets' relative value
2. Threat identification	a. Classify threats by category b. Design attack tree
3. Vulnerability appraisal	a. Determine current weaknesses in protecting assets b. Use vulnerability assessment tools
4. Risk assessment	a. Estimate impact of vulnerability on organization b. Calculate risk likelihood and impact of the risk
5. Risk mitigation	a. Decide what to do with the risk

Vulnerability Assessment Tools

Many tools are available to perform vulnerability assessments. These include port scanners, protocol analyzers, vulnerability scanners, honeypots and honeynets, banner grabbing tools, crackers, command line tools, and other tools.

- ⇨ basic security properties (5)

• availability: the ability of a system to ensure that an asset can be used by any authorized parties

• integrity: the ability of a system to ensure that an asset is modified only by authorized parties

• confidentiality: the ability of a system to ensure that an asset is viewed only by authorized parties

• authentication: the ability of a system to confirm the identity of a sender

• nonrepudiation or accountability: the ability of a system to confirm that a sender cannot convincingly deny having sent something

- ⇨ Authentication mechanisms use any of three qualities to confirm a user's identity:

• Something the user knows. Passwords, PIN numbers, passphrases, a secret handshake, and mother's maiden name are examples of what a user may know.

• Something the user is. These authenticators, called biometrics, are based on a physical characteristic of the user, such as a fingerprint, the pattern of a person's voice, or a face (picture). These authentication methods are old (we recognize friends in person by their faces or on a telephone by their voices) but are just starting to be used in computer authentications.

• Something the user has. Identity badges, physical keys, a driver's license, or a uniform are common examples of things people have that make them recognizable.

NB: Authentication is based on something
you know, are, or have.

- ⇨ Attacking and Protecting Passwords

Dictionary Attacks

Inferring Passwords Likely for a User

Guessing Probable Passwords

Defeating Concealment

Exhaustive Attack (exhaustive or brute force)

- ⇨ **Good Passwords || These properties make the password difficult (but, of course, not impossible) to determine.**

Use characters other than just a–z.

Choose long passwords.

Avoid actual names or words.

Use a string you can remember.

Use variants for multiple passwords.

Change the password regularly.

Don't write it down.

Don't tell anyone else. \\ **********social engineering

8. difference betwn.

SYMETRIC CRYPTOGRAPHY	ASYMETRIC CRYPTOGRAPH
Data is encrypted and decrypted by the same secret key	Uses two different keys for encrypt and decrypt data
Secret key is shared between the sender and recipient	The sender should have the public key of the receiver to encrypt content using it and receiver should have the private key of sender to decrypt the contents of data sent
It uses shorter keys (128 or 256 bits) thus it is faster encryption method.	Uses longer keys (2048 bits) that makes it take longer time in encryption
Vulnerable to security risks due to nature of keeping shared secret key	Very secure due to authentication requirement

9. The answer is INDIA, YOU TAKE PROTO – HELLO + 1 IN EACH KEY

10. An 8-block cipher has 28 possible input blocks. Each mapping is a permutation of the 28 input blocks; so there are 28! possible mappings; so there are 28! possible keys.

11. (a) Consider the 3-bit block cipher in Table 8.1 from the text book. Suppose the plaintext is 100100100.

Convert 3-bit (three-three bits) block cipher is 100 100 100.

- Initially assume that CBC is not used.
- Change each three-bit input with the output 3-bit cipher as specified in the table 8.1 from the text book. Then the output for input plaintext 100 is 011.

So, the resulting cipher text is *011 011 011*.

(b) Suppose she knows that a 3-bit block cipher without CBC is being employed (but doesn't know the specific cipher), then guess that the same as the three plaintext block.

c)

Now suppose that CBC is used with IV=

$c(0) = 111.$

$c(1) = K_s(m(1) \oplus c(0))$

$= K_s(100 \oplus 111)$

Here, 100 is ciphertext.

$c(2) = K_s(m(2) \oplus c(1))$

$= K_s(110 \oplus 100)$

Here, 110 is ciphertext

$c(3) = K_s(m(3) \oplus c(2))$

$= K_s(101 \oplus 110)$

Here, 101 is ciphertext

So, the resulting cipher text is **100110101**

12. Callie wants to send the message M= 13 to Alice Assume the two primary numbers chosen by Alice are p=11 and q=3, Also Assume Alice choose e=3 and d=7 Calculate,

(a) Alice's Public key (n;e)

From RSA encryption n=pq = 11x3=33

So public key (n;e) = (33,3)

(b) Alice's Private key (n;d) = (33,7)

(c) Cipher text c sent by Callie to Alice

$C = m^e \bmod n$

$C = 13^3 \bmod 33 = 19$

(d) The value of reminder R when Alice recovers the message

From $m = c^d \bmod n$

$m = 19^7 \bmod 33$, reminder R = 13

13. Virus Vs Worms

A virus is a program that can replicate itself and pass on malicious code to other nonmalicious programs by modifying them.

A worm is a program that spreads copies of itself through a network

Trojan horse: program with benign apparent effect but second, hidden, malicious effect

Virus	**Worm**
viruses must be triggered by the activation of their host	worms are stand-alone malicious programs that can self-replicate and propagate independently as soon as they have breached the system
Require activation form human for example inserting the affected USB drives to computer.	Worms do not require activation—or any human intervention—to execute or spread their code.

Summarizing the differences between viruses and worms

Virus	Worm
• Requires a host • Triggered by human interaction • Often arrives through an infected file or program (file-infector)	• Spreads independently • Doesn't require human interaction • Often arrives through a software vulnerability

Though there can be a scale of danger among viruses and worms, worms are generally considered more dangerous. Worms are sneakier, because they can infect you without you even realizing it.

14.(a) A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the data to the buffer overwrites adjacent memory locations.

For example, a buffer for log-in credentials may be designed to expect username and password inputs of 8 bytes, so if a transaction involves an input of 10 bytes (that is, 2 bytes more than expected), the program may write the excess data past the buffer boundary.

(b) ways harm can be caused from Buffer Overflow

1.Privilege escalation, the attacker regains control from the operating system, possibly with raised privileges, the attacker can gain control by masquerading the operating system, the attacker can execute commands in a powerful role

stack and heap

Overwriting Memory

(c) 5 Countermeasures of Buffer Overflow

- The most obvious countermeasure to overwriting memory is to stay within bounds. Maintaining boundaries is a shared responsibility of the programmer, operating system, compiler, and hardware
- Programming Control
- Language Features
- Code Analysers
- Separation
- Stumbling Blocks

15. A program is written to compute the sum of the integers from 1 to 10. The programmer, well trained in reusability and maintainability, writes the program so that it computes the sum of the numbers from k to n. However, a team of security specialists scrutinizes the code. The team certifies that this program properly sets k to 1 and n to 10; therefore, the program is certified as being properly restricted in that it always operates on precisely the range 1 to 10. List different ways that this program can be sabotaged so that during execution it computes a different sum, for example, 3 to 20.

Answers

a) Someone changes the source code before its complication,

b) Someone patches (i.e.,) the binary object code while it is stored on disk before execution,

c) During execution, an outside process patches the object code.

16.You receive an email message that purports to come from your bank. It asks you to click a link for some reasonable-sounding administrative purpose. How can you verify that the message actually did come from your bank?

Each blank in the country has to have a verification certification on the e-mails they sent, thus I need to verify that the email has this certification, also I need to clarify the domain that come from because it needs to be exactly the same as one from my bank. Antivirus, anti-spyware and firewall are crucial from potential attack.

17.Now play the role of an attacker. How could you intercept the message described in part (16) and convert it to your purposes while still making both the bank and the customer think the message is authentic and trustworthy?

I would send an email to the customer giving him thanks of appreciating for providing the information about the name of the bank and use the significant information to whatever propose I wanted immediately. For the bank to never know the behind scenes on what is going on, I would install a spyware or any other sneaking software on the computer of the person to monitor his activities.

18. Various Countermeasures of malware for developers and programmers