Secure Software Development
Penetration testing

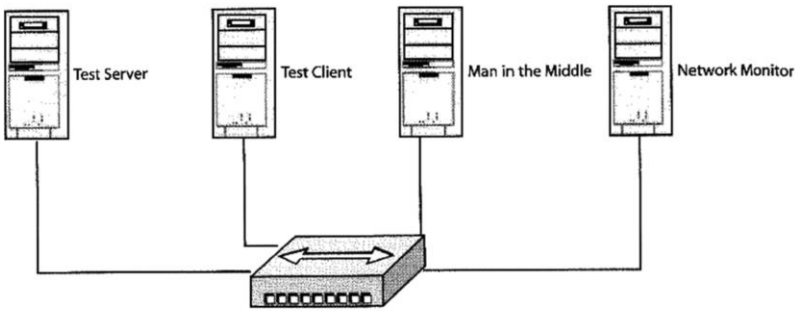
Objectives

- Describe the required environment for security testing
- List required tools for effective security testing
- Explain how passwords are stored in modern LINUX systems.
- Explain the concept of a password salt
- Explain the purpose for the NIKTO software in penetration testing.

Brain Teaser

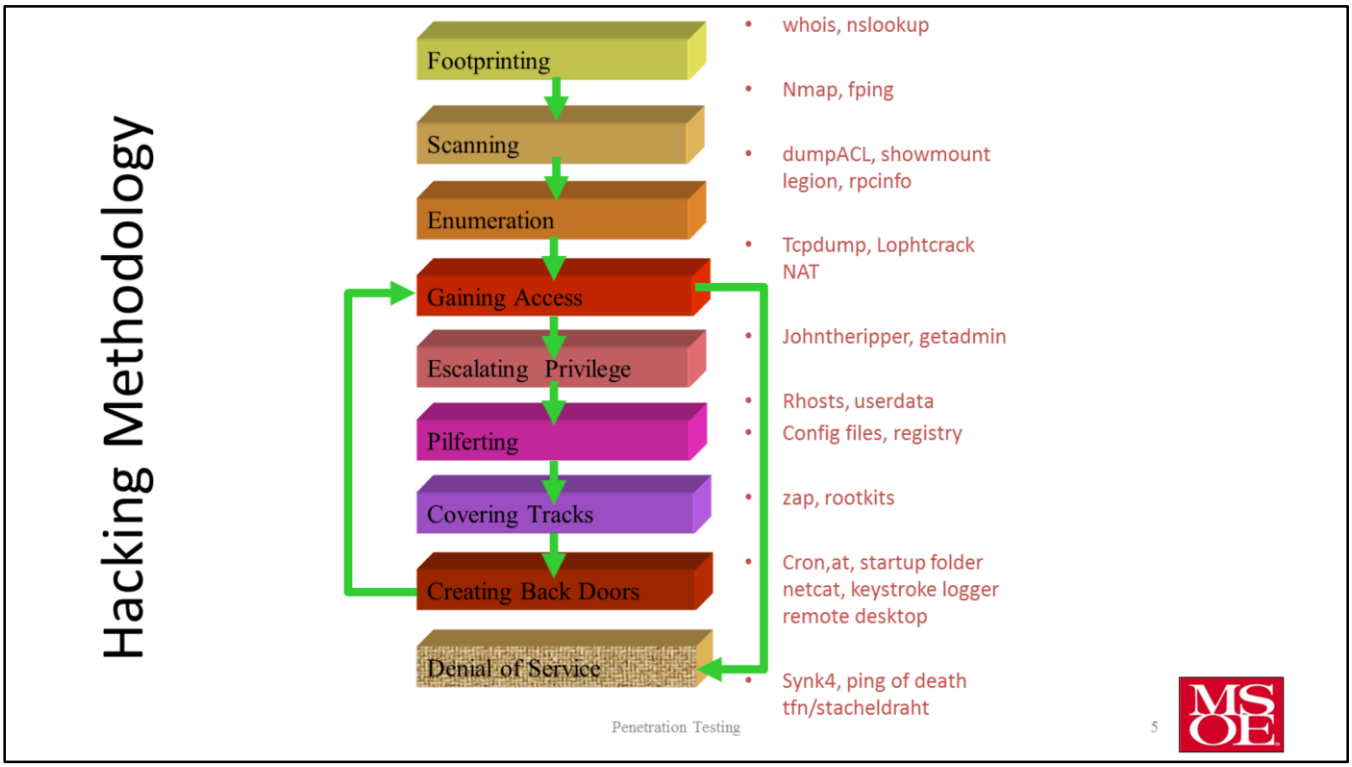
What's wrong with this door?

Testing Environment



What does this have to do with penetration testing

Penetration testing is the equivalent of just doing system level testing



Footprinting

Information gathering.

Note that for penetration tester, this step is to avoiding testing others instead of your client and to include all systems to be tested (sometime the organization will not tell you what their systems consist of).

Scanning

Which machine is up and what ports (services) are open

Focus on most promising avenues of entry.

To avoid being detect, these tools can reduce frequency of packet sending and randomize the ports or IP addresses to be scanned in the sequence.

Enumeration

Identify valid user accounts or poorly protected resource shares.

Most intrusive probing than scanning step.

Gaining Access

Based on the information gathered so far, make an informed attempted to access the target.

Escalating Privilege

If only user-level access was obtained in the last step, seek to gain complete control of the system.

Pilfering

Gather info on identify mechanisms to allow access of trusted systems.

Covering Tracks

Once total ownership of the target is secured, hiding this fact from system administrators become paramount, less they quickly end the romp.

Setting up Backdoor Connection

Once obtain the admin privilege, you install tools that allow you to run command remotely (e.g. netcat) or use the machine as a stepping stone for relaying or redirecting the msg (fpipe)

Port redirection accepts packet from one port and send it over another port. It can be used to avoid packet filter firewall.

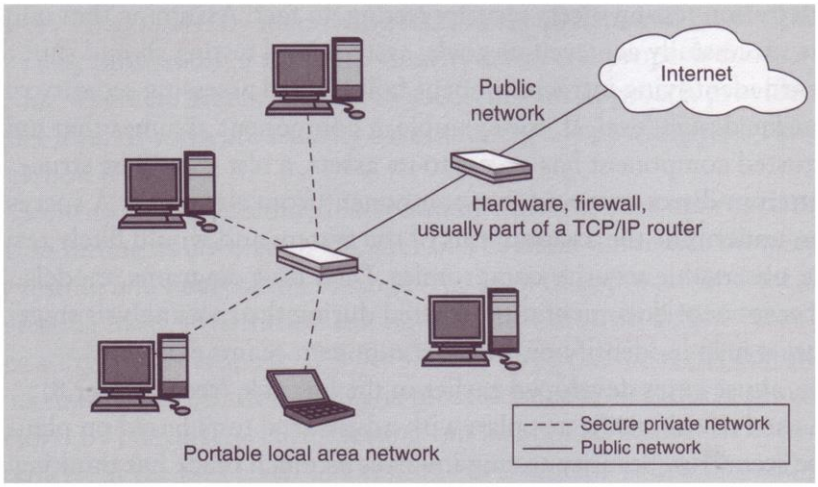
Creating Back Doors

Trap doors will be laid in various parts of the system to ensure that privilege access is easily regained whenever the intruder decides.

Denial of Services

If atacker is unsuccessful in gaining access, they may use readily available exploit code to disable a target as a last resort.

Outside In Approach to security



Traditional approaches to security focus on network infrastructure, firewalls, and port scanning

Classic security approach -> Probe network ports to see which services are listening

-> Apps to do this: Nessus or nmap

Problem: Perimeter only exists at the network / packet level

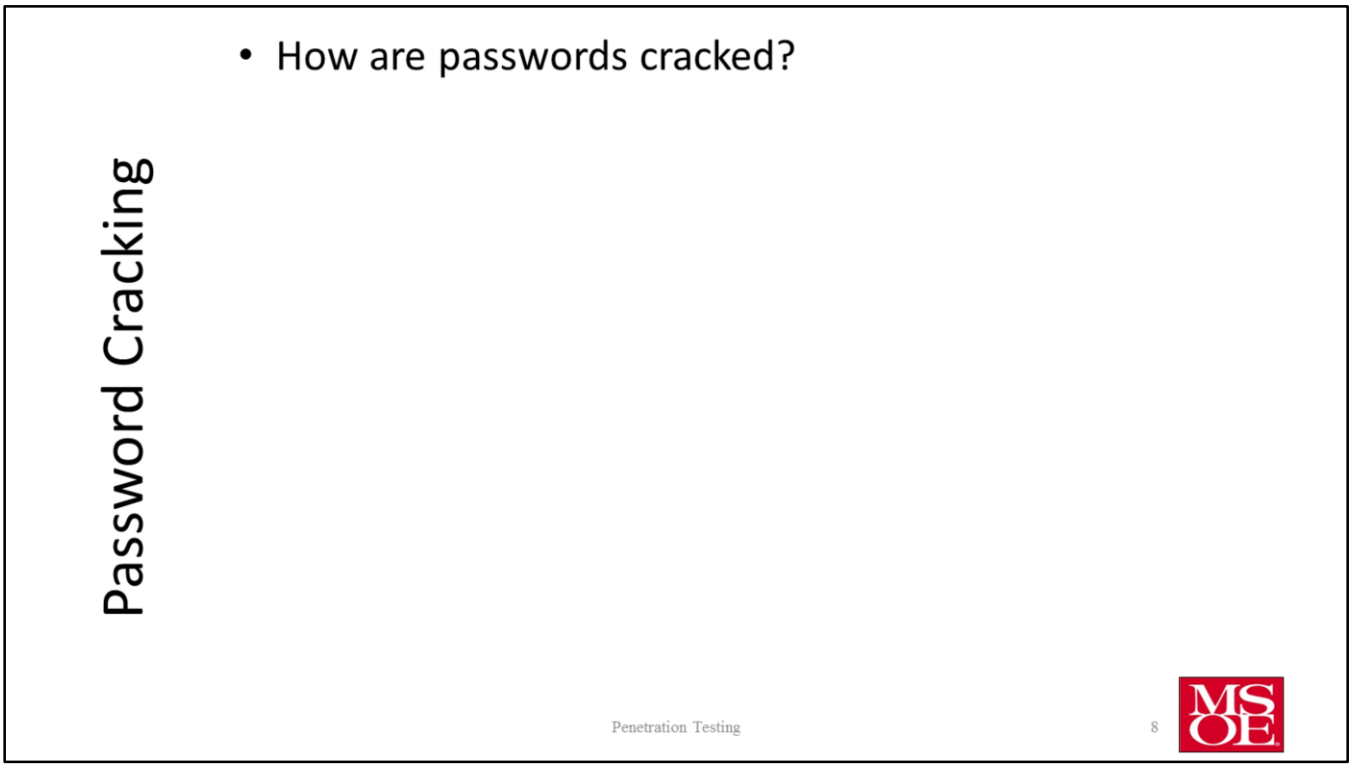
Problems: Tunneling

Example: SOAP *Simple Object Access Protocol* shuffles traffic through port 80

Solution: Security from the inside out

Lets take a look at an example

- Open Source (GPL) web server scanner
- Tests against web servers for multiple items,
  - 6400 potentially dangerous files/CGIs
  - outdated versions of servers
  - version specific problems servers.
  - server configuration items
    - multiple index files
    - HTTP server options
- identify installed web servers and software



In Chapter 7, we described authentication mechanisms including <username, password > authenticators. We also indicated that anything short of one-time passwords was not strong password authentication.

So… how are passwords broken – GUESSING AND CRACKING.

Guessing – Find or guess a user's identifier
Create a list of possible passwords
Try each one
On success you are in, else keep trying

Hampered by unsuccessful login timeout – If (n) attempts are unsuccessful, lock the system for (m) minutes – n & m variable.

Offline Password Cracking

- Most cracking is done off-line to avoid the timeout problem.
- Major steps: Find user ids
  Get encrypted or hashed passwords or password files
  Create a list of trial passwords
  Encrypt or hash the trial passwords
  See if there is a match
- Attacks: Dictionary attacks (build a dictionary of passwords).
  Brute force (try all possible passwords).
  Hybrid attacks (modified dictionary attack using altered dictionary words (party becomes p$art%y).

Password Cracking Starters

- What can we find out up front – commercial systems?
- Format for user id.
- Some user ids (e.g., guest, system, administrator)
- Password minimum/maximum length, legal characters.
- Rules of construction.
- The encryption or hash algorithm.
- Where the password file is stored by default.

/etc/password file format

(Fig.01: /etc/passwd file format - click to enlarge)

1. **Username**: It is used when user logs in. It should be between 1 and 32 characters in length.
2. **Password**: An x character indicates that encrypted password is stored in /etc/shadow file.
3. **User ID (UID)**: Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
4. **Group ID (GID)**: The primary group ID (stored in /etc/group file)
5. **User ID Info**: The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. This field use by finger command.
6. **Home directory**: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
7. **Command/shell**: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. Please note that it does not have to be a shell.

John the Ripper

- Very capable password cracker for Unix systems including S/Key files and Kerberos Ticket Granting Tickets for the Andrew File System.
- Runs cross platform
- Takes a Unix password file as input - etc/passwd or etc/shadow.
- etc/passwd is a user-level public file
- etc/shadow requires root-level access
- Modes:
  - Dictionary (called wordlist) – specify a text file to use as a dictionary.
  - Brute force (called incremental mode) – tries all possible combinations.