Lecture 3 Notes; April 4, 1997; Notetaker: Joel Baumert

PENETRATION TESTING

Announcments:
  Web page up: http://wwwcsif.cs.ucdavis.edu/~cs253/index.html
  Homework will be given on mon.
  Handout on wed.

Penetration testing (aka Tiger Teaming, Red Teaming):
  test security for a system/installation

  usually after a system has been implemented, but could be
  testing design.

  failure doesn't guarantee security.

  When testing a system primarily looking for technical flaws
    on a standalone machine or on a test network.

  When testing a site you could test operations/procedures, site
    physical/system security.

    Procedures operations include: how passwords are assigned,
      how systems are maintained.

  This class will primarily focus on system security.

  success measured based predefined goal.
    Goal based on site policy (what are you testing? when is test
      successful?)

    could be measured in terms of the number of flaws found
      or number of design errors found. It is not only important
      to not only find the flaws, but give some indication what the
      causes are and how to fix them.

    gaining priviledges (getting root on UNIX system).
    gaining access   (getting a shell when you shouldn't be able).

    test is usually bounded by constraints (time, money, physical access).

  ratings of security based on the level of resistance to penetration
    testing

    Orange Book (1985) - goverment specification for measuring security of
      system.

Three stages to penetration exercise:

1. External attacker without access
      No physical access, don't know ip/phone

      Goal: accumlate information about machine.

2. External attacker with access
      Goal: get on machine.

3. Internal attacker
      know how the system works

      Goal: aquire priveledges... read disallowed files...

Process:

        Knowledge (lear environment, system itself)
          configuration, system, installation
          government usually uses the design team to evaluate Orange book
rating
          look at design
            examine design verses goal
            network protocols
            inconsistancies

          build list of design issues
          look at implementation
            do programs do the things taht they are
              supposed to do
            read man pages looking for bounds problems
              do programs act like the man page says they do.

          figure out design should be and work from there

          look for implementation problems
            examine specs -> where are keys?

        Analyze and fix flaws
          classify flaws and develope fixes/workarounds for flaws

        Monitoring
          Some flaws cannot be fixed or are not worth fixing. Develope
            monitoring to give indication of when site being attacked.

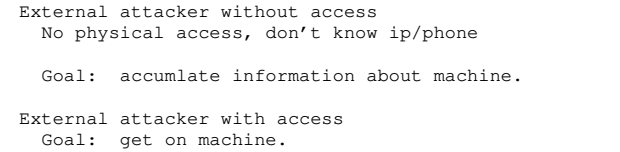
!! Record Everything !!

External Attacker
  Examine network protocols
    TCP/IP, FTP/TFTP, telnet, r* protocols, SMTP, NNTP, HTTP, NFS/NIS,
finger
    do they check for security?

Security problems:
  Backdoors:
    sendmail (1986) belived to be patched everwhere.
      flaw... connect to port, type "wiz" followed by "shell"... -> root
shell

  TOCTTOU (time of check to time of use):
    exa. program checks if the user should be able to access file... then
          does stuff to the file if user does have access. If between the
          check and the use an attacker changes the state of the system
          the attacker could access files that are disallowed.

Firewall -



  The firewall dissallows connections from one network to another.
  Must be securely maintained.
  All trafic between networks must go through it.

```
Systems we will be using:
        Solaris
        Data General DG/UX (B2 security rating)
          resistant to penetration
          has access control (MAC)
```