Security - Windows

Note Title 4/4/2011

Announcements

- Checkpoint for lab 4:
  part 1 (or start) by tomorrow
  email by 11:59 pm
  (check for downtime!)

Quiz Recap

Man-in-the-middle

switch versus hubs
↑ send everything to everyone

ARP - connects MAC to IP address

replay attack

Survey Says

- You like history

- You want more (?!) HW/labs

  overall, like DETER

- Practical lectures are good

Next lab (?)

Exploiting vulnerabilities in:

- SQL
- Perl
- C

} pick 1

OR

OS Hardening

OR

Computer Forensics

Windows history

Before NT, windows incorporated no real security.

(Really.)

We'll focus on the model built into the systems after the Windows 9x code base was scrapped.

Components on a Windows System (Security perspective)

① Security reference monitor (SRM)

- kernel-level
- access checks, audit logs, + user rights

② Local Security Authority (LSA)

- runs as lsass.exe
- governs local security policy:
  - gives tokens to accounts at login
  - password + auditing policies

③ Security Account Manager (SAM)

- database that stores local user information (in Windows System32 \ config)
- at login, the SAM process (Samsrv) takes logon (from WinLogon) + performs a lookup
- SAM does not perform logon - that is done by LSA
- binary file (not text)
- passwords stored w/ MD4 or PBKCS

4 Active directory (AD)

- LDAP (Light weight Directory Access Protocol) built into all Windows (from Windows Server 2000 on), used to support management & query operations

  Ex: email support

- Kerberos-based authentication: credentials are sent securely across network & verified at the central database

  (not LSA)

Local Versus Domain Accounts

- no AD!
- works w/ no internet
- can use workgroups for grouping of computers (but no central database)

- centrally managed
- more secure
- easier to set up network services (printers, etc.)

User SID (in AD)

S-1-5-21-AAA-BBB-CCC-RRR

- S: Simple SID
- 1: version #
- 5: identifier authority (5 means SECURITY_NT_AUTHORITY)
- 21: "not guaranted unique"
- AAA-BBB-CCC: unique # for domain
- RRR: relative ID (+1 for each account)

Usernames - two possibilites

- SAM format: DOMAIN\Username

  - Considered legacy format
  - only one I've ever seen

- User Principle Name (UPN):
  username@domain.company.com

  newer

Logon

- Username + password
  or username + smart card

- Third party support can add
  RSA/Secure ID or biometric devices

- Once logged in, a token is generated
  by OS ~~Ur~~ issued to user.
  (Contains SID, group info, + priviledges)
  This is assigned to every process
  run by user + is basis for
  access control.

Priviledges

- Systemwide permissions assigned to user accounts

Examples:

- backup files or directories
- system clock – Kerberos
- trusted computing base privilege (not even admin gets it!)
- debug programs privilege

(usually security issues)

about 40 total

Access Control Lists

Two kinds: DACL and System ACL

govern sensative OS files

DACL:

- Every object which requires access control is assigned a DACL
- Includes SID of owner plus list of access control entries (ACE)
- Each ACE is SID + access mask

The data structure supporting this is called the security descriptor (SD)

Example SD:

Owner : CORP\Blake
ACE [0] : Allow CORP\Paige Full Control
ACE [1] : Allow Administrators Full Control
ACE [2] : Allow CORP\Cheryl Read, Write + Delete

<u>Note</u> :
- No SACL here
- Owner has full control by default (but can change that in Vista)
- <u>No</u> implied access - default is deny

More Notes:

- Many developers request all access to an object, even if not needed. This is the prime reason apps fail on Windows XP (unless user is an Admin).

- Can also deny:
  ACE[0]: Deny Guests Full Control

- Put denies first!

Impersonation in Windows

Windows is multithreaded:

multiple copies of processes

Application processes may assume the identity of a user.

Why?

Security purposes, we should always run at user's access control level.

Mandatory Access Control

In Windows Vista, have Integrity Control, which goes further than DACLs.

Every object or principle is labeled:

- S-1-16-4096 (low integrity)
- S-1-16-8192 (medium)
- S-1-16-12288 (High)
- S-1-16-16384 (System)

(Default is medium)

MAC (cont.)

Can only write to objects of equal or lower integrity.

IE uses this the most, but almost all OS files are marked medium or higher.

Vulnerabilities

In 2001, MS changed its software development process to emphasize security + reduce bugs.
(reduced bugs by more than 50%)

Lifecycle:
- Mandatory security ed
- Secure / design requirements
- Threat modeling
- Attack surface analysis + reduction
- Secure coding requirements + tools
- " testing " "
- Security push
- Final security review
- security responce

Very successful!

Ex: Web server IIS version 6 only 3 reported vulnerabilities in 4 years (versus original IIS or Apache)

(SQL (2005 version) server had no vulnerabilities reported in the first 2 years!)

from MS perspective

System Hardening in Windows

(1) Attack Surface Reduction: 80/20 rule
- If the feature is not used by 80% of users, it should be off by default.

Issue: Many more non-technical users.

② Replace anonymous network protocols with authenticated ones

Ex: Blaster worm used remote procedure call (RPC) called anonymously.
SP2 required all RPCs to be authenticated.
When Zotob worm came out (exploiting RPC vulnerability in Plug 'n' Play), it was less successful even with the bug present, since the worm wasn't authenticated.

Best part: The user is unaware!