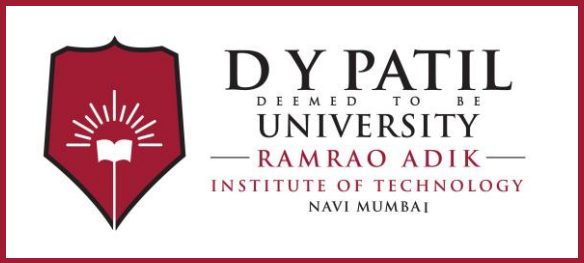


Information Security

Module: 6

- Faculty: Mrs. Bhavana Alte
- Mr. Prathmesh Gunjgur

Contents

Module 6:

Lecture 34: Operating System Security

Operating System Security

- The process of ensuring OS availability, confidentiality, integrity is known as operating system security. OS security refers to the processes or measures taken to protect the operating system from dangers, including viruses, worms, malware, and remote hacker intrusions. Operating system security comprises all preventive-control procedures that protect any system assets that could be stolen, modified, or deleted if OS security is breached.

Operating System Threats

- Program threats: The operating system's processes and kernel carry out the specified task as directed. Program Threats occur when a user program causes these processes to do malicious operations. The common example of a program threat is that when a program is installed on a computer, it could store and transfer user credentials to a hacker. There are various program threats. Some of them are as follows:
  - Virus: A virus may replicate itself on the system. As the user interacts with the program, the virus becomes embedded in other files and programs, potentially rendering the system inoperable.
  - Trojan Horse This type of application captures user login credentials. It stores them to transfer them to a malicious user who can then log in to the computer and access system resources.
  - Logic Bomb: A logic bomb is a situation in which software only misbehaves when particular criteria are met; otherwise, it functions normally.
  - Trap Door: A trap door is when a program that is supposed to work as expected has a security weakness in its code that allows it to do illegal actions without the user's knowledge.

Operating System Threats

- System Threats: System threats are described as the misuse of system services and network connections to cause user problems. These threats may be used to trigger the program threats over an entire network, known as program attacks. System threats make an environment in which OS resources and user files may be misused. There are various system threats. Some of them are as follows:
  - Port Scanning: It is a method by which the cracker determines the system's vulnerabilities for an attack. It is a fully automated process that includes connecting to a specific port via TCP/IP.
  - Worm: The worm is a process that can choke a system's performance by exhausting all system resources. A Worm process makes several clones, each consuming system resources and preventing all other processes from getting essential resources. Worm processes can even bring a network to a halt.
  - Denial of Service: Denial of service attacks usually prevents users from legitimately using the system. For example, if a denial-of-service attack is executed against the browser's content settings, a user may be unable to access the internet.

Operating System Threats

- Malware: It contains viruses, worms, trojan horses, and other dangerous software. These are generally short code snippets that may corrupt files, delete the data, replicate to propagate further, and even crash a system. The malware frequently goes unnoticed by the victim user while criminals silently extract important data.

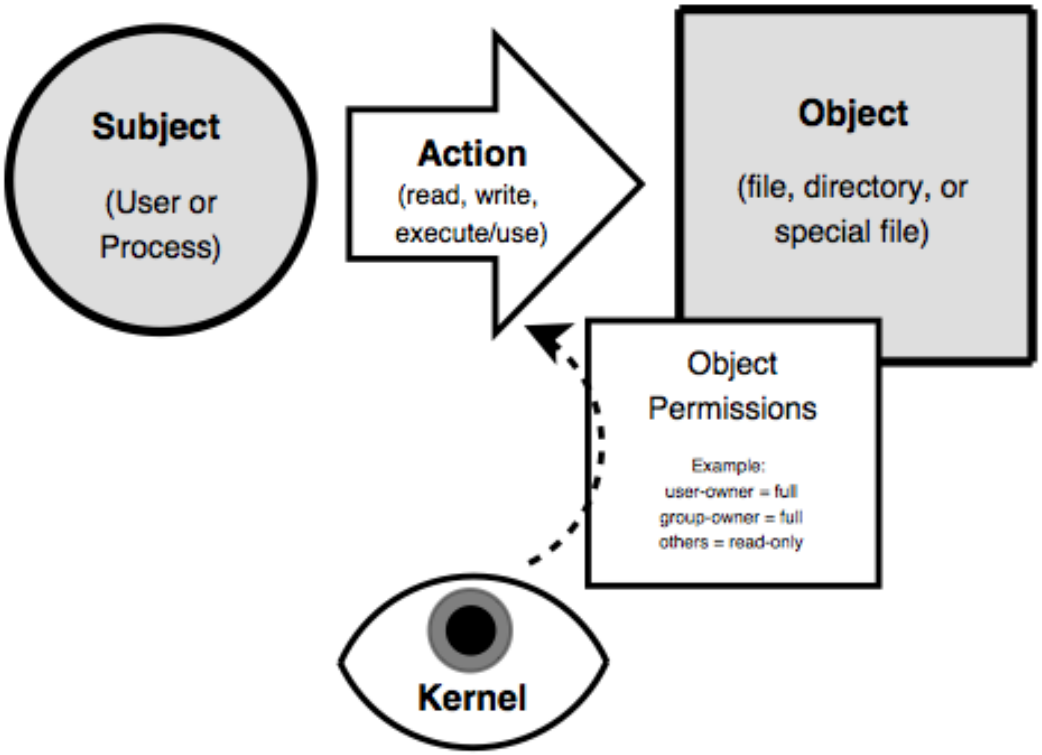
- Network Intrusion: Network intruders are classified as masqueraders, misfeasors, and unauthorized users. A masquerader is an unauthorized person who gains access to a system and uses an authorized person's account. A misfeasor is a legitimate user who gains unauthorized access to and misuses programs, data, or resources. A rogue user takes supervisory authority and tries to evade access constraints and audit collection.

- Buffer Overflow: It is also known as buffer overrun. It is the most common and dangerous security issue of the operating system. It is defined as a condition at an interface under which more input may be placed into a buffer and a data holding area than the allotted capacity, and it may overwrite other information. Attackers use such a situation to crash a system or insert specially created malware that allows them to take control of the system.

Module 6:

Lecture 35: Linux Security

Linux Security Architecture



Linux Security Architecture

- Linux's security model relies on Discretionary Access Controls (DAC).
- In the Linux DAC system, there are users, each of which belongs to one or more groups; and there are also objects: files and directories. Users read, write, and execute these objects, based on the objects' permissions, of which each object has three sets: one each defining the permissions for the object's user-owner, group-owner, and "other" (everyone else). These permissions are enforced by the Linux kernel, the "brain" of the operating system.

- The system superuser account, called "root," has the ability to both take ownership and change the permissions of all objects in the system.

Linux Security Architecture

- a user-account (user)
  - represents someone capable of using files
  - associated both with humans and processes
- a group-account (group)
  - is a list of user-accounts
  - users have a main group
  - may also belong to other groups
- user's details are kept in /etc/password
- additional group details in /etc/group

```
maestro:x:200:100:Maestro Edward Hizzersands:/home/maestro:/bin/bash
```

Listing 25-1: An /etc/password Entry For the User "maestro"

```
conductors:x:100:
pianists:x:102:maestro,volodya
```

Listing 25-2: Two /etc/group Entries

D Y PATIL DEEMED TO BE UNIVERSITY RAMRAO ADIK INSTITUTE OF TECHNOLOGY NAVI MUMBAI

Linux Security Architecture

- the first field contains the name of the user account, "maestro;"
- the second field ("x") is a placeholder for maestro's password (which is actually stored in /etc/shadow);
- the third field shows maestro's numeric userid (or "uid," in this case "200");
- and the fourth field shows the numeric groupid (or "gid," in this case "100") of maestro's main group membership.
- The remaining fields specify a comment, maestro's home directory, and maestro's default login shell.

```
maestro:x:200:100:Maestro Edward Hizzersands:/home/maestro:/bin/bash
```

Listing 25-1: An /etc/password Entry For the User "maestro"

```
conductors:x:100:
pianists:x:102:maestro,volodya
```

Listing 25-2: Two /etc/group Entries

Linux Security Architecture

- From /etc/group, each line simply contains a groupname,
- a group-password (usually unused — "x" is a placeholder),
- And numeric group-id (gid),
- a comma-delimited list of users with "secondary" memberships in the group.

```
maestro:x:200:100:Maestro Edward Hizzersands:/home/maestro:/bin/bash
```

Listing 25-1: An /etc/password Entry For the User "maestro"

```
conductors:x:100:
pianists:x:102:maestro,volodya
```

Listing 25-2: Two /etc/group Entries

Linux Security Architecture

- The simplest way to modify /etc/password and /etc/group in order to create, modify, and delete user accounts is via the commands useradd, usermod, and userdel, respectively.
- All three of these commands can be used to set and modify group-memberships, and all three commands are well documented in their respective manpages.

```
maestro:x:200:100:Maestro Edward Hizzersands:/home/maestro:/bin/bash
```

Listing 25-1: An /etc/password Entry For the User "maestro"

```
conductors:x:100:
pianists:x:102:maestro,volodya
```

Listing 25-2: Two /etc/group Entries

Linux Security Architecture

- files have two owners: a user & a group
- each with its own set of permissions
- with a third set of permissions for other
- permissions are to read/write/execute in order user/group/other, cf.
- its user-owner ("maestro") may read and write/delete the file ("rw-"); its group-owner ("conductors") may also read and write/delete the file ("rw-"); but that other users (who are neither "maestro" nor members of "conductors") may only read the file.
- There's a third permission besides "read" and "write": "execute," denoted by "x" (when set).
- Permissions are usually set via the "chmod" command.

```
-rw-rw-r--  1  maestro  conductors  35414  Mar 25 01:38 baton_dealers.txt
```

Listing 25-3: File-Listing Showing Permissions

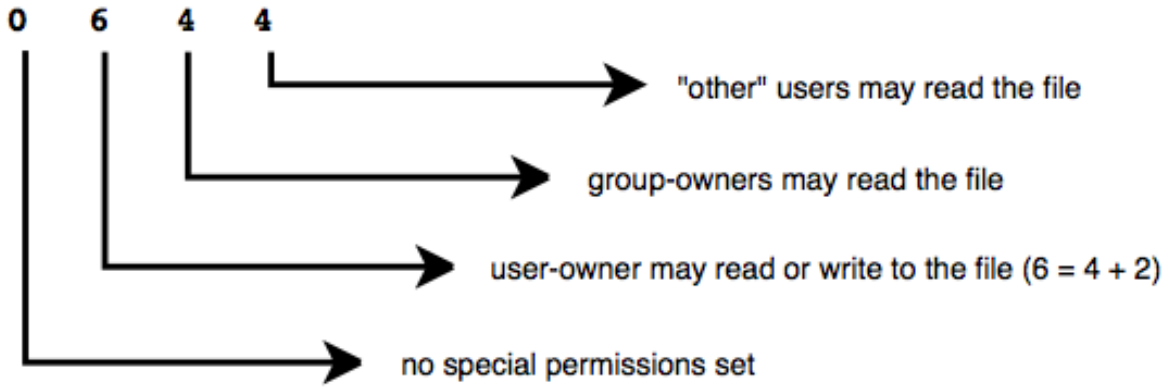
Linux Security Architecture

- Directory permissions work slightly differently from permissions on regular files. "Read" and "write" are similar; for directories these permissions translate to "list the directory's contents" and "create or delete files within the directory", respectively. "Execute" is less intuitive; for directories, "execute" translates to "use anything within or change working directory to this directory“.

- If a user or group has execute permissions on a given directory, the user or group can list that directory's contents, read that directory's files and change its own working directory to that directory, as with the command "cd". If a user or group does not have execute permissions on a given directory, its will be unable to list or read anything in it, regardless of the permissions set on the things inside.

```
bash-$ ls -l extreme_casseroles
drwxr-x---  8  biff  drummers  288  Mar 25 01:38 extreme_casseroles
```

Linux Security Architecture



Linux Security Architecture

- Originally used to lock file in memory, so it would load more quickly
- Now used on directories to limit the ability to delete
- if set must own file or dir to delete
- other users cannot delete even if have write
- Set using chmod command with +t flag, e.g.
- chmod +t extreme_casseroles
- Directory listing includes t or T flag

```
drwxrwx--T   8   biff   drummers   288   Mar 25 01:38 extreme_casseroles
```

- setuid bit means program "runs as" owner no matter who executes it
- setgid bit means run as a member of the group which owns it

Linux Security Architecture

- Kernel space refers to memory used by the Linux kernel and its loadable modules (e.g., device drivers).
- User space refers to memory used by all other processes.
- Because the kernel enforces the Linux DAC and, in real terms, dictates system reality, it's extremely important to isolate kernel space from user space. For this reason, kernel space is never swapped to hard disk. Only root may load and unload kernel modules.

Linux Vulnerabilities

- Buffer overflows: Buffer overflow is a software coding error or vulnerability that can be exploited by hackers to gain unauthorized access to corporate systems. It is one of the best-known software security vulnerabilities yet remains fairly common. This is partly because buffer overflows can occur in various ways and the techniques used to prevent them are often error-prone.

- Race conditions: A race condition occurs when two threads access a shared variable at the same time. The first thread reads the variable, and the second thread reads the same value from the variable. Then the first thread and second thread perform their operations on the value, and they race to see which thread can write the value last to the shared variable. The value of the thread that writes its value last is preserved, because the thread is writing over the value that the previous thread wrote

Linux Vulnerabilities

- Abuse of programs run “setuid root”: A setuid root program is a root-owned program with its setuid bit set; that is, a program that runs as root no matter who executes it. If a setuid root program can be exploited or abused in some way, then otherwise unprivileged users may be able to use that program to wield unauthorized root privileges, possibly including opening a root shell.

- Denial of service (DoS): A Denial-of-Service (DoS) attack is an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic, or sending it information that triggers a crash.

Linux Vulnerabilities

- Web application vulnerabilities: Web applications written in scripting languages such as PHP, Perl, and Java may not be as prone to classic buffer overflows, cross-site scripting, SQL code injection, other vulnerabilities described in depth by the Open Web Application Security Project.

- Rootkit attacks: This attack, which allows an attacker to cover her tracks, typically occurs after root compromise: If a successful attacker is able to install a rootkit before being detected, all is very nearly lost. Rootkits began as collections of “hacked replacements” for common UNIX commands (ls, ps, etc.) that behaved like the legitimate commands they replaced, except for hiding an attacker’s files, directories and processes. For example, if an attacker was able to replace a compromised Linux system’s ls command with a rootkit version of ls, then anyone executing the ls command to view files and directories would see everything except the attacker’s files and directories.

Linux Securities

OS Installation

- Security begins with O/S installation, especially choosing what software is run: since unused applications liable to be left in default, un-hardened and un-patched state.

- Generally should not run: X Window system, RPC services, R-services, inetd, SMTP daemons, telnet, etc.

- Also have some initial system software configuration:
  - setting root password
  - creating a non-root user account
  - setting an overall system security level
  - enabling a simple host-based firewall policy
  - enabling SELinux

Linux Securities

Patch Management

- Installed server applications must be:
  - configured securely
  - kept up to date with security patches
- Patching can never win “patch rat-race”
- Have tools to automatically download and install security updates
  - e.g. up2date, YaST, apt-get
  - note should not run automatic updates on change-controlled systems without testing

Network Access Controls

- TCP wrappers is a key tool to check access
  - originally tcpd inetd wrapper daemon
  - before allowing connection to service checks
    - if requesting host explicitly in hosts.allow is ok
    - if requesting host explicitly in hosts.deny is blocked
    - if not in either is ok
  - checks on service, source IP, username
  - now often part of app using libwrappers

Linux Securities

User Management

- Guiding principles in user-account security:
  - need care setting file / directory permissions
  - use groups to differentiate between roles
  - use extreme care in granting / using root privs
- Commands: chmod, useradd/mod/del, groupadd/mod/del, passwd, chage
- Info in files /etc/passwd & /etc/group
- Manage user's group memberships
- Set appropriate password ages: /etc/login.defs

Logging

- Effective logging is a key resource
- Linux logs using syslogd or Syslog-NG
  - receive log data from a variety of sources
  - sorts by facility (category) and severity
  - writes log messages to local/remote log files
- Syslog-NG preferable because it has:
  - variety of log-data sources / destinations
  - much more flexible "rules engine" to configure
  - can log via TCP which can be encrypted
- should check and customize the defaults

Linux Security Modules

- SELinux: Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies, including mandatory access controls (MAC).
- AppArmor: AppArmor ("Application Armor") is a Linux kernel security module that allows the system administrator to restrict programs' capabilities with per-program profiles. Profiles can allow capabilities like network access, raw socket access, and the permission to read, write, or execute files on matching paths. AppArmor supplements the traditional Unix discretionary access control (DAC) model by providing mandatory access control (MAC).
- Smack: Smack (full name: Simplified Mandatory Access Control Kernel) is a Linux kernel security module that protects data and process interaction from malicious manipulation using a set of custom mandatory access control (MAC) rules, with simplicity as its main design goal.
- TOMOYO: Tomoyo Linux is a MAC implementation for Linux that can be used to increase the security of a system, while also being useful purely as a systems analysis tool. Tomoyo Linux focuses on system behaviour. Tomoyo Linux allows each process to declare behaviours and resources needed to achieve their purpose. When protection is enabled, Tomoyo Linux restricts each process to the behaviours and resources allowed by the administrator.

Module 6:

Lecture 36: Windows Security Architecture

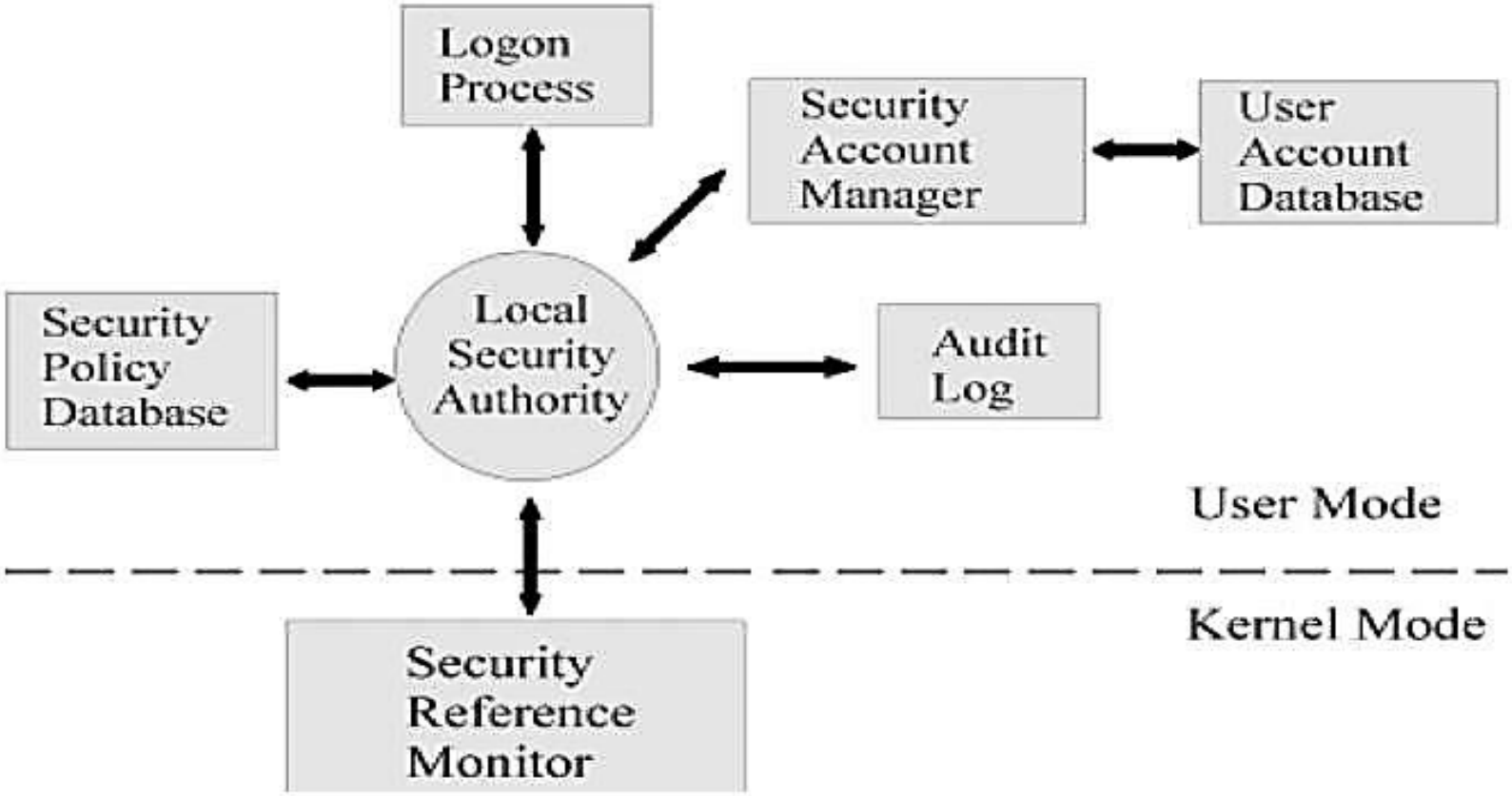
Windows Security Architecture

- Security Reference Monitor (SRM): a kernel-mode component that performs access checks, generates audit log entries, and manipulates user rights (privileges)

- Local Security Authority (LSA): responsible for enforcing local security policy

- Security Account Manager (SAM)
  - a database that stores user accounts and local users and groups security information
  - local logins perform lookup against SAM DB
  - passwords are stored using MD4

Windows Security Architecture



Windows Security Architecture

- LSA is the Central Part of NT Security. It is also known as Security Subsystem.
- In Windows 2000, the LSA is responsible for validating users for both local and remote logons. The LSA also maintains the local security policy.

- During the local logon to a machine, a person enters his name and password to the logon dialog. This information is passed to the LSA, which then calls the appropriate authentication package.

- The password is sent in a non-reversible secret key format using a one-way hash function.
- The LSA then queries the SAM database for the User's account information. If the key provided matches the one in the SAM, the SAM returns the users SID and the SIDs of any groups the user belongs to.

- The LSA then uses these SIDs to generate the security access token.

Windows Security Architecture

- The Security Accounts Manager is a database in the Windows operating system (OS) that contains user names and passwords. SAM is part of the registry and can be found on the hard disk.
- This service is responsible for making the connection to the SAM database (Contains available user-accounts and groups).
- The SAM database can either be placed in the local registry or in the Active Directory (If available).
- In the SAM, each user account can be assigned a Windows password which is in encrypted form. If someone attempts to log on to the system and the user name and associated passwords match an entry in the SAM, a sequence of events takes place ultimately allowing that person access to the system. If the user name or passwords do not properly match any entry in the SAM, an error message is returned requesting that the information be entered again.
- When you make a New User Account with a Password, it gets stored in the SAM File.
- Windows Security Files are located at “C:\Windows\System32\Config\SAM”.
- The moment operating system starts, the SAM file becomes inaccessible.

Windows Security Architecture

- The Security Reference Monitor is a security architecture component that is used to control user requests to access objects in the system. The SRM enforces the access validation and audit generation. Windows NT forbids the direct access to objects. Any access to an object must first be validated by the SRM. For example, if a user wants to access a specific file the SRM will be used to validate the request. The Security Reference Monitor enforces access validation and audit generation policy.
- The reference monitor verifies the nature of the request against a table of allowable access types for each process on the system. For example, Windows 3.x and 9x operating systems were not built with a reference monitor, whereas the Windows NT line, which also includes Windows 2000 and Windows XP, was designed with an entirely different architecture and does contain a reference monitor.

Windows Security Architecture

- Active Directory: Active Directory (AD) is Microsoft's LDAP directory included with Windows Server 2000 and later. All currently supported client versions of Windows,
- including Windows XP and Windows 7, can communicate with AD to perform
- security operations including account logon.
- A Windows client will authenticate using AD when the user logs on to the computer using a domain account rather than a local account. Like the SAM scenario, the
- user's credential information is sent securely across the network, verified by
- AD, and then, if the information is correct, the user can log on at the computer. "credential" and not "password" because a credential might take some other form, such as a public and private key pair bound to an X.509 certificate on a smart card. This is why most corporate laptops include smartcard readers

Windows Vulnerabilities

- Mandatory security education
- Secure design requirements
- Threat modeling
- Attack surface analysis and reduction
- Secure coding requirements and tools
- Secure testing requirements and tools
- Security push
- Final security review
- Security response

Windows Securities

- Account defenses:
  - user accounts can have privileged SIDs
  - least privilege dictates that users operate with just enough privilege for tasks
  - Windows XP users in local Administrators
    - for application compatibility reasons
    - can use “Secondary Logon” to run applications
    - also restricted tokens reduce per-thread privilege
  - Windows Vista reverses default with UAC
    - users prompted to perform a privileged operation
    - unless admin on Server

Windows Securities

- Low Privileged Account:
  - Windows services are long-lived processes started after booting
    - many ran with elevated privileges
    - but many do not need elevated requirements
  - Windows XP added Local Service and Network service accounts
    - allow a service local or network access
    - otherwise operate at much lower privilege level
  - Windows XP SP2 split RPC service (RPCSS) in two (RPCSS and DCOM Server Process)
    - example of least privilege in action, see also IIS6
    - direct result of Blastr worm

Windows Securities

- Account defenses:
  - user accounts can have privileged SIDs
  - least privilege dictates that users operate with just enough privilege for tasks
  - Windows XP users in local Administrators
    - for application compatibility reasons
    - can use “Secondary Logon” to run applications
    - also restricted tokens reduce per-thread privilege
  - Windows Vista reverses default with UAC
    - users prompted to perform a privileged operation
    - unless admin on Server

Windows Securities

- Stripping Priviledges:
  - another defense is to strip privileges from an account soon after an application starts
    - e.g. Index server process runs as system to access all disk volumes
    - but then sheds any unneeded privileges as soon as possible
    - using AdjustTokenPrivileges
  - Windows Vista can define privileges required by a service
  - using ChangeServiceConfig2

Windows Securities

- Network defenses:
  - have IPSec and IPv6 with authenticated network packets enabled by default in Windows Vista
    - IPv4 also enabled by default, expect less use
  - have built-in software firewall
    - block inbound connections on specific ports
      - Vista can allow local net access only
    - optionally block outbound connections (Vista)
    - default was off (XP) but now default on (Vista)

- Memory Corruption defenses.
- Browser defenses

Windows Securities

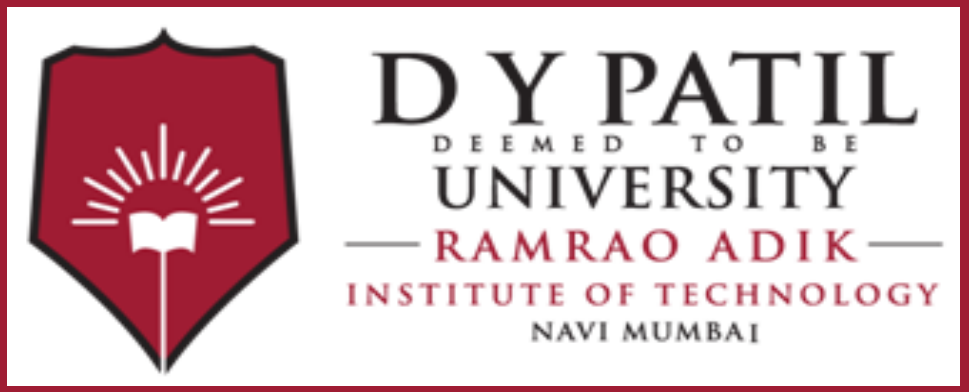
- Memory Corruption defenses.
  - many compromises exploit buffer overruns
  - Windows Vista has “Stack-Based Buffer Overrun Detection (/GS)” default enabled
    - Defends source code compiled with special /GS option
    - does not affect every function; only those with at least 4-bytes of contiguous stack data and that takes a pointer or buffer as an argument
  - against “classic stack smash”
  - No eXecuteNamed (NX) / Data Execution Prevention (DEP) / eXecution Disable (XD)
    - prevent code executing in data segments
    - as commonly used by buffer overrun exploits
    - applications linked with /NXCOMPAT option
  - Stack Randomization (Vista only)
    - randomizes thread stack base addresses
  - Heap-based buffer overrun defenses:
    - add and check random value on each heap block
    - heap integrity checking
    - heap randomization (Vista only)

Windows Securities

- Browser defenses:
  - web browser is a key point of attack
    - via script code, graphics, helper objects
  - Microsoft added many defenses to IE7
    - ActiveX opt-in
      - unloads ActiveX controls by default
      - when any then first run prompts user to confirm
  - protected mode
    - IE runs at low integrity level (see earlier)
    - so more difficult for malware to manipulate O/S)

Windows Securities

- Cryptographic services:
  - low-level crypto for encryption, hashing, signing
  - Encrypting File System (EFS)
    - allows files / directories to be encrypted / decrypted transparently for authorized users
    - generates random key, protected by DPAPI
  - Data Protection API (DPAPI)
    - manages encryption key maintenance protection
    - keys derived in part from user's password
  - BitLocker Drive Encryption
    - encrypts an entire volume with AES
    - key either on USB or TPM chip


D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Thank You