Linux OS Hardening

Biswajit Paul

(biswajit@cair.res.in)

Agenda

- Linux OS hardening : What and why ?
- How can we Harden Linux OS ?
- A tour of the on going Linux Hardening projects.

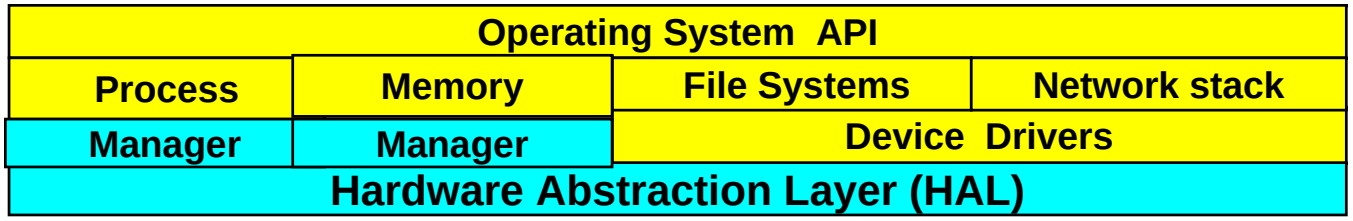
Introduction

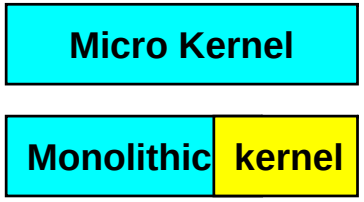
Operating System

The system software responsible for the direct control and management of hardware and basic system operations, as well as running applications such as servers, security software.

Unix Like : *AIX*, *HP-UX*, *Solaris*, *IRIX*, *Minix*, *Linux*.
Non Unix : *NetWare*, *Dos*, *Windows*.

➢ Monolithic vs Micro kernel





Windows

A family of personal computer operating systems developed by Microsoft Corporation, brain child of Bill Gates.

- ✓ User friendly Interfaces.
- ✓ Most popular OS till date.
- ✓ Holds **90%** OS market share.

➢ **Features :** *multi(tasking, processor, user), Virtual memory.*

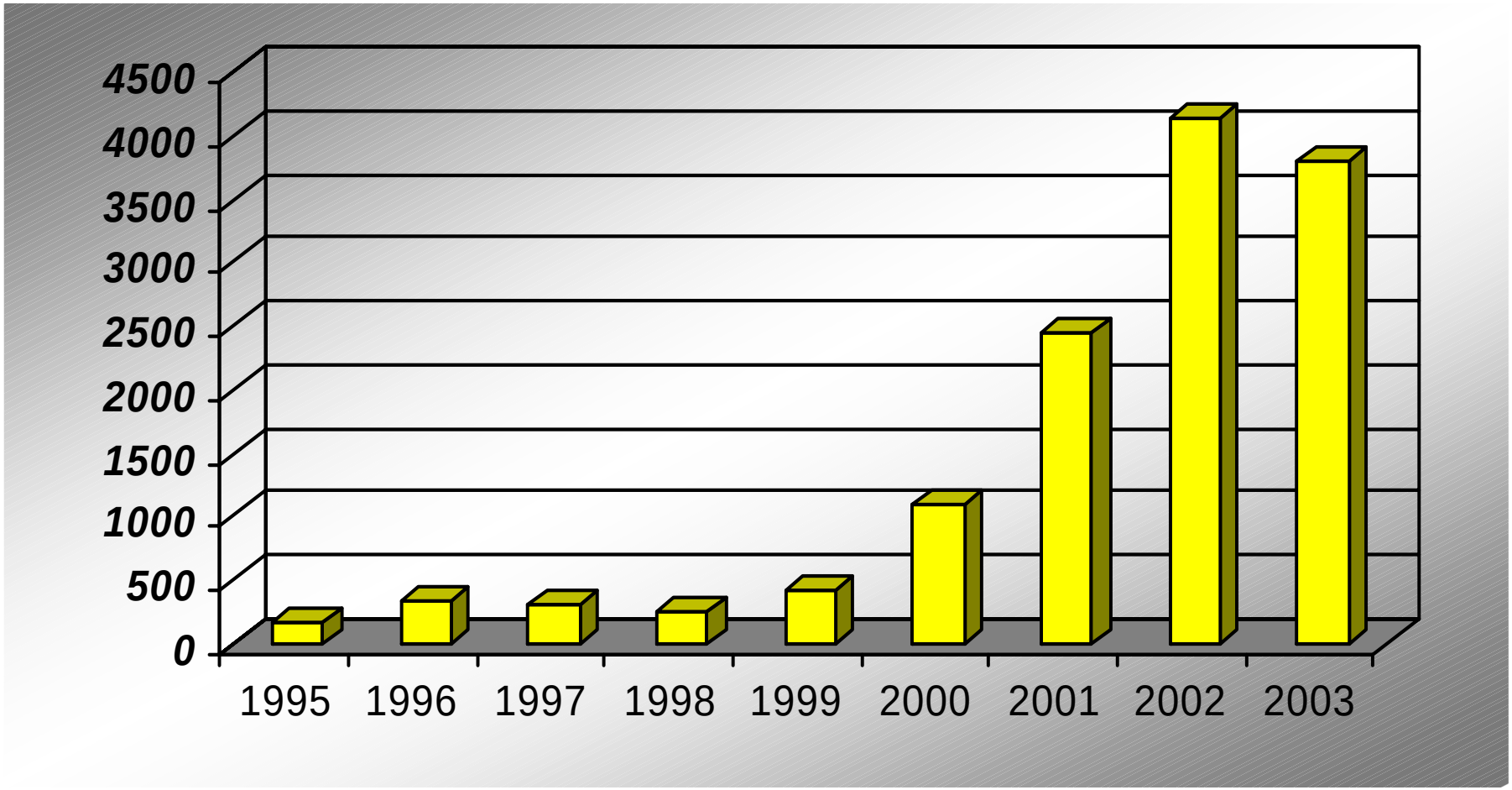
Linux

Linux is a free Unix-type operating system originally implemented by Linus Torvalds in 1991 with GNU software. Developers are from around the globe.

- ✓ Open Source Operating System.
- ✓ " Many Eyeballs" Theory .
- ✓ Free (*freedom to modify*).

➢ **Features :** *multi(tasking, processor, user), Virtual memory, Support for max number of File Systems and wide variety of hardware architecture* **.**

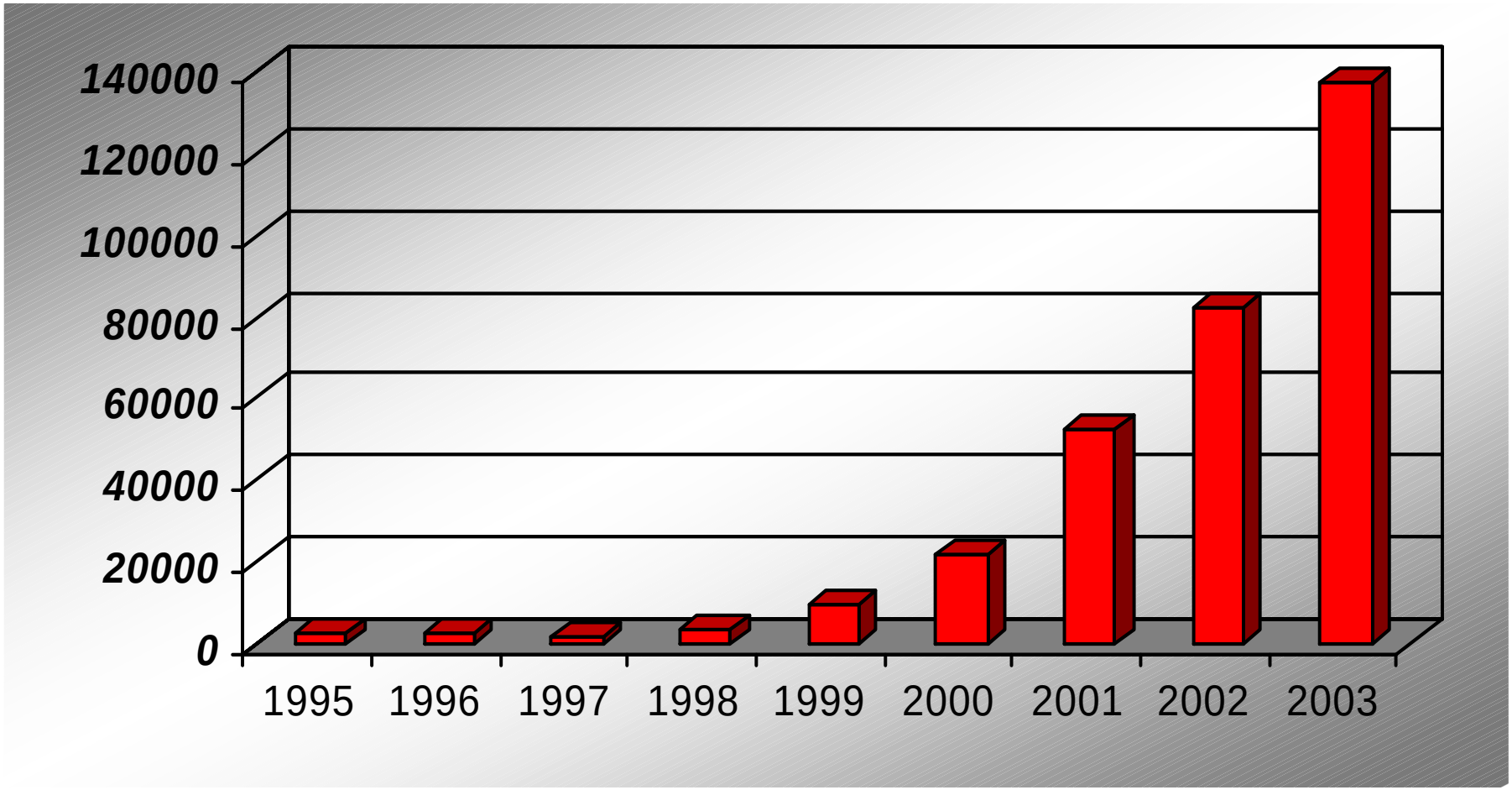
Why Security ?



Vulnerabilities reported

Source : http://cert.org

Why Security ?

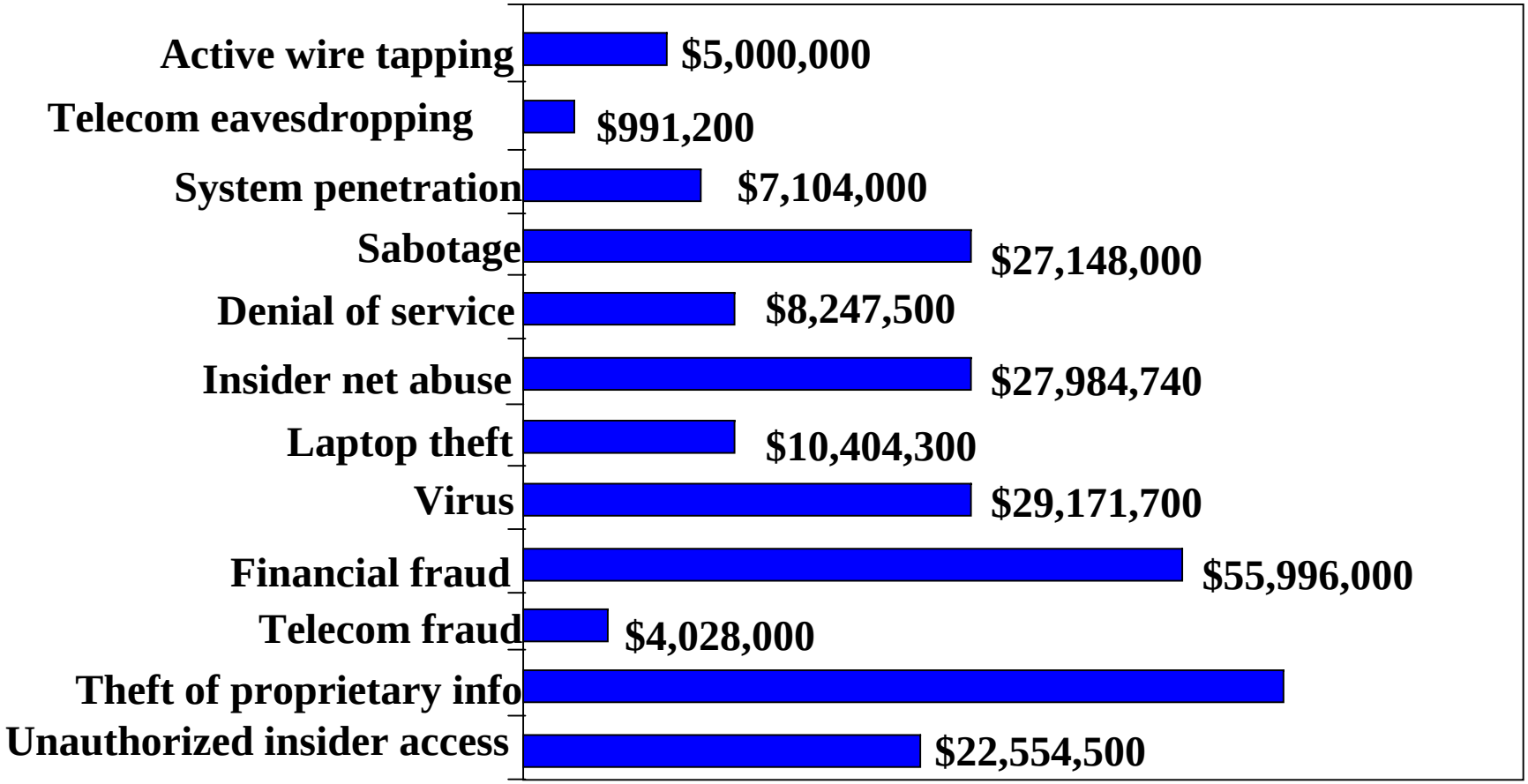


Incident reported

Source : http://cert.org

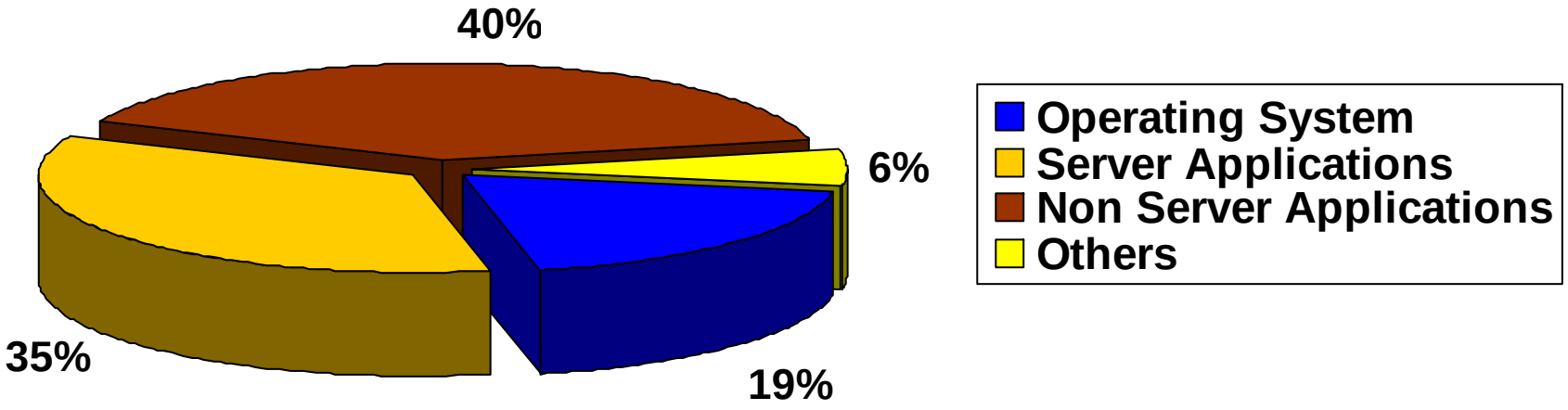
Why Security ?

Financial Losses by Type of attack



Source : Unknown

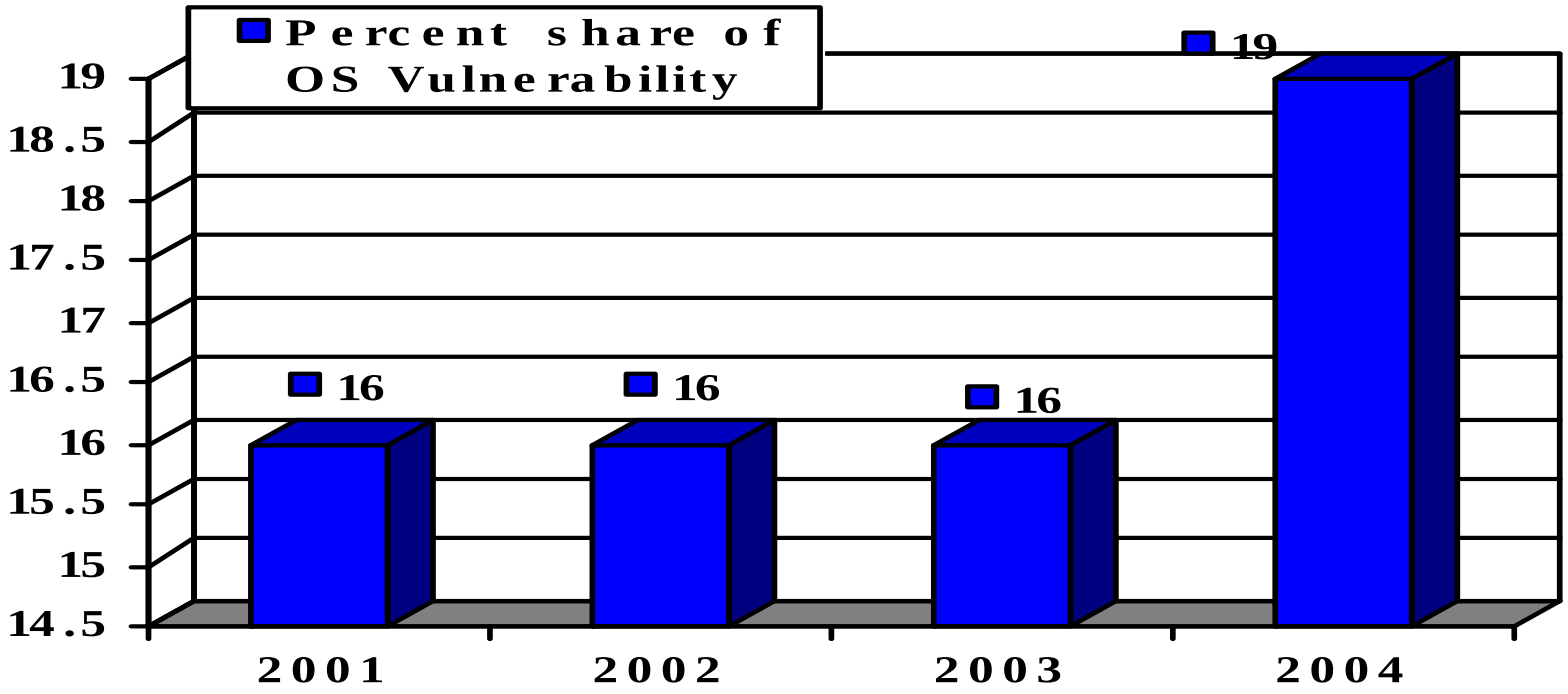
Why Secure Operating System ?



Percent share of Vulnerabilities in 2004

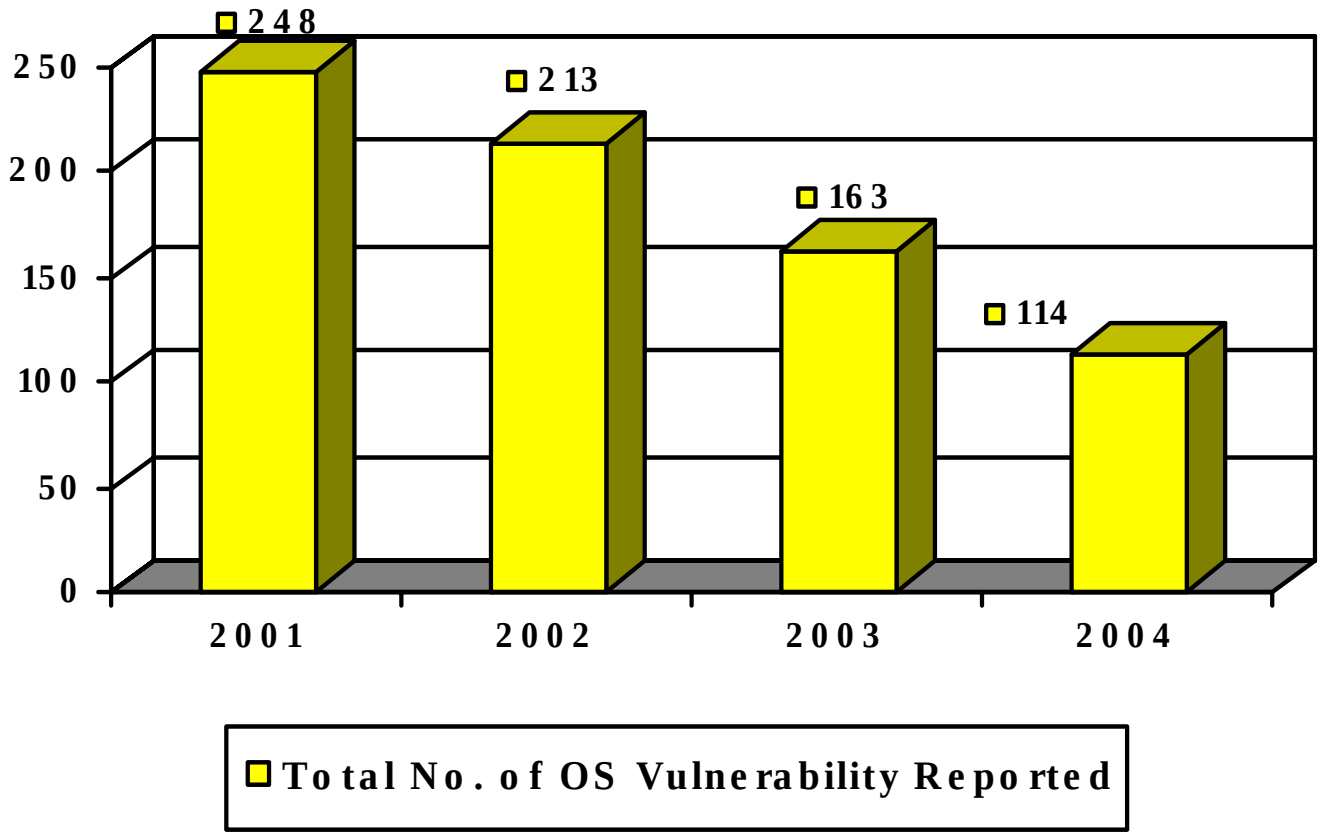
Source : http://icat.nist.gov/icat.cfm?function=statistics

Why Secure Operating System ?



Source : http://icat.nist.gov/icat.cfm?function=statistics

Why Secure Operating System ?

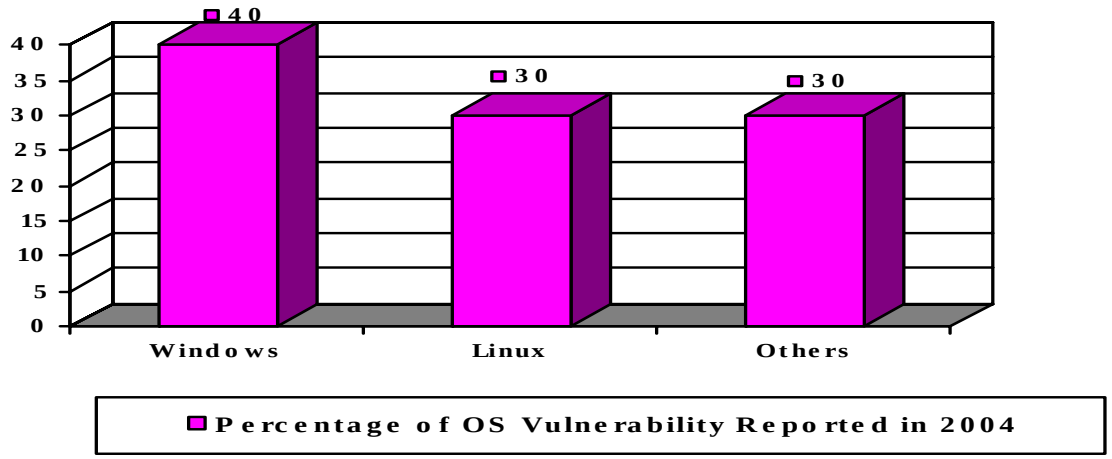


Source : http://icat.nist.gov/icat.cfm?function=statistics

Existing OS : Why not secure ?

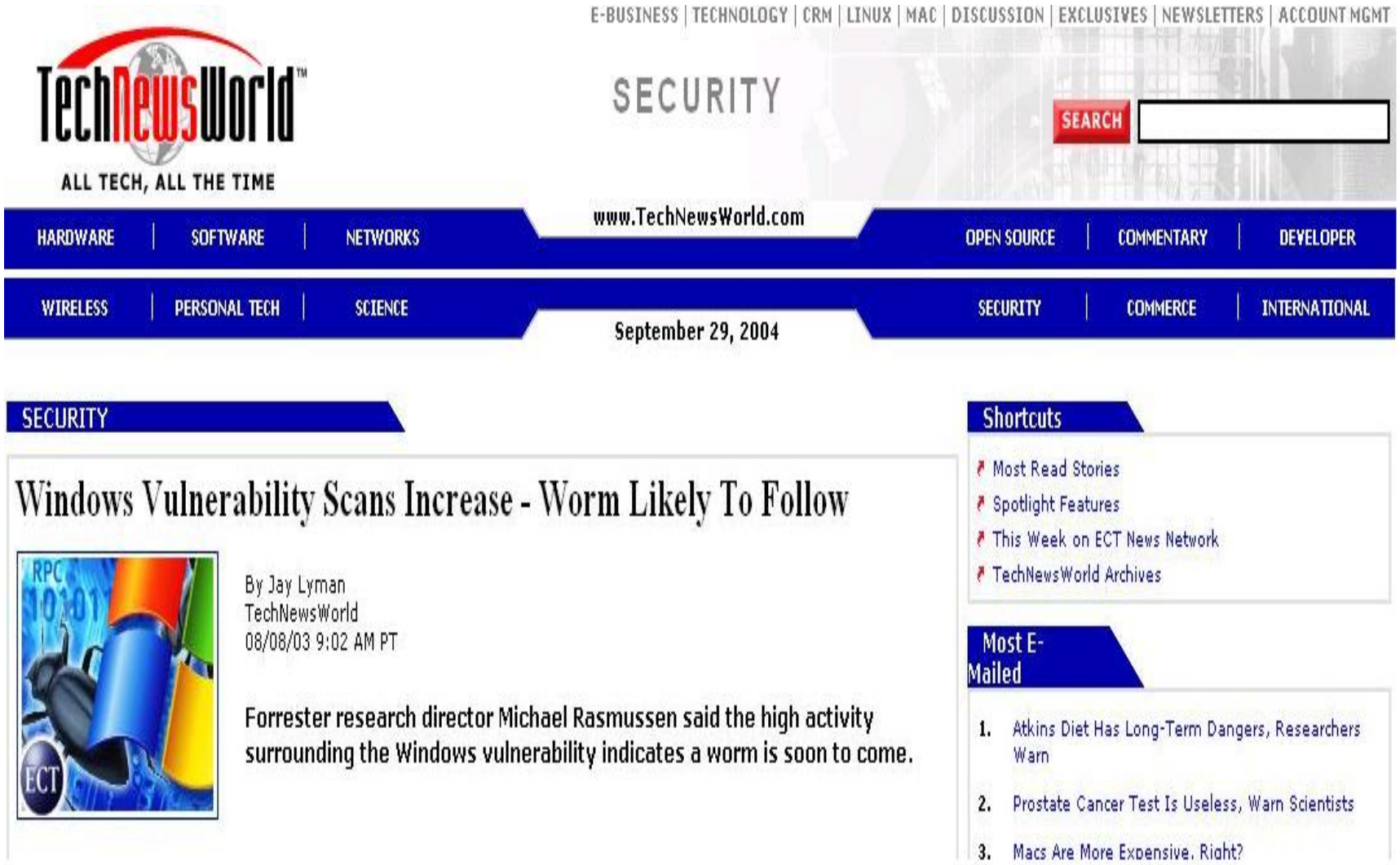
- **Design philosophy :**
  - ✓ User convenience.
  - ✓ No perception of current security threats.

- **Vulnerability in Popular OS:**



Source : Compiled from http://cert.org

Windows ?



E-BUSINESS | TECHNOLOGY | CRM | LINUX | MAC | DISCUSSION | EXCLUSIVES | NEWSLETTERS | ACCOUNT MGMT

TechNewsWorld™
ALL TECH, ALL THE TIME

SECURITY

SEARCH

www.TechNewsWorld.com

HARDWARE | SOFTWARE | NETWORKS | OPEN SOURCE | COMMENTARY | DEVELOPER

WIRELESS | PERSONAL TECH | SCIENCE | SECURITY | COMMERCE | INTERNATIONAL

September 29, 2004

SECURITY

Windows Vulnerability Scans Increase - Worm Likely To Follow

By Jay Lyman
TechNewsWorld
08/08/03 9:02 AM PT

Forrester research director Michael Rasmussen said the high activity surrounding the Windows vulnerability indicates a worm is soon to come.

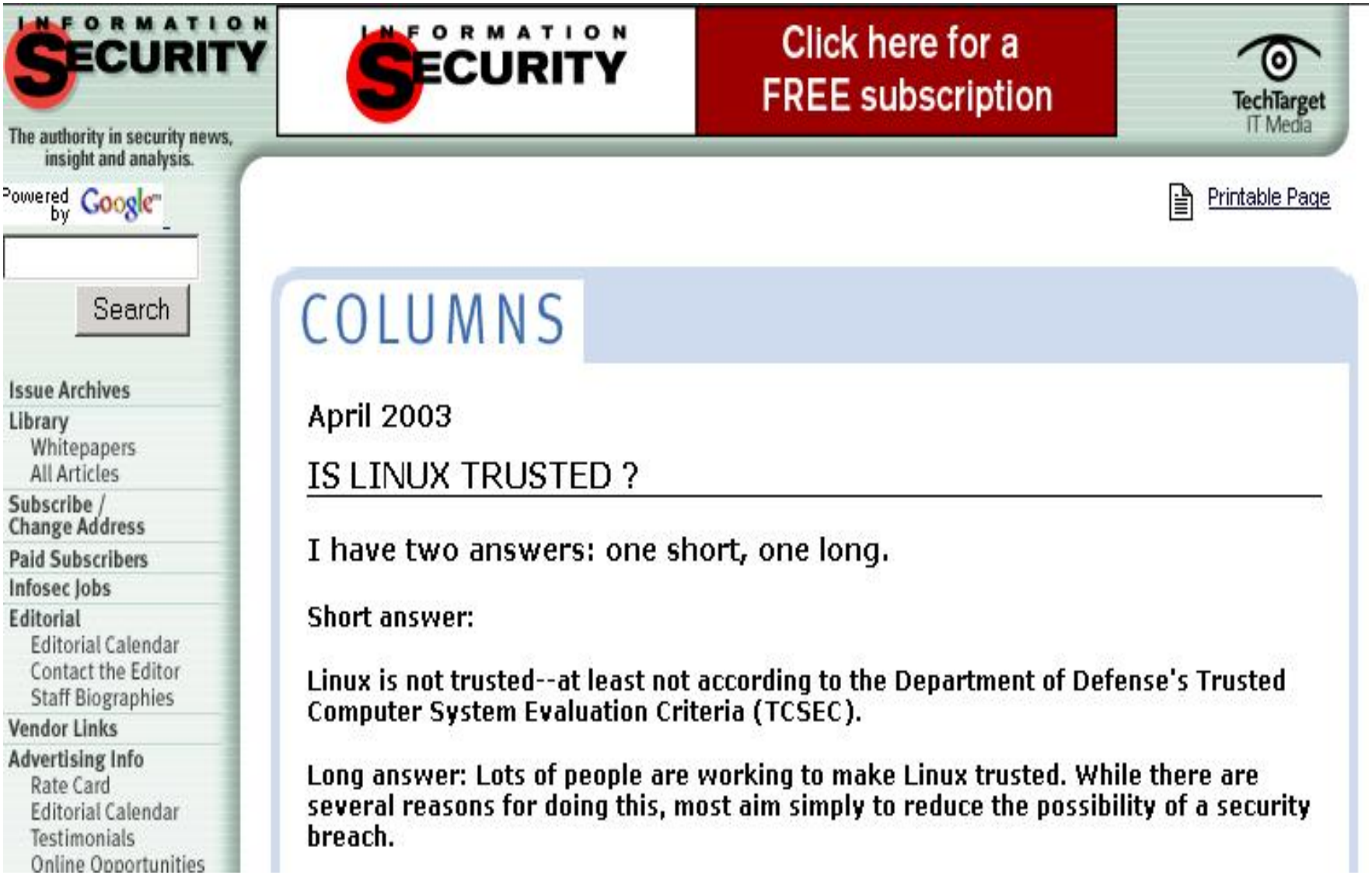
Shortcuts

- Most Read Stories
- Spotlight Features
- This Week on ECT News Network
- TechNewsWorld Archives

Most E-Mailed

1. Atkins Diet Has Long-Term Dangers, Researchers Warn
2. Prostate Cancer Test Is Useless, Warn Scientists
3. Macs Are More Expensive, Right?

Linux ?



INFORMATION SECURITY

The authority in security news, insight and analysis.

Powered by Google

Search

Issue Archives
Library
Whitepapers
All Articles
Subscribe / Change Address
Paid Subscribers
Infosec Jobs
Editorial
Editorial Calendar
Contact the Editor
Staff Biographies
Vendor Links
Advertising Info
Rate Card
Editorial Calendar
Testimonials
Online Opportunities

INFORMATION SECURITY

Click here for a FREE subscription

TechTarget IT Media

Printable Page

COLUMNS

April 2003

IS LINUX TRUSTED ?

I have two answers: one short, one long.

Short answer:

Linux is not trusted--at least not according to the Department of Defense's Trusted Computer System Evaluation Criteria (TCSEC).

Long answer: Lots of people are working to make Linux trusted. While there are several reasons for doing this, most aim simply to reduce the possibility of a security breach.

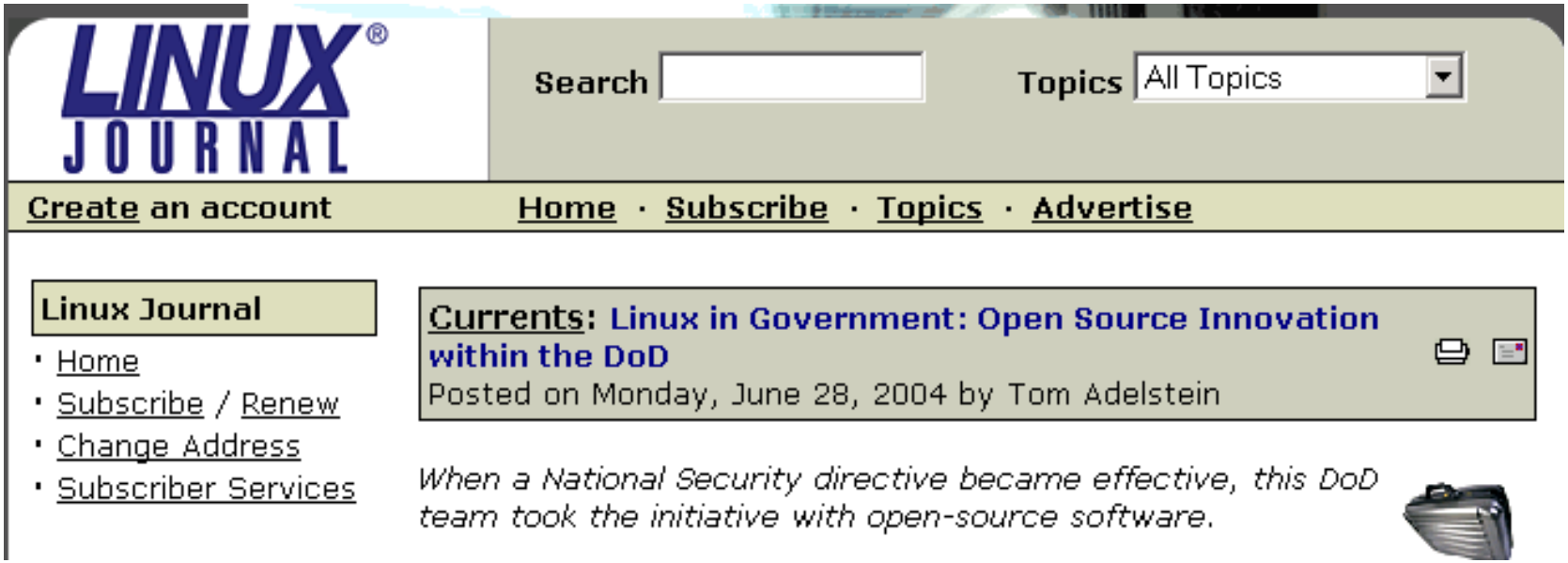
Solution …

- **Design a Secure Operating System ….!**
- **Customize a existing Operating System.**

2nd approach seems to be better. But which operating system ?

LINUX ?

- **Open Source.**
  - ✓ Can verify and modify code
  - ✓ Development is not in the custody of one person or organization.
- **“ Many Eyeballs” Theory.**



Linux OS hardening

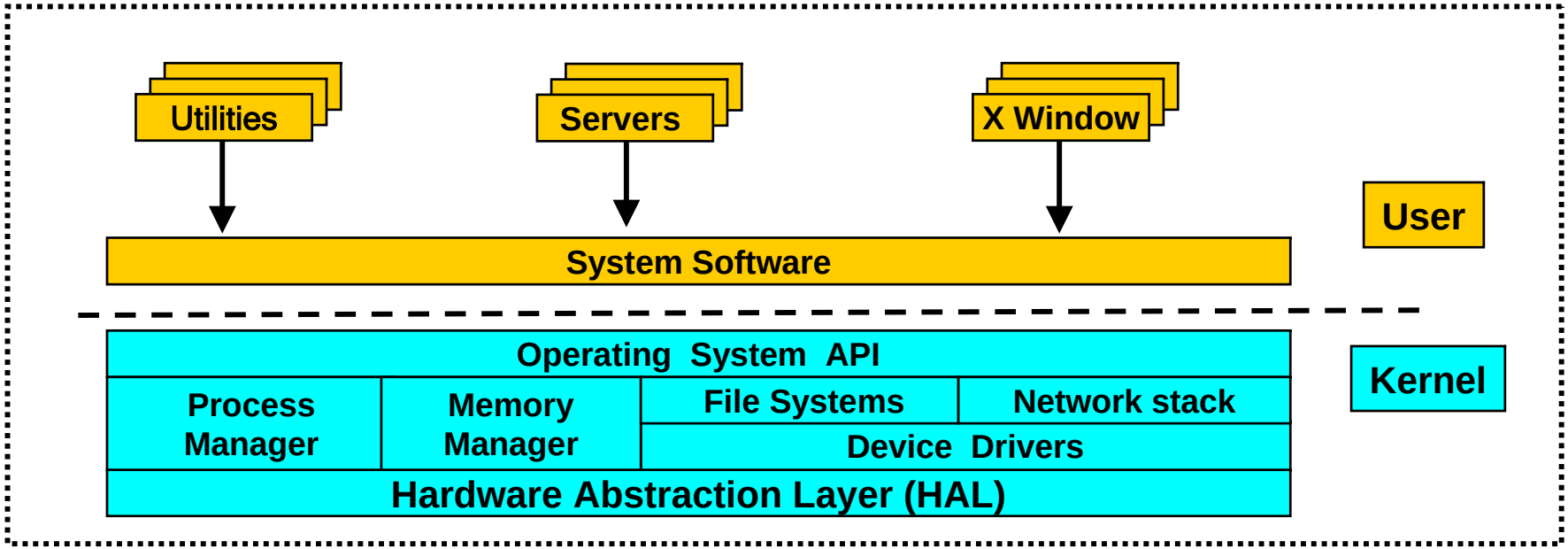
- The process of customizing the Linux Operating System to make it highly secure.

- The first step towards safeguarding systems from intrusion.

Linux Distribution

Combination of kernel , utilities and installer.

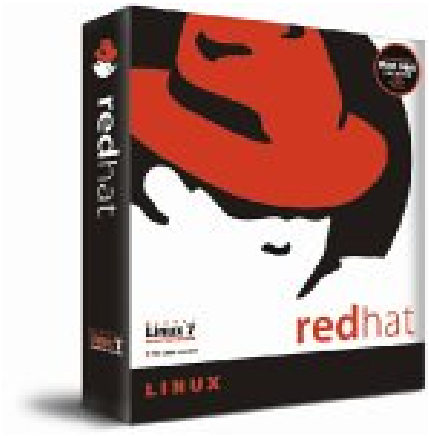
KERNEL + SOFTWARE PACKAGES



- **100+ standard Linux distributions**

Source : http://distrowatch.com

Popular Linux Distributions



Linux Hardening

Setting up Linux on the System

- **Approach :** *Generic use / Special use*

Selecting a distribution
↓
* *Installation*
↓
* *Configuration*
↓
* *Maintenance*

For generic use

* *Designing a distribution*
↓
* *Installation*
↓
* *Configuration*
↓
* *Maintenance*

For special use

* **Hardening can be done.**

STEP : 1

Selecting / Designing a Linux distribution

Selecting A Linux distribution

- **User expertise Level**

Expert : 5 Novice :0

Distribution (Ver)	*Admin*	*Install*	*Support*	*user Rank*
Red Hat (9.0)	***2***	***1***	***0***	***0***
Suse (8.1)	***0***	***1***	***1***	***0***
Mandrake (10.0)	**2**	**0**	**3**	**2**
Lindows (4.5)	**--**	**--**	**--**	**--**
Debian (3.0)	**3**	**3**	**2**	**3**

Can not do hardening ….!

Source: http://www1.ku.edu.tr/files/cit/documents/ComparisonOfLinuxDistributions.pdf

Selecting A Linux distribution

- **Hardware architecture.**

Red Hat (9.0)	*x86, AMD64, IA-64, Alpha, and Sparc*
Suse (8.1)	*x86, AMD64, PowerPC, Sparc, and Alpha*
Mandrake (10.0)	*x86, AMD64, and PowerPC.*
Lindows (4.5)	*x86*
Debian (3.0)	*x86, IA-64, Alpha, 680x0, Sparc, ARM, PowerPC, MIPS, HP PA-RISC, and IBM S/390.*

Can not do hardening ….!

Source: http://mij.oltrelinux.com/discomp/comparison.html

Designing harden Linux distribution

- **Why ?**
  - ✓ Commercial distributions are of general purpose, targeted for wide range of users.
  - ✓ Dump bundle of software on the system.
  - ✓ Do not have quick release time ..! Have to wait for distribution with latest bug fixed, updated kernel and software packages

- **Components ?**
  - ✓ Kernel.
  - ✓ Software Packages.

Designing harden Linux dist. : *Kernel*

- **1. Getting a Kernel**
  - ✓ www.kernel.org
  - ✓ Distribution CD

- **2. Selecting a Kernel**
  - ✓ Stable Version. ***(x. even. x)***
  - ✓ Verify source with md5, PGP key.
  - ✓ Read online review doc.
  - ✓ Proper Testing **:**
    - Code walk through.
    - Relies on third party.

```
# rpm -Kv apache-1.3.22-5.6.i386.rpm
apache-1.3.22-5.6.i386.rpm:
MD5 sum OK: 272ec62194bc45ace491f58c91ffcc9b
gpg: Signature made 20 July 2002, 05:23:45 ICT using DSA
key ID DB42A60E
gpg: Good signature from "Red Hat, Inc
<security@redhat.com>"
Fingerprint: CA20 8686 2BD6 9DFC 65F6 ECC4 2191 80CD
DB42 A60E
```

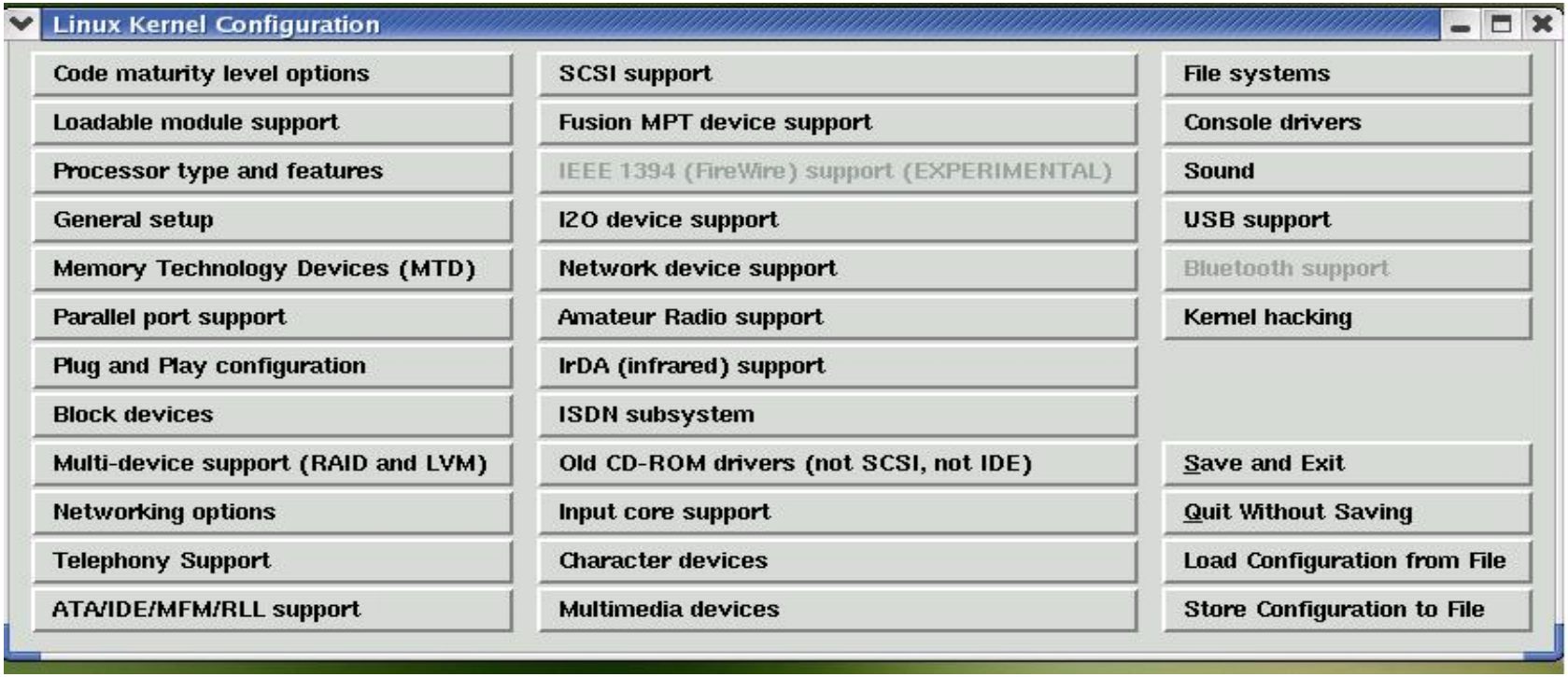
Doc: http://www.gentoo.org/doc/en/gnupg-user.xml

Designing harden Linux dist. : *Kernel*

➢ 3. Configuring a Kernel

- ✓ Turn on security options.
- ✓ Turn off unnecessary options.

Cmd : **make xconfig**



Designing harden Linux dist. : *Kernel*

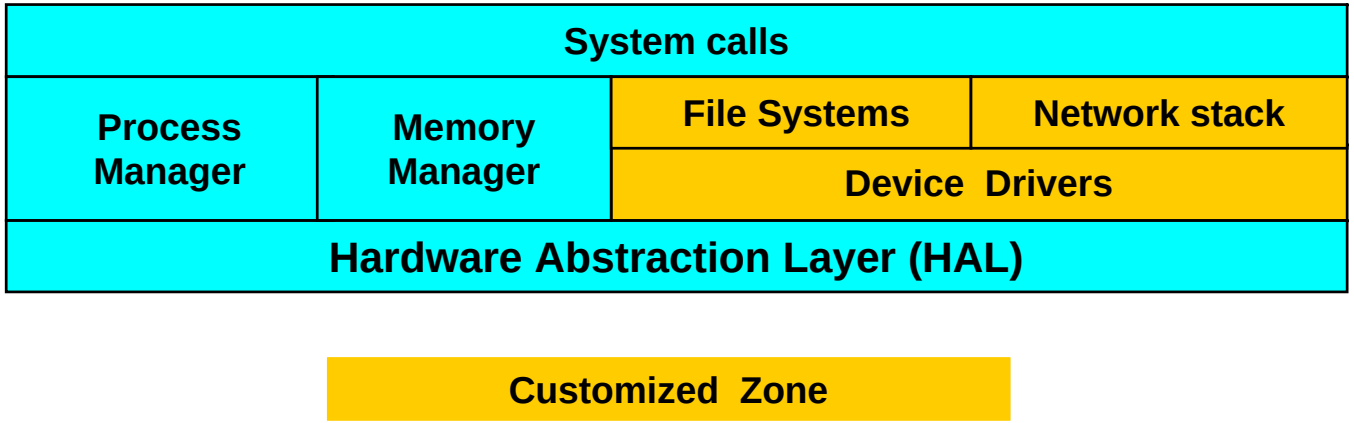
- **Kernel Security Options :**
  - ✓ **Loadable module support : OFF (root kit attack)**
  - ✓ **Network packet filtering : ON (access control)**
  - ✓ **SYN cookies : ON (DOS attack)**
  - ✓ **Advanced Router : OFF**
  - ✓ **Socket Filtering : OFF**
  - ✓ **Crypto API : ON**

- **New (2.6) :**

```
[*] Enable different security models
[ ]   Socket and Networking Security Hooks (NEW)
< >   Default Linux Capabilities (NEW)
< >   Root Plug Support (NEW)
[ ]   NSA SELinux Support (NEW)
```

Designing harden Linux dist. : *Kernel*

- **Turning off unwanted options** : *Kernel customization*



- ✓ Device Driver : Block, RAID,LVM,IDE,SCSI,Network, IrDA,Sound USB etc.
- ✓ File System : Reiser,ADFS, FFS,BFS, MSDOS,VFAT,EFS,JFFS,JFS, MINIX,NTFS,HPFS, NFS,SGI
- ✓ Network Stack : IPX, Apple Talk, DECnet etc.

Designing harden Linux dist. : *Kernel*

➢ **4. Patching Kernel :** Update the old code by applying the difference of the updated code and the old one i.e patch.

- ✓ Add custom updates, vulnerability plugging.
- ✓ Add security enhanced patches (details in last session)
- ✓ Verify patch using md5 or PGP key.
- ✓ Test properly.
- ✓ Command : ***patch –p0 <kernel.patch***

Designing harden Linux dist. : *Kernel*

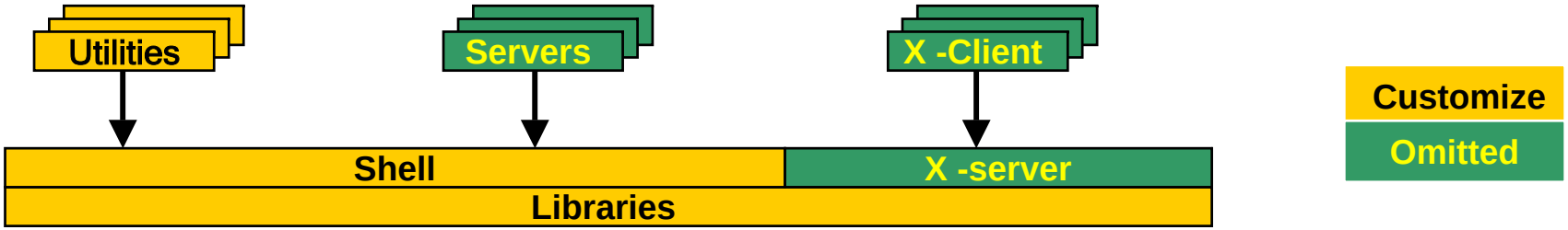
- **4. Compiling Kernel :** Build compressed binary image**.**

  - ✓ Patch the compiler GCC with stack guard option. (Buffer Overflow)
  - ✓ Compile kernel using stack guard option.
  - ✓ Command : ***make dep clean bzImage modules modules_install***

Doc : http://www.linuxgazette.com/issue68/ghosh.html

Designing harden Linux dist. : *Package*

- **1. Getting a Package Source**
  - ✓ www.rpmfind.net/linux/RPM
  - ✓ Package native sites.
  - ✓ www.sourceforge.net
  - ✓ Distribution CD
- **2. Selecting the Package List**
  - ✓ Purpose : Server, Personal Use, Office work, Security Application
  - ✓ Dependencies among packages



Designing harden Linux dist. : *Package*

- **3. Verify the Source**
  - ✓ Verify source with md5, PGP key.
  - ✓ Code walk through.

- **4. Patching the Source**
  - ✓ Add custom updates and vulnerability plugging.
  - ✓ Verify patch using md5 or PGP key.
  - ✓ Test properly.
  - ✓ Command : ***patch –p0 <kernel.patch***

Designing harden Linux dist. : *Package*

- **5. Compiling the Source :** Build binary executable**.**
  - ✓ Patch the compiler GCC with stack guard option. (Buffer Overflow)
  - ✓ Compile the package using stack guard option.
  - ✓ Link the package with library compiled with stack guard option
  - ✓ Use same library version for all packages.

- **6. Building Package :**
  - ✓ Build the package in accordance with the package manager available in the distribution.

Doc : http://redhat.com/docs/books/max-rpm/max-rpm.pdf

http://debian.org/doc/devel-manuals

Designing harden Linux dist.

- **Creating an Installation CD :**
  - ✓ Write an installer or pick from popular distribution.
  - ✓ Verify CD integrity using md5
  - ✓ Password based installation.

Doc : http:// tldp.org/HOWTO/RedHat-CD-HOWTO/

http:// k12linux.mesd.k12.or.us/at/roswell-ltsp.html

http:// 256.com/gray/docs/rh_boot

http:// www.redhat.com/docs/manuals/linux/RHL-9-Manual/custom-guide/

STEP : 2

Hardened Installation

Hardened Installation

- **Disk partitioning :** Separate partition for directories.
  - ✓ **swap** : 2x RAM SIZE
  - ✓ **/** : As small as possible
  - ✓ **/boot** : Small
  - ✓ **/usr** : Depends upon system binary selected
  - ✓ **/tmp** : Small
  - ✓ **/home** : No of user.
  - ✓ **/var** : Large

- **Advantage :** Different mount permission for directories can be set.

Hardened Installation

- **Installation Mode :**
  - ✓ Custom.

- **Package Selection :**
  - ✓ Purpose: Server, Personal Use, Office work, Security Application
  - ✓ No use, no install.

STEP : 3

Hardened Configuration

Hardened Configuration

- **Boot Loader :**
  - ✓ timeout=00 (boot time interaction)
  - ✓ Set Password
- **Services :**
  - ✓ Disable unnecessary services
    - Permanent : edit /etc/inet.d/ directory
    - At boot time : edit rcX.d directory (X : run-level)
    - Temporary : service <service name > stop OR kill pid
  - ✓ Run servers as special user not root.
  - ✓ Follow guideline for secure configuration of individual services.

Hardened Configuration

wu-ftpd - WordPad

File Edit View Insert Format Help

```
# default: on
# description: The wu-ftpd FTP server serves FTP connections. It uses \
#     normal, unencrypted usernames and passwords for authentication.
service ftp
{
      disable     = yes
      socket_type       = stream
      wait              = no
      user              = root
      server                  = /usr/sbin/in.ftpd
      server_args       = -l -a
      log_on_success          += DURATION USERID
      log_on_failure          += USERID
      nice              = 10
}
```

Hardened Configuration

- **Files :**
  - Set proper permission mode of system critical files.
  - Remove bit from root gain program.
  - Remove usual and hidden files.
  - Remove unowned files.

Hardened Configuration

- **Root access :**
  - ✓ Restrict terminal access **:** edit /etc/securetty
  - ✓ Restrict Single User Mode **:** Add ~~:S:wait:/sbin/sulogin in /etc/inittab
  - ✓ Make sure system directories are mentioned first in search PATH.
  - ✓ Make sure that "**.**" is not there in search PATH
  - ✓ No ".rhostfile"
  - ✓ Set auto logout time **:** add TMOUT=<value> in /etc/profile

Hardened Configuration

- **User Configuration :**
  - ✓ Set up the shadow password file (if necessary).
  - ✓ Configure PAM as appropriate for the relevant commands.
  - ✓ Define user account password selection and aging settings.
  - ✓ Set up other default user account restrictions as appropriate (e.g., resource limits).
  - ✓ Plan the system,s group structure if necessary, as well as other similar items like projects.
  - ✓ Set up default user initialization files, in */etc/skel* or elsewhere, as well as the system-wide initialization files.
  - ✓ Remove unneeded predefined accounts.

Hardened Configuration

- **Network configuration :**
  - Configure Firewall
  - Configure IPSec

Hardened Configuration

- **Log configuration :**
  - ✓ Use SWATCH : system watch dog.
  - ✓ Configure syslog
  - ✓ Use Scanlogd

STEP : 4

Maintenance

Audit the System

- **Why :** To know current system status
  - ✓ Vulnerability in the packages.
  - ✓ Configuration correctness

- **How :** Using available tools
  - ✓ Linux security audit tool, TARA
  - ✓ NESSUS , SARA, NMAP, Bassalite

Doc : http:// usat.sourceforge.net

http:// www-arc.com

http:// nessus.org

http:// nmap.org

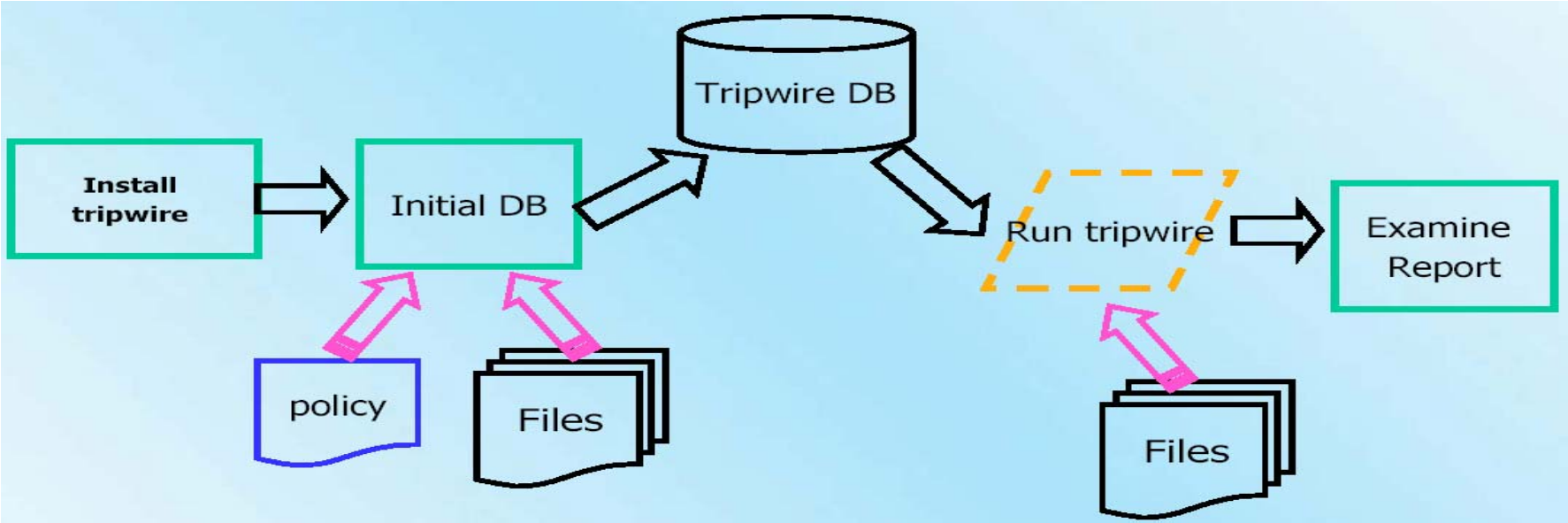
System Update

- **Kernel Update :**
  - ✓ Apply new patches
  - ✓ Apply security patches

- **Packages Update:**
  - ✓ Apply new patches
  - ✓ Apply security patches

System Integrity Check

- **Integrity assessment on files :**
  - ✓ Prevents from warm , trojan horses.
  - ✓ Use Tripware



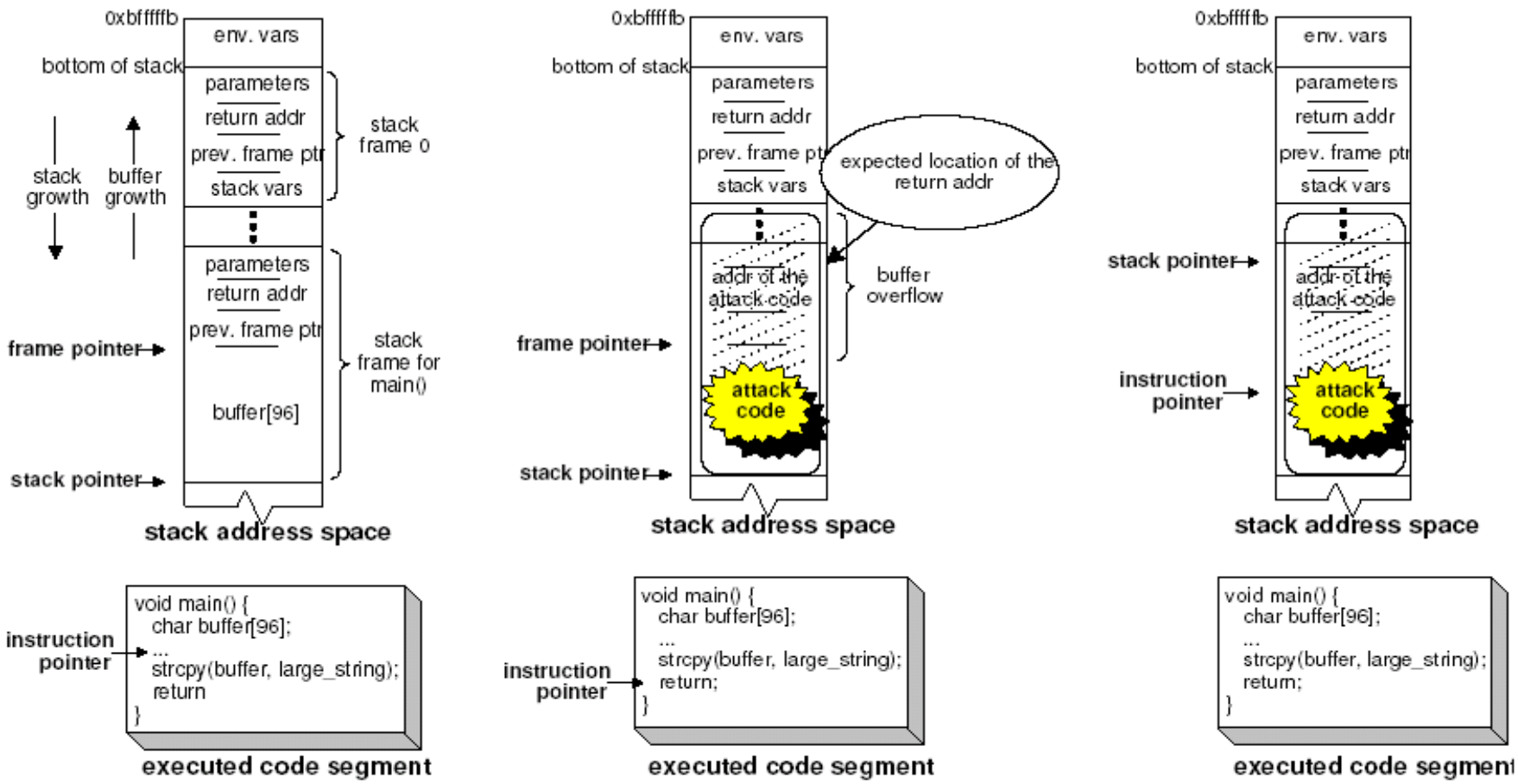
Doc : http:// sourceforge.net

Analysis Log.

- **To know more about the system :**
  - ✓ Boot : /var/log/boot
  - ✓ login : /var/log/auth.log
  - ✓ Security : /var/log/security
  - ✓ Process : /proc/PID

Linux Security Architecture

Buffer overflow



(a) before the attack (b) after injecting the attack code (c) executing the attack code

An attack study

- **Attack :** Buffer overflow vulnerability in sendmail provides root access.
  - ✓ How : Writing an exploit to call execve with “/bin/sh” as arg
  - ✓ Why : Sendmail runs as root. So /bin/sh will also have as root privilege.
  - ✓

```
char shellcode[] =
        "\xeb\x27\x5e\x31\xc0\x88\x46\x07\x88\x46\x0a\x89\x76\x0b\x8d"
        "\x5e\x08\x89\x5e\x0f\x89\x46\x13\xb0\x0b\x89\xf3\x8d\x4e\x0b"
        "\x8d\x56\x13\xcd\x80\xb0\x01\x31\xdb\xcd\x80\xe8\xd4\xff\xff"
        "\xff\x2f\x62\x69\x6e\x2f\x73\x68\x23\x2d\x69\x23\x41\x41\x41"
        "\x41\x42\x42\x42\x42\x43\x43\x43\x43";
```

- **Can not OS security mechanism prevent this ?**
- **What is wrong with the current OS security architecture ?**

Existing Security Architecture

- **Discretionary Access Control (DAC)** : By restricting a subject's (user) access to an object (file).
  - ✓ ls –al / usr/bin/passwd

    r-xs--x--x 1 root root 16192 Aug 14 2002 /usr/bin/passwd.

- **Drawback** :
  - ✓ Subject and object definition is limited to user and file only.
  - ✓ All programs running under a subject have equal ownership over all the objects.

- **How to prevent sendmail problem ?**
  - ✓ Treat sendmail as a subject
  - ✓ Define a security policy to enforce sendmail access over an object.

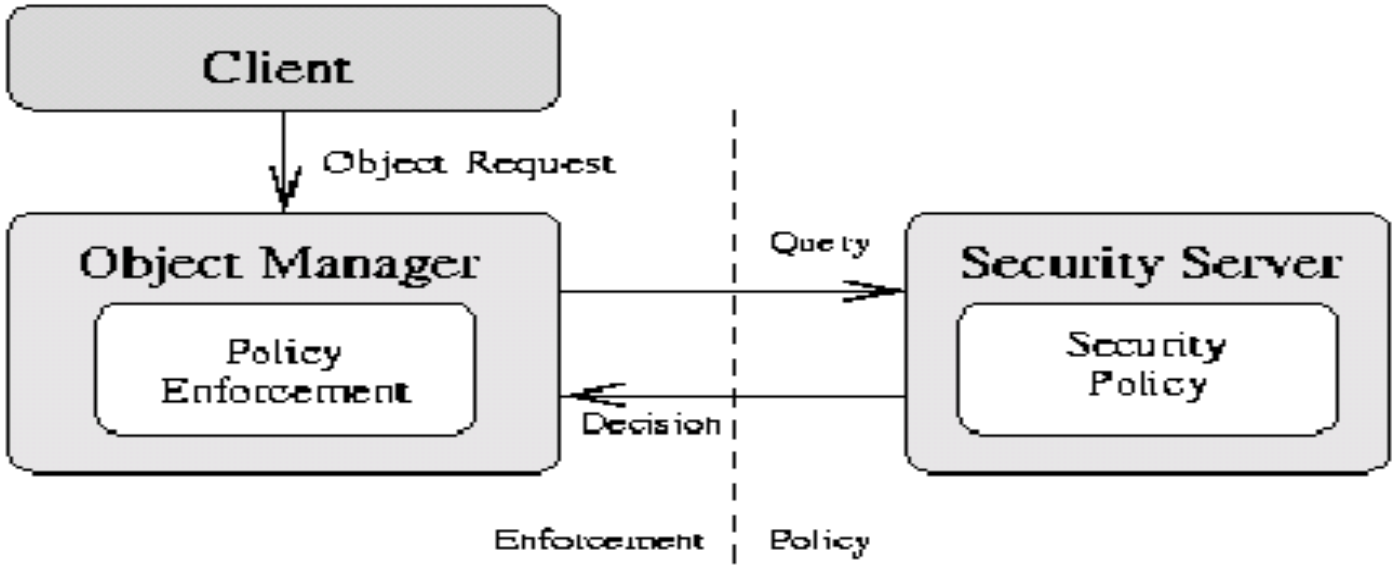
NEW Security Architecture

- **MAC – Mandatory Access Control**
  - ✓ Classify subjects and objects with labels.
  - ✓ How a subject will interact with the object, defines the security policy.

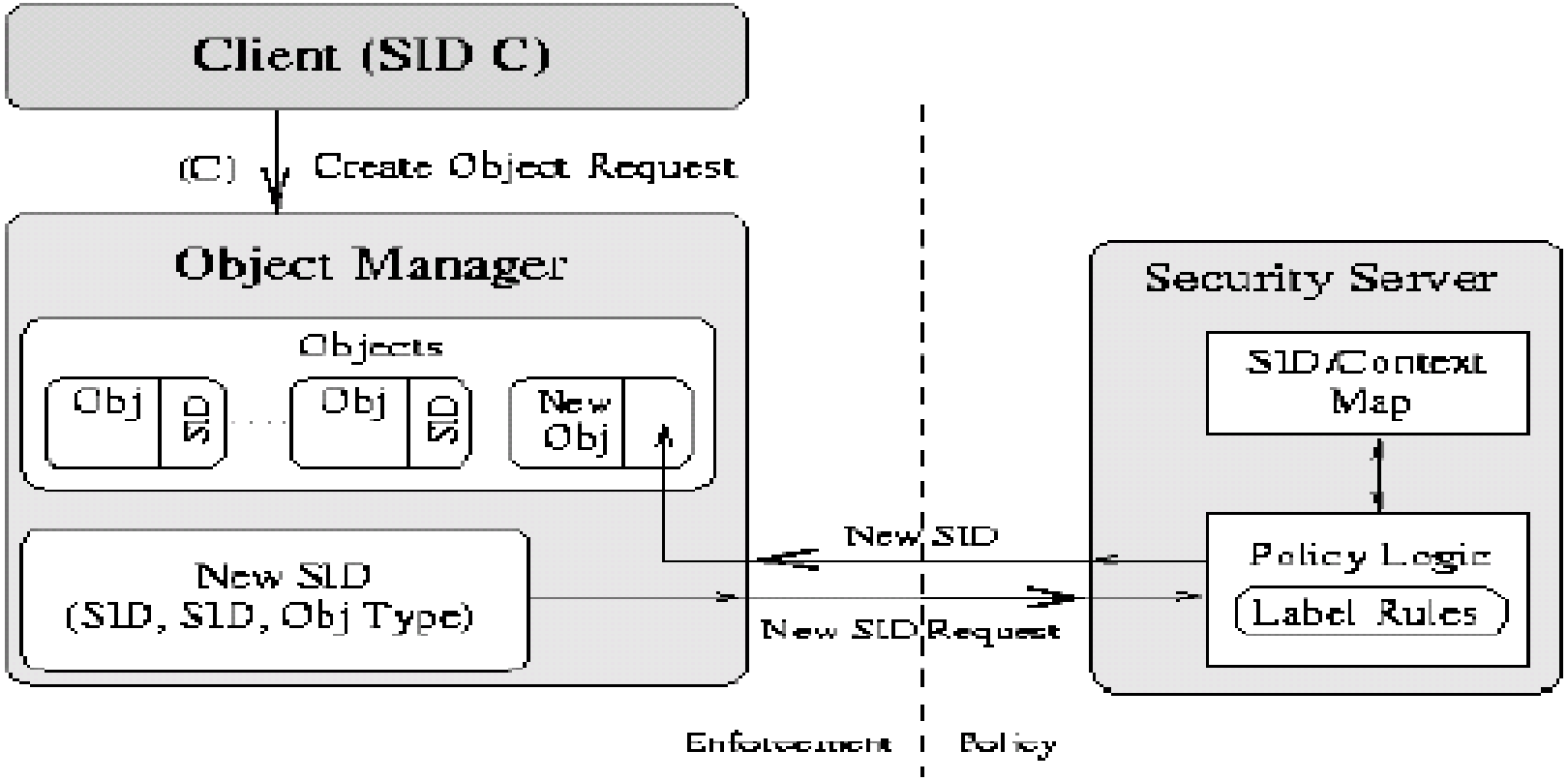
- **MAC –Limitation**
  - ✓ Special trusted subject that act outside the model.
  - ✓ Fails to tightly control the relationship of subjects and the code it execute.

- **Flask Architecture**
  - ✓ Modified MAC to overcome its limitation
  - ✓ Developed by NSA and Secure Computing Corporation

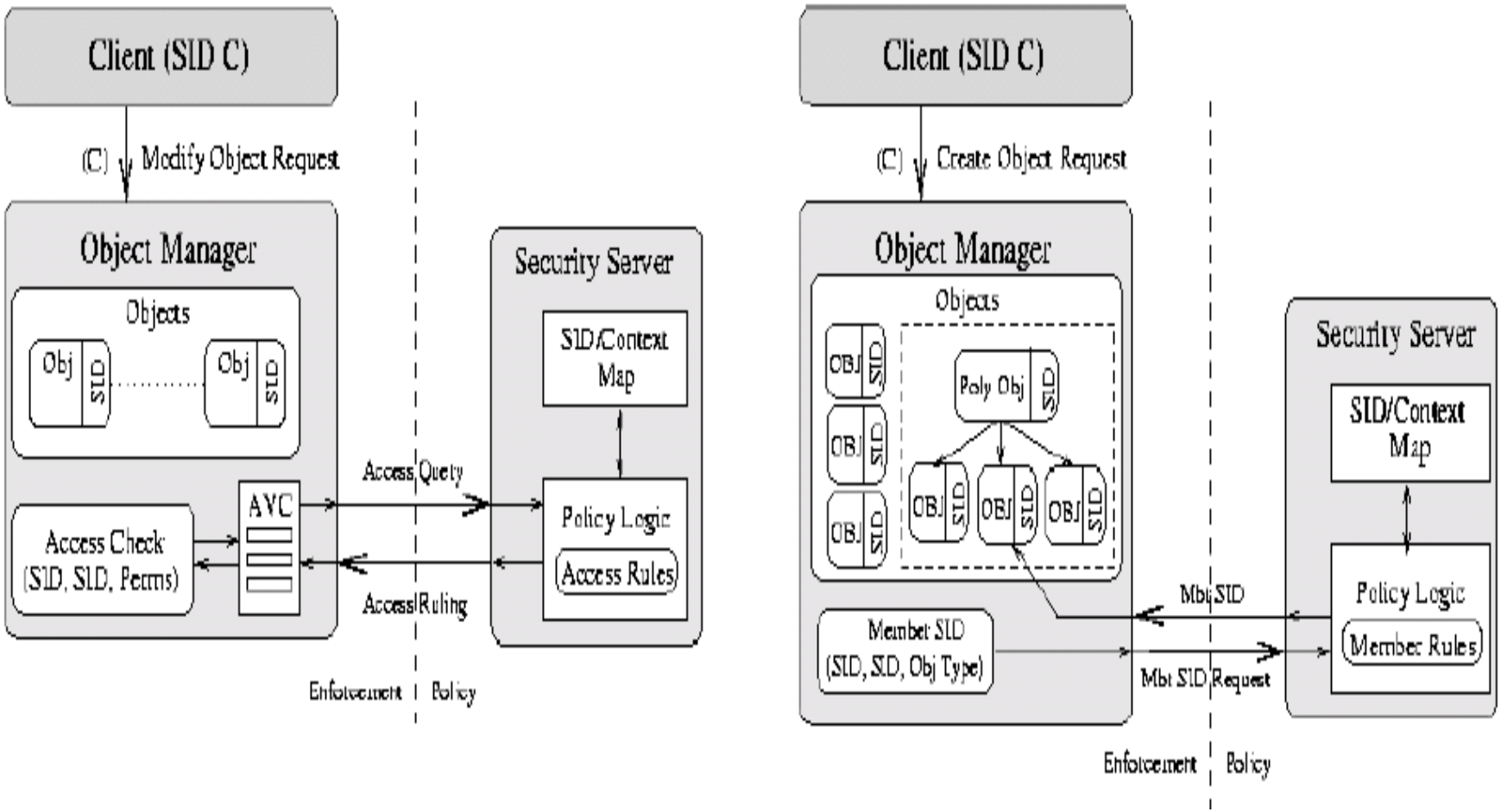
Flask Security Architecture



Flask Security Architecture



Flask Security Architecture



A tour of the Linux Hardening Projects

1. Linux Security Module (LSM)

The Linux Security Modules (LSM) project has developed a lightweight, general purpose, access control framework for the mainstream Linux kernel that enables many different access control models to be implemented as loadable kernel modules.

- **Framework provides :**
  - ✓ It adds opaque security fields to certain kernel data structures
  - ✓ Calls to security hook functions at various points within the kernel code,
  - ✓ Generic security system call
  - ✓ Function for registration and un registration of security modules
  - ✓ Moves most of the capabilities logic into an optional security module

- **Adopted By :** SELinux, DTE, LIDS

URL: http://lsm.immunix.org/

2. SELinux

Security enhanced Linux is a secure Linux distribution form NSA implementing role bases access control mechanism.

- Security enhanced Linux is designed by NSA.
- Realized FLASK architecture.
- Implemented using LSM frame work.
- Available as kernel patch

URL: http://www.nsa.gov/selinux/index.cfm

3. DTE

DTE is an enhanced form of type enforcement, a table-oriented access control mechanism .As with many access control schemes, type enforcement views a system as a collection of active entities (subjects) and a collection of passive entities (objects). In type enforcement, an access control attribute called a domain is associated with each subject (process), and another attribute called a type is associated with each object (file, message, shared memory segment, etc.). A global table, the Domain Definition Table (DDT), represents allowed access modes between domains and types (e.g., read, write, execute), and another table, the Domain Interaction Table (DIT), represents allowed access modes between domains (e.g., signal, create, destroy). As a system runs, access attempts are mediated using table lookups: access attempts for modes not authorized in the tables are denied.

- Realised MAC like access control Mechanism
- Implemented using LSM ramework

URL: http://www.usenix.org/publications/library/proceedings/security95/badger.html

4. LIDS

Linux intrusion detection System is a light weight security patch for protecting files, getting security alerts.

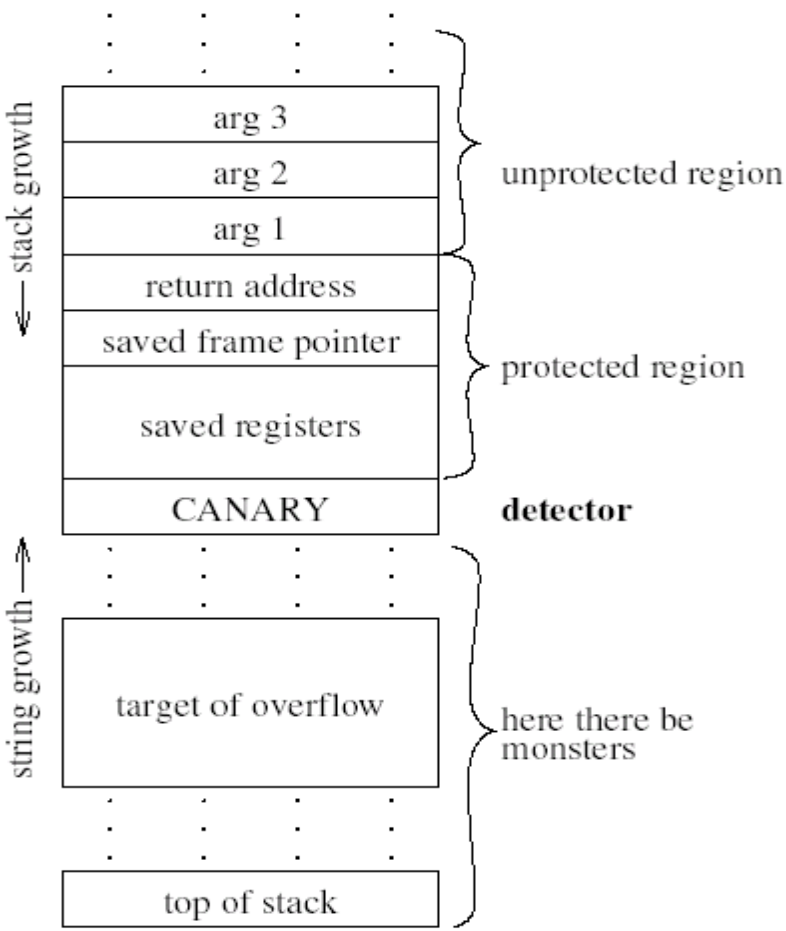
- Implemented using LSM frame work.
- Protection through access control list.
- Available as kernel patch

URL: http://lids.org

5. STACK GUARD

Modify compiler to generate code capable of preventing buffer over flow attacks..

- Overcome Buffer Overflow Problem.
- Use CANARY : fixed, random.



URL: http://immunix.org

Thank You

Biswajit Paul

(biswajit@cair.res.in)