PROJECT REPORT
ON

LINUX-BASED NETWORK INTRUSION DETECTION SYSTEM

SUBMITTED BY

DEBAJIT ADHIKARY
AMIT GUPTA
MIHIR DESAI
TARUN GOEL

2003 – 2004

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 PROBLEM DEFINITION

Implementation of Linux-based Network Intrusion Detection System.

BRIEF DESCRIPTION OF THE TOPIC

An intrusion is somebody attempting to break into or misuse your system. The word "misuse" is broad, and can reflect something severe as stealing confidential data to something minor such as misusing your email system for spam.

An "Intrusion Detection System (IDS)" is a system for detecting such intrusions.

Network intrusion detection systems (NIDS) monitors packets on the network wire and attempts to discover if a hacker/cracker is attempting to break into a system (or cause a denial of service attack). A typical example is a system that watches for large number of TCP connection requests (SYN) to many different ports on a target machine, thus discovering if someone is attempting a TCP port scan. A NIDS may run either on the target machine who watches its own traffic (usually integrated with the stack and services themselves), or on an independent machine promiscuously watching all network traffic (hub, router, probe). Note that a "network" IDS monitors many machines, whereas the others monitor only a single machine (the one they are installed on).

1.2 BACKGROUND / RELEVANCE

IDS AND FIREWALLS

A common misunderstanding is that firewalls recognize attacks and block them. This is not true. Firewalls are simply a device that shuts off everything, then turns back on only a few well-chosen items. In a perfect world, systems would already be "locked down" and secure, and firewalls would be unneeded. The reason we have firewalls is precisely because security holes are left open accidentally.

Thus, when installing a firewall, the first thing it does is stops ALL communication. The firewall administrator then carefully adds "rules" that allow specific types of traffic to go through the firewall. For example, a typical corporate firewall allowing access to the Internet would stop all UDP and ICMP datagram traffic, stops incoming TCP connections, but allows outgoing TCP connections. This stops all incoming connections from Internet hackers, but still allows internal users to connect in the outgoing direction.

A firewall is simply a fence around you network, with a couple of well chosen gates. A fence has no capability of detecting somebody trying to break in (such as digging a hole underneath it), nor does a fence know if somebody coming through the gate is allowed in. It simply restricts access to the designated points.

In summary, a firewall is not the dynamic defensive system that users imagine it to be. In contrast, an IDS is much more of that dynamic system. An IDS does recognize attacks against the network that firewalls are unable to see.

Another problem with firewalls is that they are only at the boundary to your network. Roughly 80% of all financial losses due to hacking come from inside the network. A firewall a the perimeter of the network sees nothing going on inside; it only sees that traffic which passes between the internal network and the Internet.

Some reasons for adding IDS to you firewall are:

- Double-checks misconfigured firewalls.
- Catches attacks that firewalls legitimate allow through (such as attacks against web servers).
- Catches attempts that fail.
- Catches insider hacking.

RELEVANCE

As we know that Linux is an open source operating system, network security is a vital issue. Infact if we discus the sometimes overwhelming amount of security required in managing a network, we can quickly come to the conclusion that we can devote nearly all of our time to network and system security and never have a shortage of projects.

One member of our team made the comment that in that case, we would
"…have a soccer team full of goalies!" So, the questions of where we draw the line in the sand with respect to security goes on.

IDS (Intrusion Detection System) could be pertinent solution to the Linux security matters.

To appreciate the actual relevance of IDS we should know what an ideal IDS is i.e. what all things it is supposed to do.

Intrusion detection systems are an emerging new technology when talking about network security. More and more organizations realize the importance of a capable system that detects and alerts on the occurrence of an intrusion. We realize that being informed is the best weapon in the security analyst's arsenal. On the other hand, it will help to keep most vendors honest about their products and services.

What IDS stands for is self-explanatory. A system that detects any intrusion attempts or attacks and hopefully notifying the administrator in the best possible way. Here there is another problem to be solved. Getting an IDS would be easy but the problem is to know what are the IDEAL IDS to use.

Following explications manifest the diligent effort of the group members to develop an IDEAL IDS.

(1) Supernumerary intrusion detection capabilities in Linux Operating system are amalgamated into a single product.

Normally, an IDS is neither a firewall nor a router. In our product we have included firewalling as well as routing interfaces. Along with these facilities, added features like port scanning and blocking, DNS zone protection are also incorporated in this product to make it consummate.

(2) Vague interfaces of the already available IDSes often lead to a misleading construe, creating requisite conversancy of the Linux System Administrator snatching the handy behavior of the product.

GUI buttons with unambiguous construe adorns the interface leading no misgivings of conscience.

Conspicuous interfaces of our IDS make even a novice user to handle the product flawlessly.

(3) Remote manageability is also an important requirement of products today so that products can be managed from anywhere in the world. This product provides a remote management console based on an Apache back end.

4) Automatic backup and Auto-recovery of critical services on intrusion will be an added feature.

(5) Features like logs & user authentication are added to make the product even more secure.

1.3 LIST OF KEYWORDS

1. IDS (Intrusion Detection System)
2. Tripwire
3. File System Integrity Check
4. Network Security
5. Firewall
6. IPtables
7. Port Scanning
8. DNS Zone Protection
9. Intrusion Prevention
10. Backup
11. Logs
12. File Tampering
13. Hacking
14. User Authentication
15. Network Adapter Protection
16. Remote Management Console
17. PHP
18. Auto-Recovery
19. Apache back end
20. Linux system security

1.4 SCOPE OF THE PROJECT

(1) Linux is an open source free operating system with a wealth of intrusion detection capabilities. These are scattered across different products.

The keystone of this project will be to develop a product with all these capabilities inherently built into it.

(2) The lack of a well-defined intuitive interface in such products requires the network administrator to have a prior working knowledge of Linux network administration, narrowing the widespread use of such systems.

This project aims to rectify this scenario by providing an intuitive PHP-based web-interface.

(3) Remote manageability is also an important requirement of products today so that products can be managed from anywhere in the world. This project aims to provide a remote management console based on an Apache back end.

(4) It will be inherently capable of acting as a firewall.

(5) Backup and restoration of critical files.

1.5 APPLICATIONS

1. ANOMALY DETECTION

The goal of anomaly detection is to analyze the network or system and infer what is normal and what is not. It then applies statistical or measures to subsequent events and determines whether they match the model of statistic of "normal". After determining what is normal and what events are outside of a probability window of normal, it will then generate reports and alerts. Under this structure, it gives a more tunable control of false positives.

A typical anomaly detection approach would be:

(1) a neural network
which is based on probability patterns recognition;

(2) a statistical analysis
which is a modeling behavior of users and
looking for deviations from the norms;

(3) a state change analysis
which is modeling system's state and
looking for deviations from the norms.

If it works, it might conceivably catch any possible attacks and attacks that we haven't seen before. We may be able also catch and determine attacks that are close variants to a previously known attack. A more valued advantage is that it will not require administrator to constantly keeping up on hacking techniques.

2. MISUSE DETECTION

The goal of misuse detection is that it must know what constitute an attack and then detect it. A typical misuse detection approaches would be something like a “network grep”, which looks for strings in network connections which might indicate an attack is in progress. After the strings were found, it then makes a pattern matching in which it encodes siries of states that are passed through during the course of an attack. Below is a simpler example of misuse detection:

“Change ownership of /etc/passwd” _ “open /etc/passwd for write” _ ALERT A good side of this detection method is that it is easy to implement, ease of deployment, easy to update and understand, very fast and with low false positives.

A darker side of it is that it cannot detect something, which are previously unknown. It requires constant updates with the new rules and off course easier to fool.

Examples of misuse detection products are:

(1) ISS RealSecure
(2) Cisco Netranger
(3) NAI CyberCop
(4) NFR Network Flight Recorder.

The base model of the deployment is to feed rule sets to customer as a subscription service. What usually a misuse detection looks for are things like IP Fragments Attacks, Source Routing, ISS Scan check, Rwhod check, Rlogin –froot, Imap buffer smash, Ping flooding, Ping of Death, SATAN Scan checks, Rlogin decode, TFTP get passwd check etc.

The concept is somewhat similar to a virus scanning system. Some of the similarities are:

(1) Both rely on meta-rules of vulnerabilties

(2) Both need frequent updates of rules

(3) Both are easily fooled by slight mutations in virus or attack signatures

(4) Both are fairly log in generating false positives.

3. BURGLAR ALARMS

Burglar alarms is based move on site policy that alerts an administrator to a possible policy violation. What can be emphasized on this is that the goal is to detect events that may not be a “security” event but more to be an indication of a policy violation such as new routers, subnets or web servers. It is a misuse detection system that is carefully targeted on certain policy. An example of this is an administrator may not care about people portscanning your firewall from the outside BUT the administrator may care profoundly about people port-scanning your mainframe from the inside. What an administrator can do is to set up a misuse detector to watch for misuses such as internal port scanning to the mainframe (among other things) violating the policy.

1.6 SYSTEM REQUIREMENT SPECIFICATIONS

SOURCE-BASED FIREWALLING RULE

- Administrator opens browser
- Access the management console
- Authenticate the session
- Accesses the firewall screen
- Enters the source to be blocked or allowed access.
- Submits
- System confirms that the rule has been added.

These use cases can be modified e.g. first three actions in the above use case can be combined to form one action as pre-condition – logged on and valid session.

SOURCE AND DESTINATION BASED FIREWALLING RULE

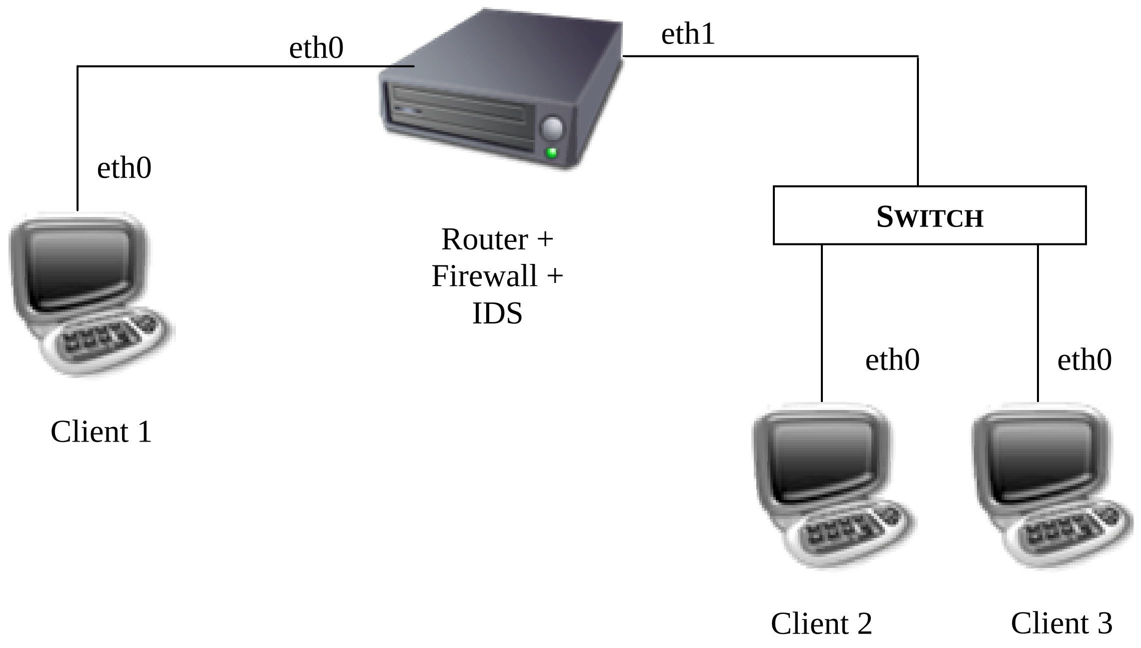
- Administrator opens browser
- Access the management console
- Authenticate the session
- Accesses the firewall screen
- Enters the source and the destination to be blocked or allowed access.
- Submits
- System confirms that the rule has been added.

Similar specifications can be defined for other use cases. Though the actions in all these use cases are apparently redundant, the functionality of each use case is unique.

1.7 REQUIREMENTS

REQUIREMENT SPECIFICATIONS

(1) Network with minimum 3 PC's
(2) Switch 100Mbps
(3) Ethernet cards 10/100Mbps
(4) UTP CAT-5 Cable
(5) RJ-45 Connectors



CHAPTER 2

CURRENT THEORY AND PRACTICES

2.1 TRIPWIRE

INTRODUCTION

Tripwire data integrity assurance software monitors the reliability of critical system files and directories by identifying changes made to them. It does this through an automated verification regimen run at regular intervals. If Tripwire detects that a monitored file has been changed, it notifies the system administrator via email. Because Tripwire can positively identify files that have been added, modified, or deleted, it can speed recovery from a break-in by keeping the number of files which must be restored to a minimum. These abilities make Tripwire an excellent tool for system administrators seeking both intrusion detection and damage assessment for their servers.

Tripwire works by comparing files and directories against a database of file locations, dates they were modified, and other data. This database contains baselines — which are snapshots of specified files and directories at a specific point in time. The contents of the baseline database should be generated before the system is at risk of intrusion, meaning before it is connected to the network. After creating the baseline database, Tripwire compares the current system to the baseline and reports any modifications, additions, or deletions.

FUNCTIONAL OVERVIEW OF TRIPWIRE

The Tripwire utility is actually quite straightforward in what it does and how it is configured. When invoked, Tripwire reads a configuration file and a policy file. The configuration file tells it about the location of Tripwire files, and the policy file specifies which files and directories to pay attention to, and what information to gather about those directories and files.

Once the information is gathered by running Tripwire in the initialize mode, it is placed in the database, which is then encrypted. Whenever Tripwire is run again, it uses the information it finds in the database and compares it to the information that is actually on the system. This is known as compare mode. If it finds and discrepancies (changes), it reports them to the system administrator. All of these changes should be checked out thoroughly. If unauthorized activity is identified, the administrator will need to recover the system. If the changes are a result of authorized activity, the administrator has the option of updating the database so that future Tripwire compare runs do not report the same differences. Updating is accomplished though Tripwire's update mode. Figure 2 illustrates the overall functionality of Tripwire.

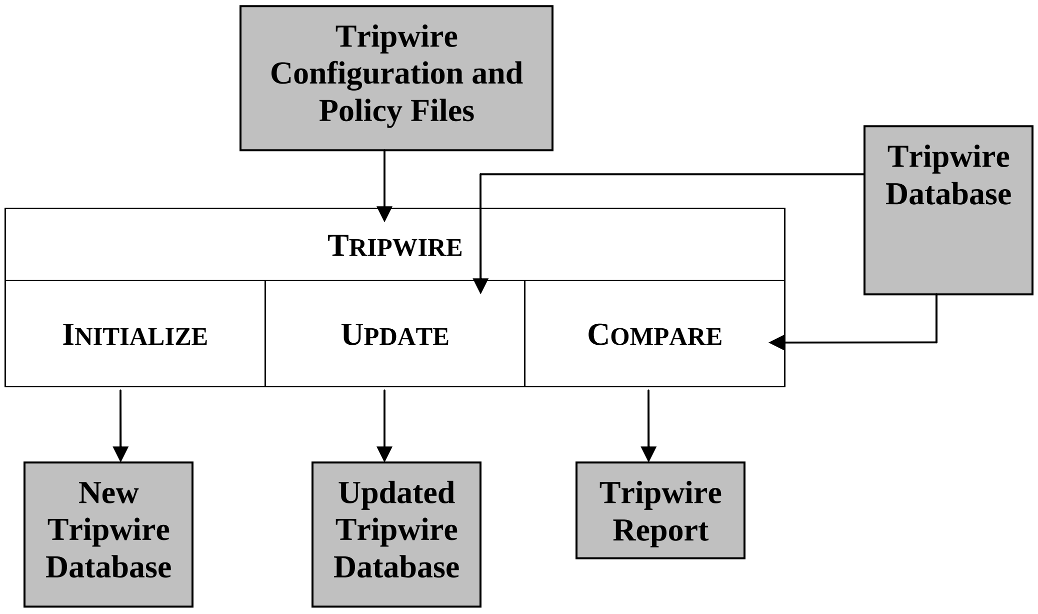


FIGURE 2: FUNCTIONAL OVERVIEW OF TRIPWIRE

HOW TRIPWIRE WORKS

The following flowchart illustrates how Tripwire works:

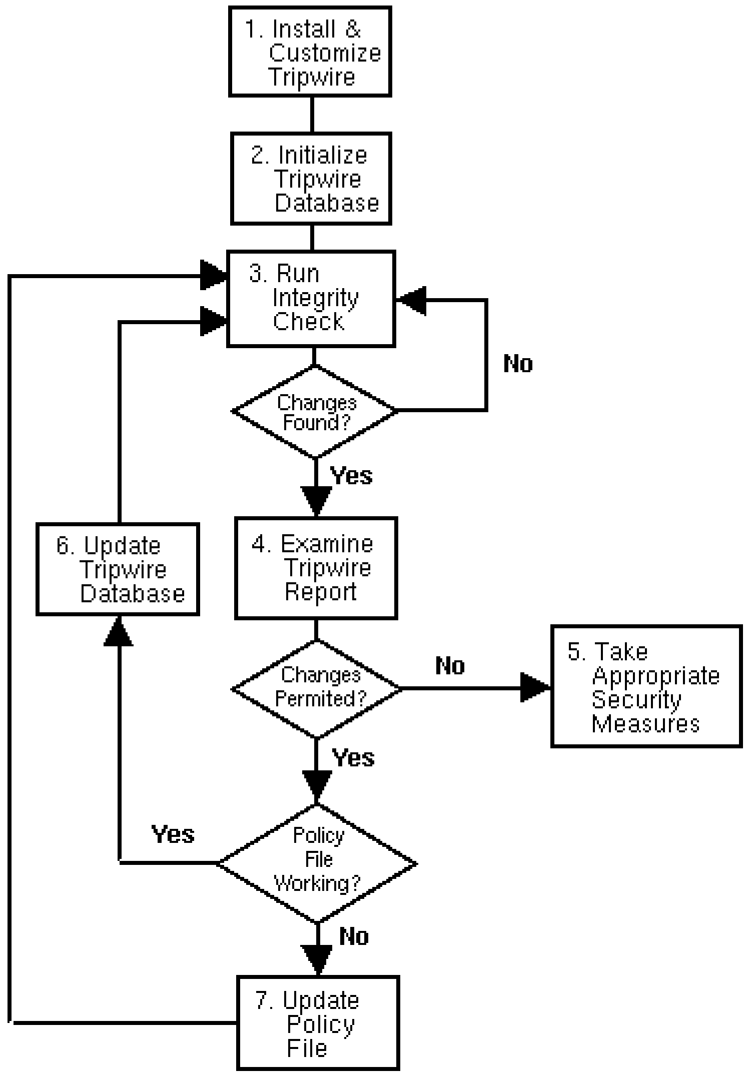

1. Install & Customize Tripwire
2. Initialize Tripwire Database
3. Run Integrity Check
Changes Found?
No
Yes
4. Examine Tripwire Report
Changes Permited?
No
5. Take Appropriate Security Measures
Yes
Policy File Working?
Yes
6. Update Tripwire Database
No
7. Update Policy File

FIGURE 1: USING TRIPWIRE

The following describes in more detail the numbered blocks shown in Figure 1

1. INSTALL TRIPWIRE AND CUSTOMIZE THE POLICY FILE.

Install the Tripwire RPM.. Then, customize the sample configuration and policy files (`/etc/tripwire/twcfg.txt` and `/etc/tripwire/twpol.txt` respectively), and run the configuration script, `/etc/tripwire/twinstall.sh`.

2. INITIALIZE THE TRIPWIRE DATABASE.

Build a database of critical system files to monitor based on the contents of the new, signed Tripwire policy file, `/etc/tripwire/tw.pol`.

3. RUN A TRIPWIRE INTEGRITY CHECK.

Compare the newly-created Tripwire database with the actual system files, looking for missing or altered files.

4. EXAMINE THE TRIPWIRE REPORT FILE.

View the Tripwire report file using `/usr/sbin/twprint` to note integrity violations.

5. IF UNAUTHORIZED INTEGRITY VIOLATIONS OCCUR, TAKE APPROPRIATE SECURITY MEASURES.

If monitored files have been altered inappropriately, you can either replace the original files from backup copies, reinstall the program, or completely reinstall the operating system.

6. IF THE FILE ALTERATIONS ARE VALID, VERIFY AND UPDATE THE TRIPWIRE DATABASE FILE.

If the changes made to monitored files are intentional, edit Tripwire's database file to ignore those changes in subsequent reports.

7. IF THE POLICY FILE FAILS VERIFICATION, UPDATE THE TRIPWIRE POLICY FILE.

To change the list of files Tripwire monitors or how it treats integrity violations, update the supplied policy file (`/etc/tripwire/twpol.txt`), regenerate a signed copy (`/etc/tripwire/tw.pol`), and update the Tripwire database.

OVERVIEW OF TRIPWIRE BINARIES

SCRIPT	DESCRIPTION
/usr/sbin/tripwire	The main Tripwire binary. This is used • To create the initial online database of information • To perform integrity checks of the system • To reflect authorized system changes into the online database. • To update the policy • To test the Tripwire email functionality
/usr/sbin/twadmin	Facilitates • Management of configuration and policy files • Management of local and site keys • Encryption of files
/ust/sbin/twprint	Provides a way to print Tripwire databases and reports
/etc/tripwire/tw.cfg	The Tripwire configuration file. Specifies location of Tripwire files and information needed to email reports. This is the encrypted version of the file, and the site key is needed to make changes. An unencrypted copy may be found in /etc/tripwire/twcfg.txt
/etc/tripwire/twcfg.txt	Text version of the configuration file. Not meant to be permanently available
/etc/tripwire/tw.pol	The Tripwire policy file. This is a collection of rules specifying which files and directories should be monitored, and what attributes of those targets should be watched for changes. This file is encrypted with the site key
/etc/tripwire/twpol.txt	A temporary, plaintext copy of the policy file

CONFIGURING TRIPWIRE

After the Tripwire binaries have been installed, the administrator needs to create the site and the local keys that will be used to encrypt the database, policy and configuration files, and the Tripwire reports, and to customize the policy and configuration files. The usual steps are outlined in the table below.

STEP	PURPOSE
/usr/sbin/tripwire --init	Create the baseline database.
/usr/sbin/tripwire --check	Run in integrity checking mode. Changes reports to standard output.
modify /etc/tripwire/twcfg.txt	Update configuration file.
modify /etc/tripwire/twpol.txt	Update policy file.

The above steps are briefly described below:

(1) INITIALIZING THE TRIPWIRE DATABASE

When initializing its database, Tripwire builds a collection of file system objects based on the rules in the policy file. This database serves as the baseline for integrity checks.

To initialize the Tripwire database, use the following command:

```
/usr/sbin/tripwire --init
```

Once you finish these steps successfully, Tripwire has the baseline snapshot of your file system necessary to check for changes in critical files. After initializing the Tripwire database, you should run an initial integrity check. This check should be done prior to connecting the computer to the network and putting it into production.

(2) RUNNING AN INTEGRITY CHECK

By default, the Tripwire RPM adds a shell script called `tripwire-check` to the `/etc/cron.daily/` directory. This script automatically runs an integrity check once per day.

You can, however, run a Tripwire integrity check at any time by typing the following command:

```
/usr/sbin/tripwire --check
```

During an integrity check, Tripwire compares the current state of file system objects with the properties recorded in its database. Violations are printed to the screen and an encrypted copy of the report is created in `/var/lib/tripwire/report/`.

(3) UPDATING THE TRIPWIRE CONFIGURATION FILE

If you want to change Tripwire's configuration file, you should first edit the sample configuration file `/etc/tripwire/twcfg.txt`. If you deleted this file (as you should whenever you are finished configuring Tripwire), you can regenerate it by issuing the following command:

```
twadmin --print-cfgfile > /etc/tripwire/twcfg.txt
```

Tripwire will not recognize any configuration changes until the configuration text file is correctly signed and converted to `/etc/tripwire/tw.pol` with the `twadmin` command.

Use the following command to regenerate a configuration file from the `/etc/tripwire/twcfg.txt` text file:

```
/usr/sbin/twadmin --create-cfgfile -S site.key
/etc/tripwire/twcfg.txt
```

Since the configuration file does not not alter any Tripwire policies or files tracked by the application, it is not necessary to regenerate the Tripwire database.

(4) UPDATING THE TRIPWIRE POLICY FILE

If you want to change the files Tripwire records in its database, change email configuration, or modify the severity at which certain violations are reported, you need to edit your Tripwire policy file.

First, make whatever changes are necessary to the sample policy file `/etc/tripwire/twpol.txt`. If you deleted this file (as you should whenever you are finished configuring Tripwire), you can regenerate it by issuing the following command:

```
twadmin --print-polfile > /etc/tripwire/twpol.txt
```

A common change to this policy file is to comment out any files that do not exist on your system so that they will not generate a `file not found` error in your Tripwire reports. For example, if your system does not have a `/etc/smb.conf` file, you can tell Tripwire not to try to look for it by commenting out its line in `twpol.txt` with the `#` character as in the following example:

```
#     /etc/smb.conf     -> $(SEC_CONFIG) ;
```

Next, you must generate a new, signed `/etc/tripwire/tw.pol` file and generate an updated database file based on this policy information. Assuming `/etc/tripwire/twpol.txt` is the edited policy file, use this command:

```
/usr/sbin/twadmin --create-polfile -S site.key
/etc/tripwire/twpol.txt
```

You will be asked for the site password. Then, the `twpol.txt` file will be encrypted and signed.

It is important that you update the Tripwire database after creating a new `/etc/tripwire/tw.pol` file. The most reliable way to accomplish this is to delete your current Tripwire database and create a new database using the new policy file.

If your Tripwire database file is named `debajit.ids.com.twd`, type this command:

```
rm /var/lib/tripwire/debajit.ids.com.twd
```

Then type the following command to create a new database using the updated policy file:

```
/usr/sbin/tripwire –init
```

To make sure the database was correctly changed, run the first integrity check manually and view the contents of the resulting report.

THE TRIPWIRE POLICY FILE

In the Tripwire policy file, we will find a listing of files and directories that we wish to fingerprint. The fingerprint is simply a collection of information about the file or directory, like file size, permission settings, inode number. It may also include hash values from one or more one-way hash functions. Associated with each file or directory we will find a collection of property masks that determines the information to be collected for the fingerprint. The administrator will need to customize the policy file in order to ensure the right files are fingerprinted.

The elements in a policy file fall into one of four categories: comments, rules, variables and directives. Each of these categories is described below.

(1) COMMENTS

All text following a '#' for the remainder of the line is considered to be a comment and will be ignored by Tripwire.

(2) RULES

Rules denote what objects on a system should be checked, and specify the attributes associated with that object that should be monitored for change. Rules can also direct Tripwire not to check certain objects. There are two kinds of rules: ***normal rules*** and ***stop rules.***

NORMAL RULES

FORMAT OF A "NORMAL" POLICY RULE

object_name -> property_mask ;

OBJECT NAME

An object_name refers to a file, directory, or device that begins with a leading "/". An object can be represented by a policy file variable. If the object is a directory, the directory itself along with all the objects contained in the directory will be monitored.

PROPERTY MASK

The property_mask represents a collection of information about an object and is used to create a fingerprint of the object.

TRIPWIRE PROPERTY MASKS	
SELECTION MASK	**DESCRIPTION**
p	Permission and file mode bits
i	Inode number
n	Link count
u	UID
s	File size
g	GID
r	ID of device pointed to by inode
l	File increased in size
a	Access timestamp
m	Modification timestamp
c	Inode change timestamp
t	File type
d	Inode storage disk device number
b	Number of blocks
C	CRC-32. Not useful for security purposes
M	MD5 (Message-Digest Algorithm)
S	SHA-1 (NIST Secure Hash Algorithm)
H	Haval (A 128-bit signature algorithm)
+*mask*	Includes the *mask* in the fingerprint. For example +*pin* means to include permission bits, inode number and link count.
ReadOnly	+*pinugtsbmCM-rlacSH* (read-only template, for read-only files and directories)
Dynamic	+*pinugtd-srlbamcCMSH* (intended for files and directories that are dynamic in behaviour, such as home directories)
Growing	+*pinugtdl-srbamcCMSH* (Intended for files that should always grow)
Device	+*pugsdr-intlbamcCMSH* (Intended for objects that Tripwire should not try to open, such as device files)
IgnoreAll	*-pinugtsdrlbamcCMSH*

	(Permits you to monitor presence or absence of an object, but no other attributes)
IgnoreNone	*+pinugtsdrbamcCMSH-l* (Turns on checking of all attributes)

STOP POINT RULES

A rule that directs Tripwire to not scan an object is known as a "Stop Point" rule.

FORMAT OF A STOP RULE

! object_name;

EXAMINING TRIPWIRE REPORTS

The `/usr/sbin/twprint` command is used to view encrypted Tripwire reports and databases.

VIEWING TRIPWIRE REPORTS

The `twprint -m r` command will display the contents of a Tripwire report in clear text. You must, however, tell `twprint` which report file to display.

A `twprint` command for printing `Tripwire` reports looks similar to the following:

```
/usr/sbin/twprint -m r --twrfile
/var/lib/tripwire/report/<name>.twr
```

The `-m r` option in the command directs `twprint to decode a Tripwire report. The --twrfile option directs twprint to use a specific` Tripwire report file.

The name of the Tripwire report that you want to see includes the name of the host that Tripwire checked to generate the report, plus the creation date and time. You can review previously saved reports at any time. Simply type ls /var/lib/tripwire/report to see a list of Tripwire reports.

Tripwire reports can be rather lengthy, depending upon the number of violations found or errors generated. A sample report starts off like this:

```
Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:          root
Report created on:            Fri Jan 12 04:04:42 2001
Database last updated on:     Tue Jan  9 16:19:34 2001

=======================================================================
Report Summary:
=======================================================================
Host name:                    some.host.com
Host IP address:              10.0.0.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/some.host.com.twd
Command line used:            /usr/sbin/tripwire --check

=======================================================================
Rule Summary:
=======================================================================
-----------------------------------------------------------------------
Section: Unix File System
-----------------------------------------------------------------------
  Rule Name                 Severity Level    Added    Removed  Modified
  ---------                 --------------    -----    -------  --------
  Invariant Directories     69                0        0        0
  Temporary directories     33                0        0        0
* Tripwire Data Files       100               1        0        0
  Critical devices          100               0        0        0
  User binaries             69                0        0        0
  Tripwire Binaries         100               0        0        0
```

VIEW TRIPWIRE DATABASES

You can also use `twprint` to view the entire database or information about selected files in the Tripwire database. This is useful for seeing just how much information Tripwire is tracking on your system.

To view the entire Tripwire database, type this command:

```
/usr/sbin/twprint -m d --print-dbfile | less
```

This command will generate a large amount of output, with the first few lines appearing similar to this:

```
Tripwire(R) 2.3.0 Database

Database generated by:        root
Database generated on:        Tue Jan  9 13:56:42 2001
Database last updated on:     Tue Jan  9 16:19:34 2001

=================================================================
Database Summary:
=================================================================
Host name:                    some.host.com
Host IP address:              10.0.0.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/some.host.com.twd
Command line used:            /usr/sbin/tripwire --init

=================================================================
Object Summary:
=================================================================
-----------------------------------------------------------------
# Section: Unix File System
-----------------------------------------------------------------
     Mode        UID          Size       Modify Time
     ------      ----------   ---------- ----------
 /
     drwxr-xr-x  root (0)     XXX        XXXXXXXXXXXXXXXXX
 /bin
     drwxr-xr-x  root (0)     4096       Mon Jan  8 08:20:45 2001
 /bin/arch
     -rwxr-xr-x  root (0)     2844       Tue Dec 12 05:51:35 2000
 /bin/ash
     -rwxr-xr-x  root (0)     64860      Thu Dec  7 22:35:05 2000
 /bin/ash.static
     -rwxr-xr-x  root (0)     405576     Thu Dec  7 22:35:05 2000
```

To see information about a particular file that Tripwire is tracking, such as `/etc/hosts`, use the following command:

```
/usr/sbin/twprint -m d --print-dbfile /etc/hosts
```

The result will look similar to this:

```
Object name:  /etc/hosts

Property:                Value:
-------------            -----------
Object Type              Regular File
Device Number            773
Inode Number             216991
Mode                     -rw-r--r--
Num Links                1
UID                      root (0)
GID                      root (0)
```

See man page for `twprint` for more options.

UPDATING THE TRIPWIRE DATABASE

To update the Tripwire database so it accepts valid policy violations, Tripwire first cross-references a report file against the database and then integrates into it valid violations from the report file. When updating the database, be sure to use the most recent report.

Use the following command to update the Tripwire database, where *`name`* is the name of the most recent report file:

```
/usr/sbin/tripwire --update --twrfile
/var/lib/tripwire/report/<name>.twr
```

Tripwire will display the report file using the default text editor specified on the `EDITOR` line of the Tripwire configuration file. This gives you an opportunity to deselect files you do not wish to update in the Tripwire database.

All proposed updates to the Tripwire database start with an `[x]` before the file name, similar to the following example:

```
Added:
```

```
[x] "/usr/sbin/longrun"

Modified:
[x] "/usr/sbin"
[x] "/usr/sbin/cpqarrayd"
```

If you want to specifically exclude a valid violation from being added to the Tripwire database, remove the `x`.

To edit files in the default text editor, `vi`, type `i` and press [Enter] to enter insert mode and make any necessary changes. When finished, press the [Esc] key, type `:wq`, and press [Enter].

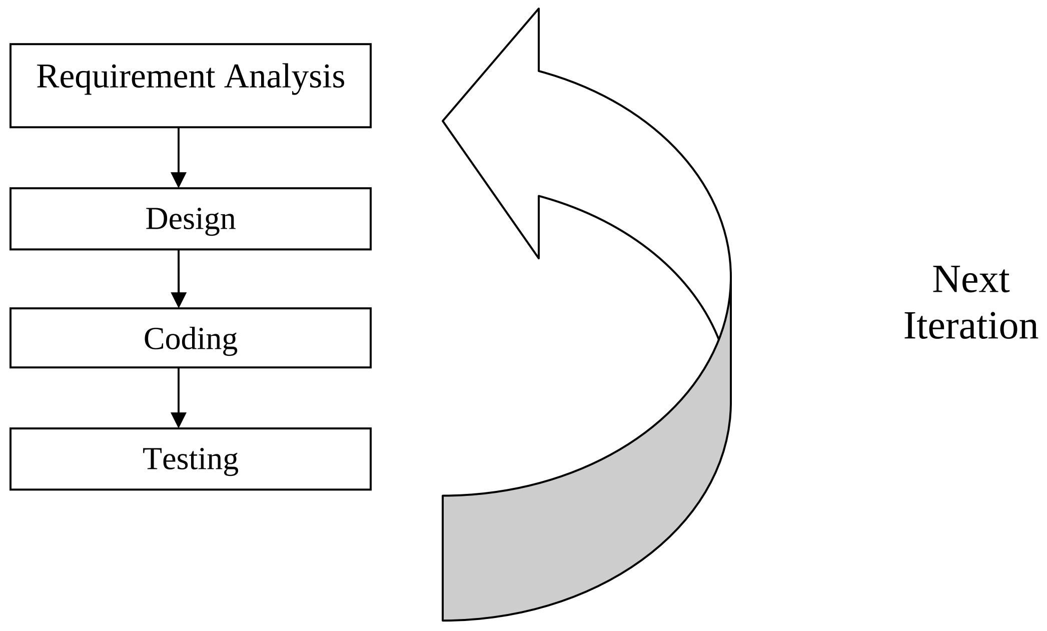
After the editor closes, enter your local password and the database will be rebuilt and signed.

After a new Tripwire database is written, the newly authorized integrity violations will no longer show up as warnings.

CHAPTER 3
DESIGN ISSUES

3.1 DEVELOPMENT MODEL USED

The Iterative Model will be used for the implementation of this project. This will encompass a requirement analysis phase followed by the design phase followed by coding and testing. These steps will be carried out in an iterative fashion. This is superior to the conventional Waterfall Model in that this model will allow us to restrict the finer scope of the project at will.



3.2 USE CASES

Various use cases are identified during the requirement analysis phase based on the user perspective, that is from the end user point of view. These use cases can be broadly categorized into

(1) FIREWALLING-BASED USE CASES

- Source-based firewalling rule
- Destination-based firewalling rule
- Source and destination-based firewalling rule
- Prorocol-based firewalling rule
- Source and protocol-based firewalling rule
- Port-based filtering
- Source and port-based firewalling rule
- Source, port and destination-based firewalling rule
- One way firewalling
- Authentication rule

(2) IDS BASED USE CASES

- Set and configure IDS.
- Show integrity report
- Activate/deactivate log watch
- Show log reports

(3) NETWORK ADAPTER BASED USE CASES

- View network adapter information

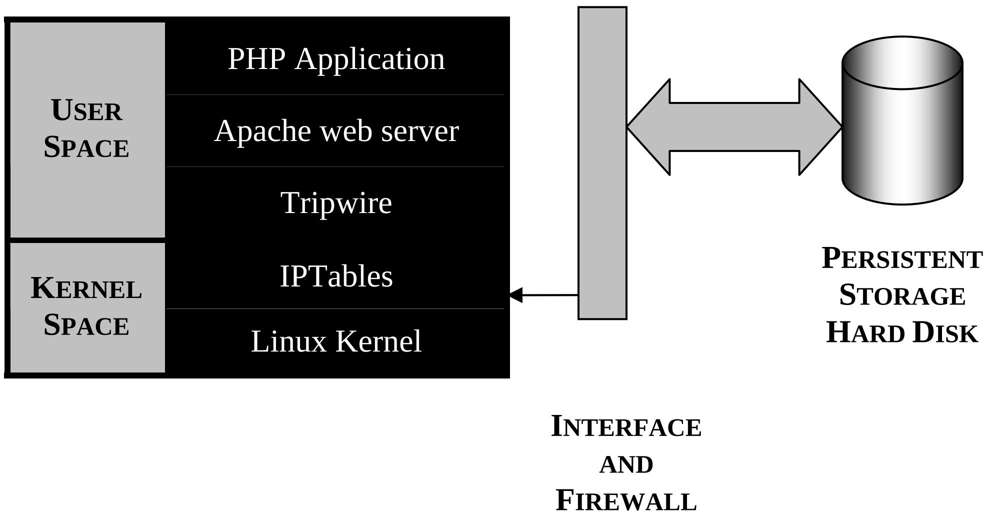
- Enable/Disable network adapter
- Protect adapter settings

(4) DNS BASED USE CASES

- View existing DNS zones
- Protect particular zone(s)

CHAPTER 4
PROJECT DESIGN

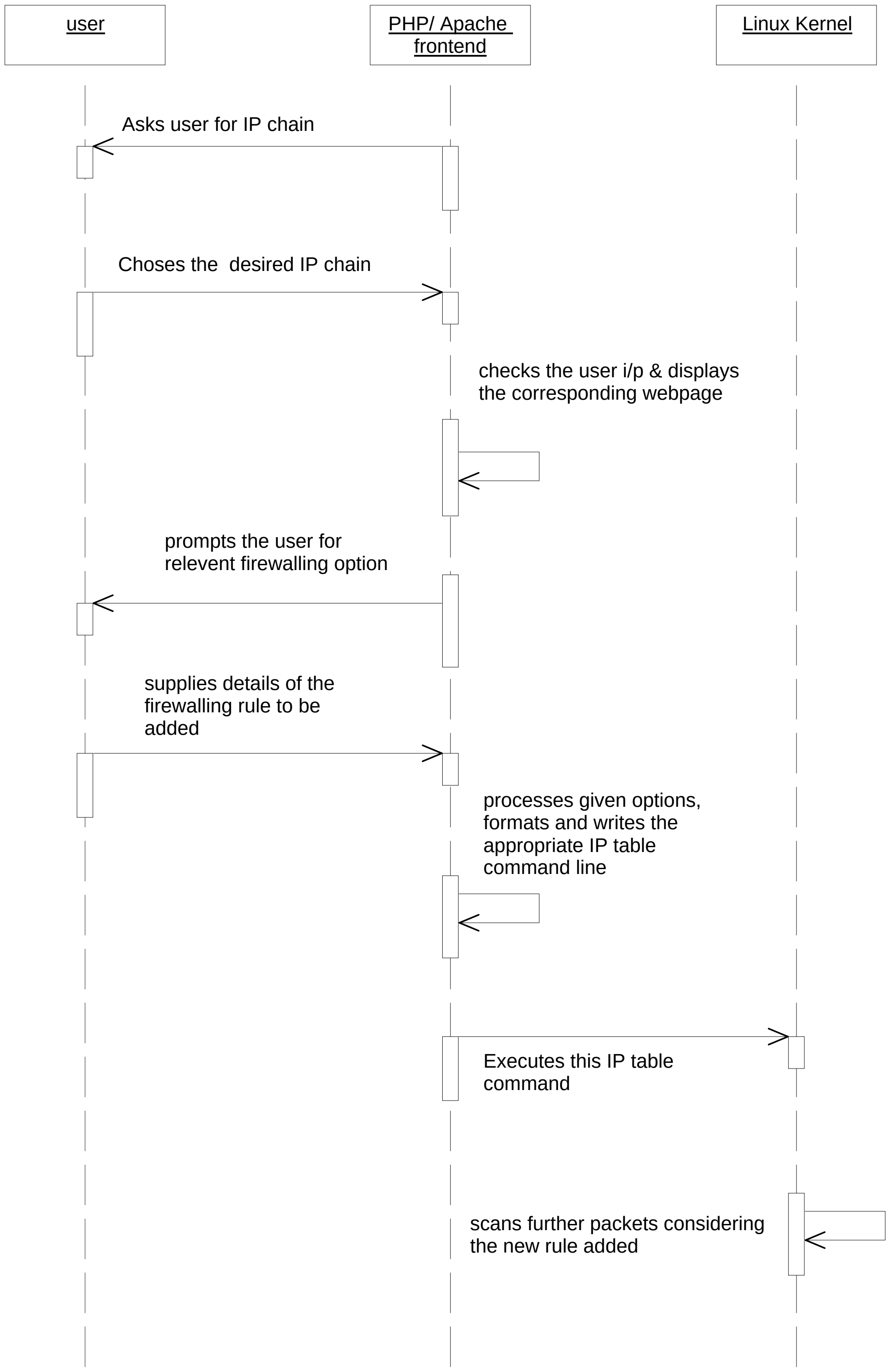
4.1 PRELIMINARY DESIGN OF THE PROJECT



The persistent storage will be used to store the following information:

(1) Various logs
(2) IDS settings
(3) Firewalling policies
(4) User account information
(5) Various user and administrator passwords

4.2 UML SEQUENCE DIAGRAMS


user
PHP/ Apache frontend
Linux Kernel
Asks user for IP chain
Choses the desired IP chain
checks the user i/p & displays the corresponding webpage
prompts the user for relevent firewalling option
supplies details of the firewalling rule to be added
processes given options, formats and writes the appropriate IP table command line
Executes this IP table command
scans further packets considering the new rule added

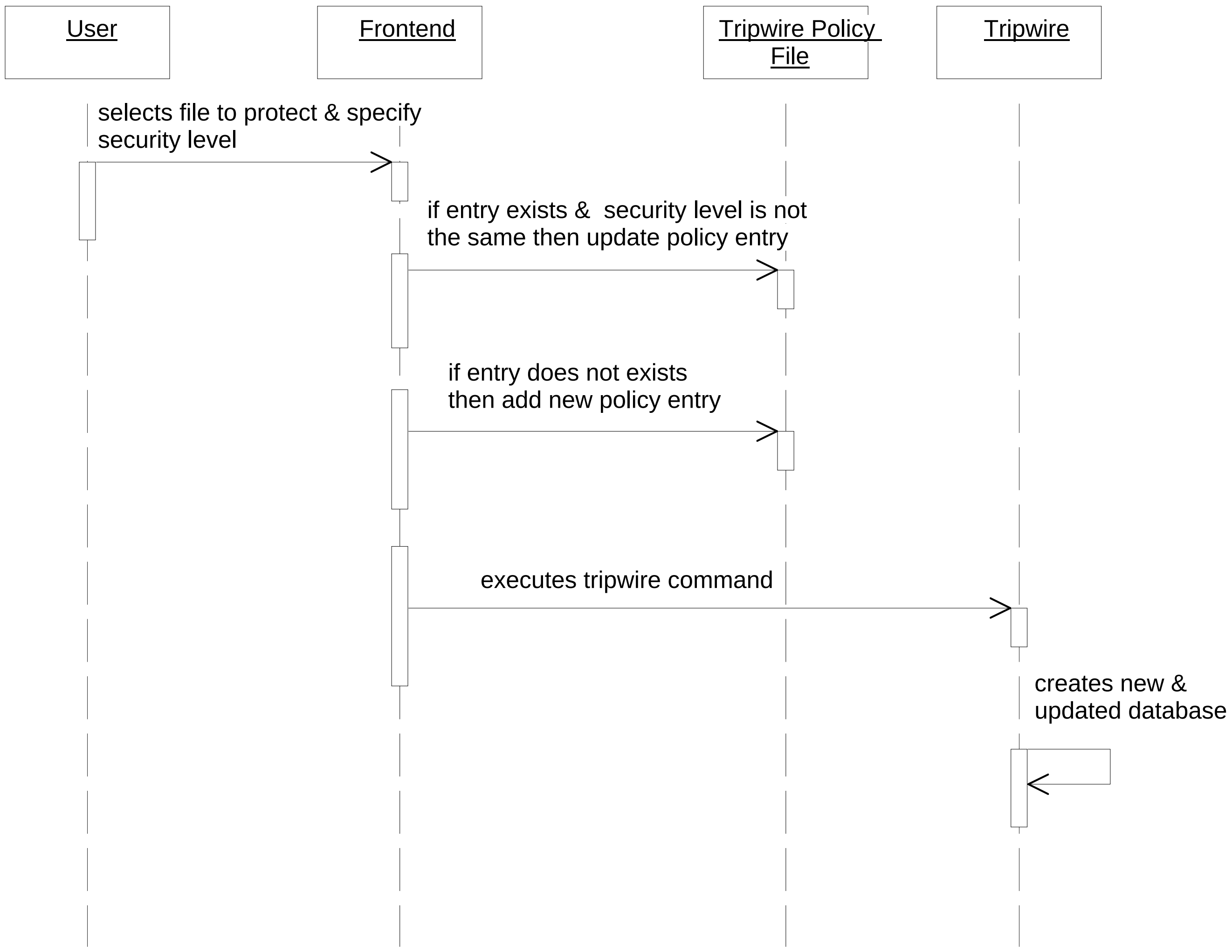

User
Frontend
Tripwire Policy File
Tripwire
selects file to protect & specify security level
if entry exists & security level is not the same then update policy entry
if entry does not exists then add new policy entry
executes tripwire command
creates new & updated database

FIGURE: FILE PROTECTION

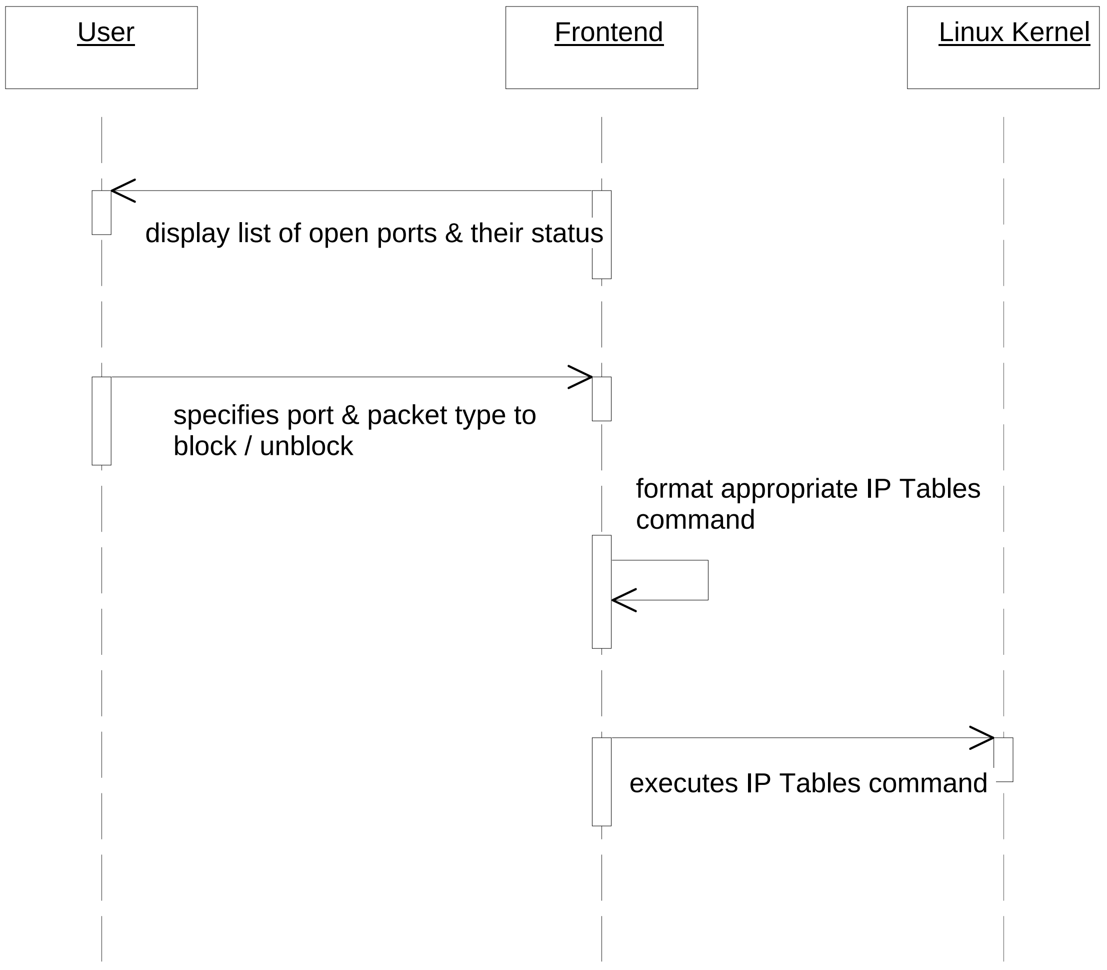

User
Frontend
Linux Kernel
display list of open ports & their status
specifies port & packet type to block / unblock
format appropriate IP Tables command
executes IP Tables command

FIGURE: PORT BLOCKING

4.3 DESCRIPTION

DIFFERENT MODULES

(1) Intrusion detection module
(2) File protection
(3) Firewalling
(4) Port blocking/unblocking
(5) Remote management console
(6) Server protection (DNS)
(7) Network adapter level protection
(8) Network information

PLATFORM USED

OS	Linux Kernel 2.4, 2.6
FRONT END	PHP 5.0/HTML
MIDDLEWARE	Apache Web Server 2.0
IMPLEMENTATION LANGUAGE	PHP DHTML (HTML, JavaScript) Shell scripting Awk

CHAPTER 5
IMPLEMENTATION ASPECTS

5.1 IMPLEMENTATION OF REMOTE MANAGEMENT CONSOLE

Remote manageability is also an important requirement of products today so that products can be managed from anywhere in the world. This project aims to provide **a remote management console created using PHP/DHTML over an Apache back end.**

PHP

INTRODUCTION

WHAT IS PHP?

PHP (recursive acronym for "PHP: Hypertext Preprocessor") is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

HOW THIS IS DIFFERENT FROM A SCRIPT WRITTEN IN OTHER LANGUAGES LIKE PERL OR C

Instead of writing a program with lots of commands to output HTML, you write an HTML script with some embedded code to do something (in this case, output some text). The PHP code is enclosed in special start and end tags that allow you to jump into and out of "PHP mode".

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server. If you were to have a script similar to the above on your server, the client would receive the results of running that script, with no way of determining what the underlying code may be. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

WHAT CAN PHP DO?

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

There are three main fields where PHP scripts are used.

1. **Server-side scripting**.

   This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a webserver and a web browser. You need to run the webserver, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server.

2. **Command line scripting**.
3. **Writing client-side GUI applications**.
4. **High Portability**.

   PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. For the majority of the servers PHP has a module, for the others supporting the CGI standard, PHP can work as a CGI processor.

FUNCTIONS USED IN THE PROJECT

ARRAY FUNCTIONS

1. array

array **array** ([mixed ...])

Returns an array of the parameters. The parameters can be given an index with the `=>` operator.

2. sort

void **sort** (array array [, int sort_flags])

This function sorts an array. Elements will be arranged from lowest to highest when this function has completed.

3. addcslashes

string **addcslashes** (string str, string charlist)

Returns a string with backslashes before characters that are listed in *`charlist`* parameter. It escapes `\n`, `\r` etc. in C-like style, characters with ASCII code lower than 32 and higher than 126 are converted to octal representation.

4. explode

array **explode** (string separator, string string [, int limit])

Returns an array of strings, each of which is a substring of *`string`* formed by splitting it on boundaries formed by the string *`separator`*. If *`limit`* is set, the returned array will contain a maximum of *`limit`* elements with the last element containing the rest of *`string`*.

If *`separator`* is an empty string (""), **explode()** will return **`FALSE`**. If *`separator`* contains a value that is not contained in *`string`*, then **explode()** will return an array containing *`string`*.

REGULAR EXPRESSION FUNCTIONS

1. ereg

int **ereg** (string pattern, string string [, array regs])

Searches a *`string`* for matches to the regular expression given in *`pattern`*.

If matches are found for parenthesized substrings of *`pattern`* and the function is called with the third argument *`regs`*, the matches will be stored in the elements of the array *`regs`*. $regs[1] will contain the substring which starts at the first left parenthesis; $regs[2] will contain the substring starting at the second, and so on. $regs[0] will contain a copy of the complete string matched.

Searching is case sensitive.

Returns **`TRUE`** if a match for pattern was found in *`string`*, or **`FALSE`** if no matches were found or an error occurred.

2. split

array **split** (string pattern, string string [, int limit])

Returns an array of strings, each of which is a substring of *`string`* formed by splitting it on boundaries formed by the regular expression *`pattern`*. If *`limit`* is set, the returned array will contain a maximum of *`limit`* elements with the last element containing the whole rest of *`string`*. If an error occurs, **split()** returns **`FALSE`**.

To split off the first four fields from a line from `/etc/passwd`:

FILE FUNCTIONS

1. file

array **file** (string filename [, int use_include_path])

file() returns the file in an array. Each element of the array corresponds to a line in the file, with the newline still attached. Upon failure, **file()** returns **`FALSE`**.

2. tempnam

string **tempnam** (string dir, string prefix)

Creates a file with a unique filename in the specified directory. If the directory does not exist, **tempnam()** may generate a file in the system's temporary directory, and return the name of that.

3. tmpfile

resource **tmpfile** (void)

Creates a temporary file with an unique name in write mode, returning a file handle The file is automatically removed when closed (using fclose()), or when the script ends.

FTP FUNCTIONS

1. ftp_connect

resource **ftp_connect** (string host [, int port [, int timeout]])

Returns a FTP stream on success or **`FALSE`** on error.

ftp_connect() opens an FTP connection to the specified `host`. The `port` parameter specifies an alternate port to connect to. If it is omitted or set to zero, then the default FTP port, 21, will be used.

The `timeout` parameter specifies the timeout for all subsequent network operations. If omitted, the default value is 90 seconds..

2. ftp_close

void **ftp_close** (resource ftp_stream)

ftp_close() closes `ftp_stream` and releases the **resource**. After calling this function, you can no longer use the FTP connection and must create a new one with **ftp_connect()**.

3. ftp_get

bool **ftp_get** (resource ftp_stream, string local_file, string remote_file, int mode [, int resumepos])

ftp_get() retrieves `remote_file` from the FTP server, and saves it to `local_file` locally. The transfer `mode` specified must be either **`FTP_ASCII`** or **`FTP_BINARY`**. Returns **`TRUE`** on success or **`FALSE`** on failure.

4. ftp_put

bool **ftp_put** (resource ftp_stream, string remote_file, string local_file, int mode [, int startpos])

ftp_put() stores `local_file` on the FTP server, as `remote_file`. The transfer `mode` specified must be either **`FTP_ASCII`** or **`FTP_BINARY`**. Returns **`TRUE`** on success or **`FALSE`** on failure.

PHP MYSQL FUNCTIONS

1. mysql_connect

resource **mysql_connect** ([string server [, string username [, string password [, bool new_link [, int client_flags]]]]])

Returns a MySQL link identifier on success, or **`FALSE`** on failure.

mysql_connect() establishes a connection to a MySQL server. The following defaults are assumed for missing optional parameters: *`server`* = 'localhost:3306', *`username`* = name of the user that owns the server process and *`password`* = empty password.

The *`server`* parameter can also include a port number. eg. "hostname:port" or a path to a socket eg. ":/path/to/socket" for the localhost.

2. mysql_query

resource **mysql_query** (string query [, resource link_identifier [, int result_mode]])

mysql_query() sends a query to the currently active database on the server that's associated with the specified link identifier. If *`link_identifier`* isn't specified, the last opened link is assumed. If no link is open, the function tries to establish a link as if **mysql_connect()** was called with no arguments, and use it.

The optional *`result_mode`* parameter can be MYSQL_USE_RESULT and MYSQL_STORE_RESULT. It defaults to MYSQL_STORE_RESULT, so the result is buffered. See also **mysql_unbuffered_query()** for the counterpart of this behaviour.

3.mysql_select_db

bool **mysql_select_db** (string database_name [, resource link_identifier])

Returns **TRUE** on success or **FALSE** on failure.

mysql_select_db() sets the current active database on the server that's associated with the specified link identifier. If no link identifier is specified, the last opened link is assumed.

PHP SUPERGLOBALS

$_SERVER

Variables set by the web server or otherwise directly related to the execution environment of the current script. Analogous to the old `$HTTP_SERVER_VARS` array (which is still available, but deprecated).

$_POST

Variables provided to the script via HTTP POST. Analogous to the old `$HTTP_POST_VARS` array (which is still available, but deprecated).

$_SESSION

Variables which are currently registered to a script's session. Analogous to the old `$HTTP_SESSION_VARS` array.

EXCERPT FROM OUR PROJECT ILLUSTRATING THE USE OF PHP

Snippet of script used in the file protection module

```
<?php
	$lines = file("securityLevel");
	foreach($lines as $line)
	{
		list($name, $desc) = explode(":", $line);
		$desc = trim($desc);
		echo "<option>$desc</option>";
	}
?>
```

5.2 IMPLEMENTATION OF USER AUTHENTICATION, LOGS AND SETTINGS

For enhanced security, flexibility and efficiency, a large number of IDS features such as the following are implemented using a MySQL database backend:

(1) User authentication
(2) IDS logs
(3) User preferences

The relevant information is stored in the form of tables in a MySQL database, and retrieved on demand.

MYSQL

INTRODUCTION

MySQL is a very fast, multi-threaded, multi-user, and robust SQL (Structured Query Language) relational database server. MySQL Server is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

THE TECHNICAL FEATURES OF MYSQL SERVER

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). MySQL Server also provides a multi-threaded library which one can link into applications to get a smaller, faster, easier-to-manage product.

INTERNALS AND PORTABILITY

- Works on many different platforms.
- Fully multi-threaded using kernel threads. This means it can easily use multiple CPUs if they are available.
- Uses very fast B-tree disk tables (MyISAM) with index compression.
- Relatively easy to add another storage engine. This is useful if you want to add an SQL interface to an in-house database.
- A very fast thread-based memory allocation system.
- Very fast joins using an optimized one-sweep multi-join.
- In-memory hash tables which are used as temporary tables.

SECURITY

A privilege and password system that is very flexible and secure, and allows host-based verification. Passwords are secure because all password traffic is encrypted when one connects to a server.

SCALABILITY AND LIMITS

Handles large databases. MySQL Server has been successfully used with databases that contain 50 million records.

TABLES USED

(1) USER

This table is used to store login information in the form of usernames and their associated passwords.

FIELD NAME	FIELD TYPE	ADDITIONAL INFORMATION
user	Text	Primary Key
password	Text	

(2) LOG

This table is used to store IDS logs. Logs are generated whenever some action takes place, such as the following:

(1) A filesystem integrity check is performed
(2) A new user is added
(3) A password is changed
(4) An authorized user logs in or logs out
(5) A port is blocked or unblocked
(6) A new rule is added to the firewall

FIELD NAME	FIELD TYPE	DESCRIPTION
date	Text	Date stamp
time	Text	Timestamp
remoteAddress	Text	User's IP Address
serverPort	Text	Server side port
remotePort	Text	User side port
action	Text	Description of action done
userAgent	Text	Information about the user agent used to access the IDS remote management console

5.3 SHELL SCRIPTING

Shell scripts are nothing but programs written in the form of individual shell commands and other variables, operators, control structures.

The shell script will resemble a higher level language program but will be executed one command at a time sequentially. There is no preprocessing involved like compilation. Therefore if there are any syntactical mistakes, they are not revealed beforehand. There is absolutely no restriction on the extension such file names should have. Shell scripts run slower than compiled languages like C, but for many jobs speed is no hurdle.

A shell script is executed after assigning executable permission to the script file or can also be run with the ‘sh’ command. We can also run a script non-interactively by specifying arguments in the command line. These arguments are accepted into the positional parameters $1, $2, and so on. Every script returns an exit status on termination. This value is stored in the parameter $?. Zero specifies a true value. Any non Zero value points to failure.

We can specify the shell a script must use by placing the statement
#! /bin/sh in the first line of the script. ‘sh’ should be replaced by ‘ksh’ and ‘bash’ when working with the Korn and bash shells.

EXCERPT FROM THE PROJECT ILLUSTRATING THE USE OF SHELL SCRIPTING

SCRIPT FOR UPDATING THE TRIPWIRE DATABASE

```
#!/bin/bash

export filename="$1"
export path="$2"
export pathEscaped="$3"
export password="$4"

tripwire --update -V "./editor \"$pathEscaped\"" -P "$4" --
twrfile "$filename"
```

This script internally calls another script ‘editor’ shown below

```
#!/bin/bash

echo "pe=$1"
echo "fn=$2"

sed -e 's/^\[x\]/\[ \]/g'  "$2" > "tempName"

a="s/^\[ \] \"$1\"$/\[x\] \"$1\"/"

echo "a is=***$a***"
sed -e "$a" "tempName" > "$2"
```

5.4 AWK

awk doesn’t belong to the do-one-thing-well family of UNIX command. In fact, it can do several things- and some of them quiet well. Unlike other filters, it operates at the field level and can easily access, transform and format individual fields in a line. It also accepts regular expressions for pattern matching, has C-type programming constructs, variables and several built–in functions. Awk is not just a command, but a programming language too. It uses an unusual syntax that uses two components and requires single quotes and curly braces:

awk options ‘selection criteria { action } ’ file(s)

The selection criteria (a form of addressing) filters input and selects line for the action component to act on. This component is enclosed within curly braces. The address (rather, the selection criteria) and action constitute an awk program that is surrounded by a set of single quotes.

The selection criteria in awk have wider scope than in *‘sed (stream editor)’*

EXAMPLE

awk program to print all the user names and user IDs of users having ID greater than 200

awk –F: ‘ $3 > 200 { print $1 , $3   } /etc/passwd

Selection Criteria Action

EXCERPT FROM THE PROJECT TO ILLUSTRATE THE USE OF AWK

# AWK PROGRAM TO CHECK IF FILE EXISTS AS A VALID ENTRY IN THE TRIPWIRE POLICY FILE

```
awk '/^[ \t]*[^#][ \t]*$awkPath/ { gsub(/[$\(\) ]/, \"\", $3); print
$1,\":\", $3; }' $policyFileName
```

5.5 DNS SERVER PROTECTION

Our project automatically detects the existing DNS zones and lists them. It provides a convenient facility to protect these zones by adding the zone files of the desired zones in the backup and restoring them if some unauthorized user tries to tamper them.

DNS (DOMAIN NAME SYSTEM SERVER)

The Domain name server (DNS) is essentially a distributed database that translates host names into IP address.

SNAPSHOT OF /ETC/NAMED.CONF FILE (DNS CONFIGURATION FILE)

```
zone "debajit.com" {
      type master;
      file "debajit.com.zone";
};
zone "." {
      type hint;
      file "named.ca";
};
zone "0.0.127.in-addr.arpa" {
      type master;
      file "named.local";
};
zone "100.168.192.in-addr.arpa" {
      type master;
      file "100.168.192.in-addr.arpa.zone";
};
```

Below table shows various zones and their zone files.

ZONE	ZONE FILES
. (a single dot representing internet root server)	Named.ca
0.0.127.in-addr.arpa	Named.local
debajit.com	debajit.com.zone
100.168.192.in-addr.arpa	100.168.192.in-addr.arpa.zone

ZONE FILE PROTECTION

Zone names and their corresponding zone files are extracted from the named.conf DNS configuration file, and the user has the choice to *protect* one or more zones.

The files corresponding to the user-selected zone(s) are added to the Tripwire policy file for protection from tampering.

IMPLEMENTATION

(1) Two Bash scripts *getfile* and *getzone* are used to respectively extract the DNS zone names and zone files.

getzone BASH SCRIPT

```
#! /bin/bash
grep "^[\t ]*zone"  $1 | cut -d "\"" -f2 > $2
```

(2) These zone names are listed, and the user has the choice to simply click and protect a zone.

(3) The user is then redirected to the file protection module wherein he can specify the security level and the fingerprint details.

(4) Accordingly, appropriate entries are made in the Tripwire policy file.

5.6 PROTECTION AT NETWORK ADAPTER LEVEL

We provide a facility to completely disable a network interface through the remote management console such that all network communication through that network interface is completely cut off.
This feature can be used as a last resort to protect the system in case of critical situations.

This is implemented using ifcfg scripts, ifconfig ,ifup and ifdown

IFCONFIG

DESCRIPTION

Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time to set up interfaces as necessary. After that, it is usually only needed when debugging or when system tuning is needed.

SYNTAX

ifconfig [interface] [options]

OPTIONS

OPTION	DESCRIPTION
interface	The name of the interface.This is usually a driver name followed by a unit number, for example eth0 for the first Ethernet interface.
up	This flag causes the interface to be activated. It is implicitly specified if an address is assigned to the interface.
down	This flag causes the driver for this interface to be shut down.
netmask addr	Set the IP network mask for this interface. This value defaults to the usual class A, B or C network mask (as derived from the interface IP address), but it can be set to any value.
Address	The IP address to be assigned to this interface.

5.7 PORT SCANNING AND PORT BLOCKING

We provide a facility to list all the open ports and blocking incoming, outgoing or forwarded packets passing through that port, through the remote management console such that all network communication through that port can be completely cut off if required. This feature can be used to block a port if any malicious port is detected to be open. It can also be used to block certain standard services which are assigned standard ports. The listing of all the open ports is implemented using netstat.

NETSTAT

Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

Netstat prints information about the Linux networking subsystem. The type of information printed is controlled by the first argument, as follows:

By default, netstat displays a list of open sockets. If you don’t specify any address families, then the active sockets of all configured address families will be printed.

OPTIONS

--numeric , -n	Show numerical addresses instead of trying to determine symbolic host, port or user names
--numeric-ports	shows numerical port numbers but does not affect the resolution of host or user names
-a, --all	Show both listening and non-listening sockets. With the --interfaces option, show interfaces that are not marked
--tcp	Lists information about tcp network connections.
--udp	Lists information about udp network connections.

5.8 KERNEL IP ROUTING TABLE

Our project also provides an added facility to display the kernel ip routing table. This is implemented using the ***route command***.

ROUTE

It is used to show / manipulate the IP routing table.

DESCRIPTION

Route manipulates the kernel’s IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig(8) program.

When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.

OUTPUT

The output of the kernel routing table is organized in the Following Columns

Destination	The destination network or destination host.
Gateway	The gateway address or ’*’ if none set.
Genmask	The netmask for the destination net; ’255.255.255.255’ for a host destination and ’0.0.0.0’ for the default route.
Iface	Interface to which packets for this route will be sent.

5.9 FIREWALLING

This project along with intrusion detection capabilities, like file system integrity check, also provides a well defined, intuitive and remotely manageable interface for a Linux firewall. This module uses current firewall management tool, IP Tables as the base. All such current tools available have a non-intuitive and cryptic command line interface for firewalling which makes it even more difficult for a novice Linux user to manage the firewall. Thus setting up of a firewall is often neglected by the average user , henceforth, compromising network security. Our module is designed in such a way that any user can manage a firewall without sound knowledge of Linux.

IPTABLES

iptables - administration tool for IPv4 packet filtering and NAT

SYNTAX

iptables [-t table] -[AD] chain rule-specification [options]
 ptables [-t table] -D chain rulenum [options]
iptables [-t table] -P chain target [options]

DESCRIPTION

Iptables is used to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel. Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.

Each chain is a list of rules which can match a set of packets. Each rule specifies what to do with a packet that matches. This is called a ***target***, which may be a jump to a user-defined chain in the same table.

TARGETS

A firewall rule specifies criteria for a packet, and a target. If the packet does not match, the next rule in the chain is the examined; if it does match, then the next rule is specified by the value of the target, which can be the name of a user-defined chain or one of the special values ACCEPT, DROP, QUEUE, or RETURN.

TARGET	DESCRIPTION
Accept	Let the packet through
Drop	Drop the packet on the floor.
Queue	pass the packet to userspace (if supported by the kernel).
Return	stop traversing this chain and resume at the next rule in the previous (calling) chain

OPTIONS

The options that are recognized by iptables can be divided into several different groups.

COMMANDS

These options specify the specific action to perform. Only one of them can be specified on the command line unless otherwise specified below. For all the long versions of the command and option names, you need to use only enough letters to ensure that iptables can differentiate it from all other options.

OPTION	DESCRIPTION	ADDITIONAL DETAILS
-A, --append	Append one or more rules to the end of the selected chain.	When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination.
-D, --delete chain rulenum	Delete one or more rules from the selected chain	Two versions of this command are

		• The rule can be specified as a number in the chain (starting at 1 for the first rule) • A rule to match.
-F, --flush [chain]	Flush the selected chain.	This is equivalent to deleting all the rules one by one.
-L, --list [chain]	`List all rules in the selected chain`	`If no chain is selected, all chains are listed.`

PARAMETERS

The following parameters make up a rule specification (as used in the add, delete, insert, replace and append commands).

-p, --protocol[!] protocol	The protocol of the rule or of the packet to check.	The specified protocol can be one of **tcp**, **udp**, **icmp**, or **all,** or it can be a **Numeric value**, representing one of these protocols or a different one.
-s, --source, --src [!] address[/mask] [!] [port[:port]]	Source specification.	Address can be either a hostname, a network name, or a plain IP address. The mask can be either a network mask or a plain number, specifying the number of 1's at the left side of the network mask.
--source-port [!] [port[:port]]	allows separate specification of the source port or port range.	The flag --sport is an alias for this option.
-d, --destination, --dst [!] address[/mask] [!] [port[:port]]	Destination specification.	A "destination port" refers to the numeric ICMP code.

--destination-port [!] [port[:port]]	allows separate specification of the ports.	The flag --dport is an alias for this option.
--icmp-type [!] typename	allows specification of the ICMP type	This is often more convenient than appending it to the destination specification.
-j, --jump target	specifies the target of the rule; ie. what to do if the packet matches it	The target can be a user-defined chain (not the one this rule is in) or one of the special targets which decide the fate of the packet immediately. If this option is omitted in a rule, then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented.
-i, --interface [!] name		Name of an interface via which a packet is going to be received (only for packets entering the INPUT, FORWARD and PREROUTING chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+", then any interface which begins with this name will match. If this option is omitted, any interface name will match.

tcp

These extensions are loaded if `--protocol tcp' is specified. It provides the following options:

--source-port [!] port[:port]

Source port or port range specification. This can either be a service name or a port number. An inclusive range can also be specified, using the format port:port. If the first port is omitted, "0" is assumed; if the last is omitted, "65535" is assumed. If the second port greater then the first they will be swapped. The flag --sport is a convenient alias for this option.

--destination-port [!] port[:port]

Destination port or port range specification. The flag --dport is a convenient alias for this option

udp

These extensions are loaded if `--protocol udp' is specified. It provides the following options:

--source-port [!] port[:port]

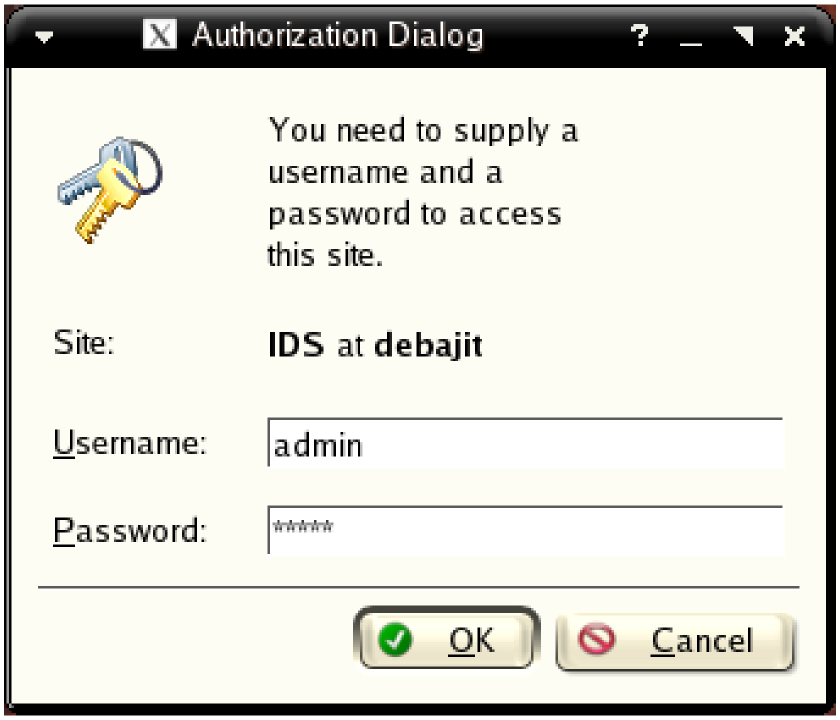
Source port or port range specification. See the description of the --source-port option of the TCP extension for details.

--destination-port [!] port[:port]

Destination port or port range specification. See the description of the --destination-port option of the TCP extension for details.

CHAPTER 6
USER MANUAL

6.1 LOGIN SCREEN



You need to supply a username and a password to access this site.

Site: **IDS** at **debajit**

Username: admin

Password: *****

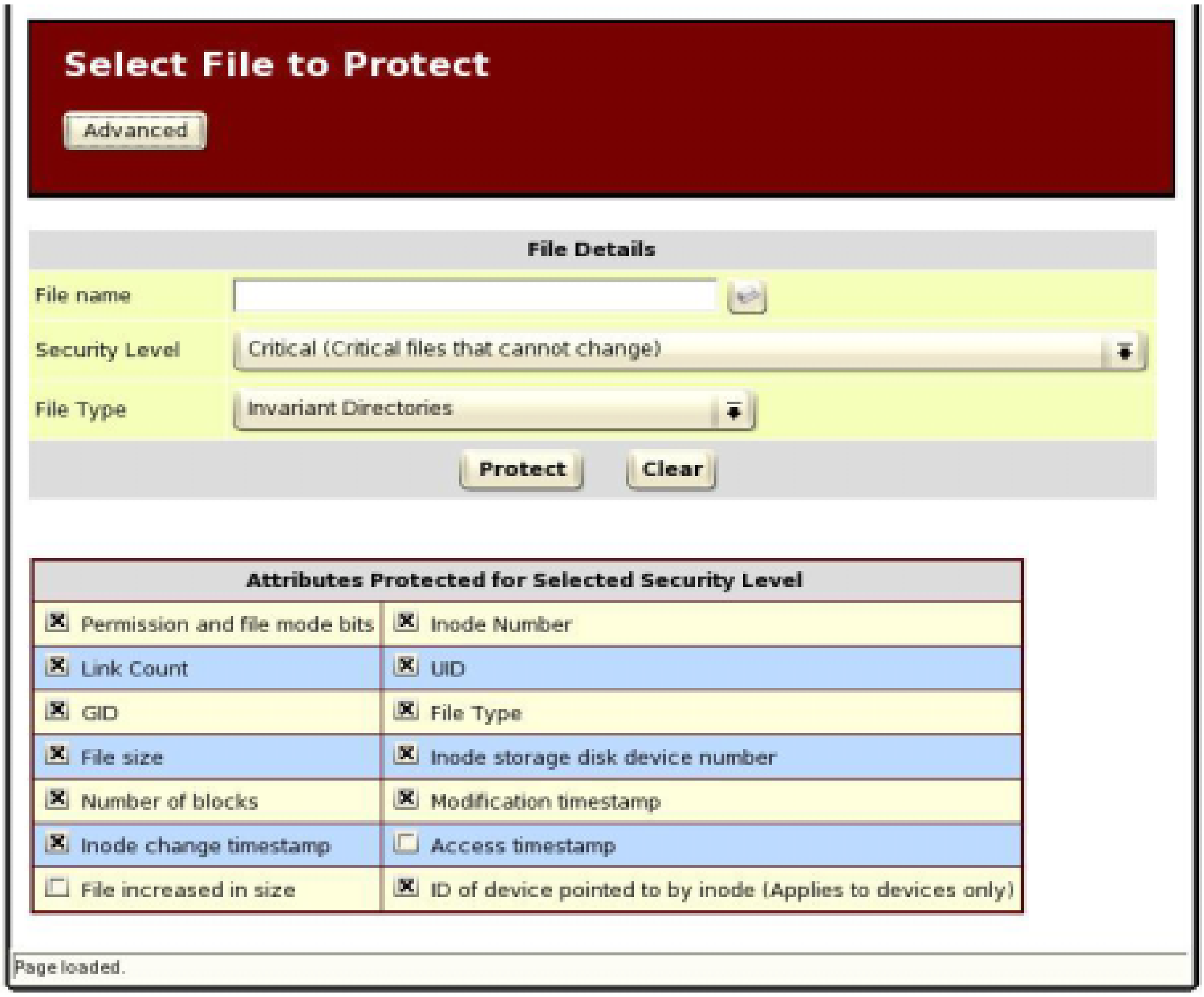
This is the first screen you will see when you try to access the IDS remote management console. You must enter a valid username and password to login.

(1) If this is the first time you are using the IDS, then enter "admin" for both username and password.

(2) After you log in, you can add or modify user accounts. You may also change your own password. Just go to the IDS section and click on Users.

(3) If login is successful, you will be redirected to the main screen of the IDS remote management console, as shown in Figure 2.

6.2 FILE PROTECTION



Select File to Protect

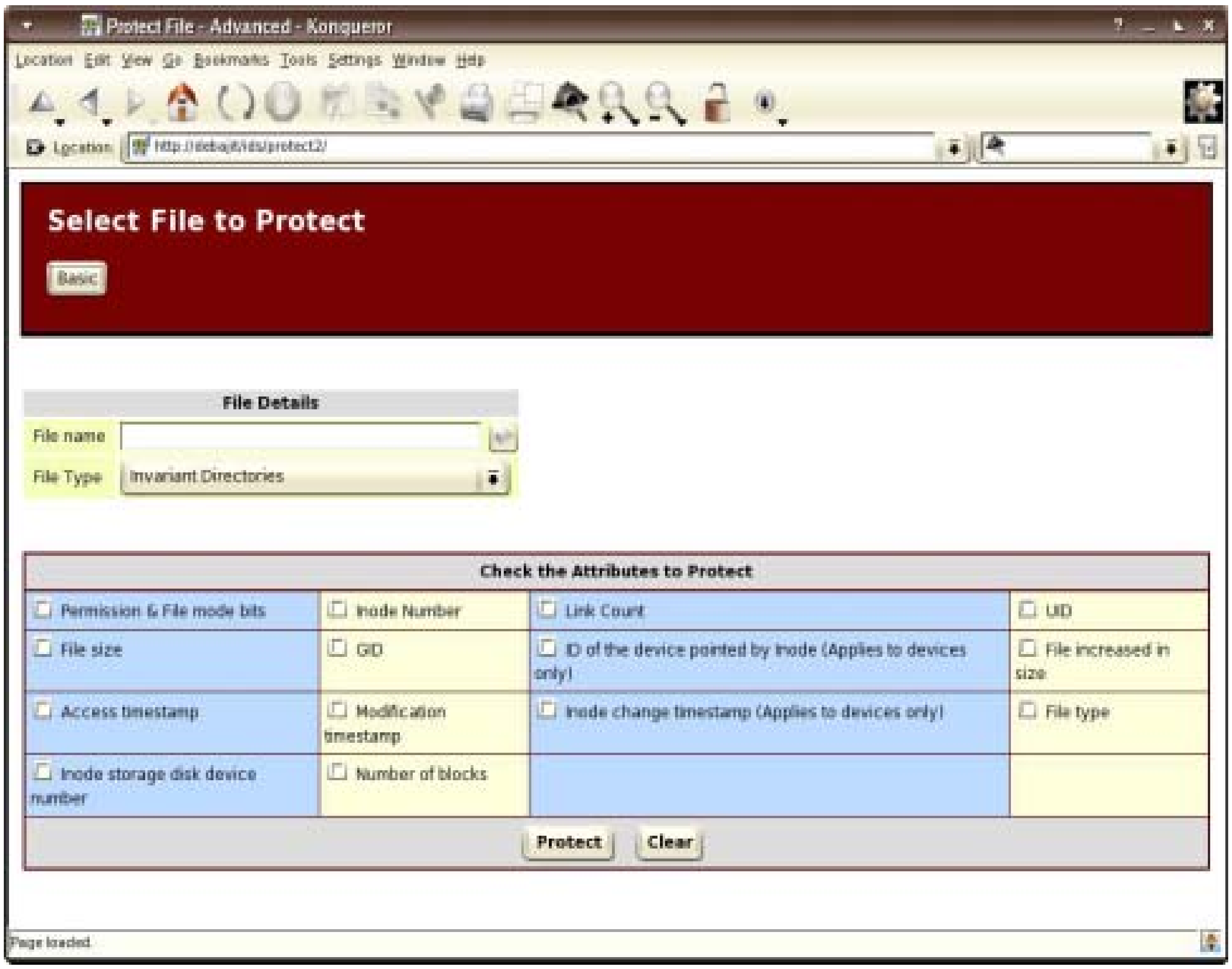
File Details	
File name	
Security Level	Critical (Critical files that cannot change)
File Type	Invariant Directories

Attributes Protected for Selected Security Level	
☒ Permission and file mode bits	☒ Inode Number
☒ Link Count	☒ UID
☒ GID	☒ File Type
☒ File size	☒ Inode storage disk device number
☒ Number of blocks	☒ Modification timestamp
☒ Inode change timestamp	☐ Access timestamp
☐ File increased in size	☒ ID of device pointed to by inode (Applies to devices only)

Page loaded.

1) The screen shows the details of the file which need to be protected.
2) The user enters the file name which need to be protected and selects appropriate security level. The file attributes get selected by their own for the selected security level.
3) When user clicks "**Protect**" button the file gets protected.
4) When user clicks "**Advanced**" button the user is directed to advanced file protection screen.

FILE PROTECTION


Protect File - Advanced - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location:
Basic
Protect
Clear

http://debajit/ids/protect2/

Select File to Protect

File Details	
File name	
File Type	Invariant Directories

Check the Attributes to Protect			
☐ Permission & File mode bits	☐ Inode Number	☐ Link Count	☐ UID
☐ File size	☐ GID	☐ ID of the device pointed by inode (Applies to devices only)	☐ File increased in size
☐ Access timestamp	☐ Modification timestamp	☐ Inode change timestamp (Applies to devices only)	☐ File type
☐ Inode storage disk device number	☐ Number of blocks		

Page loaded.

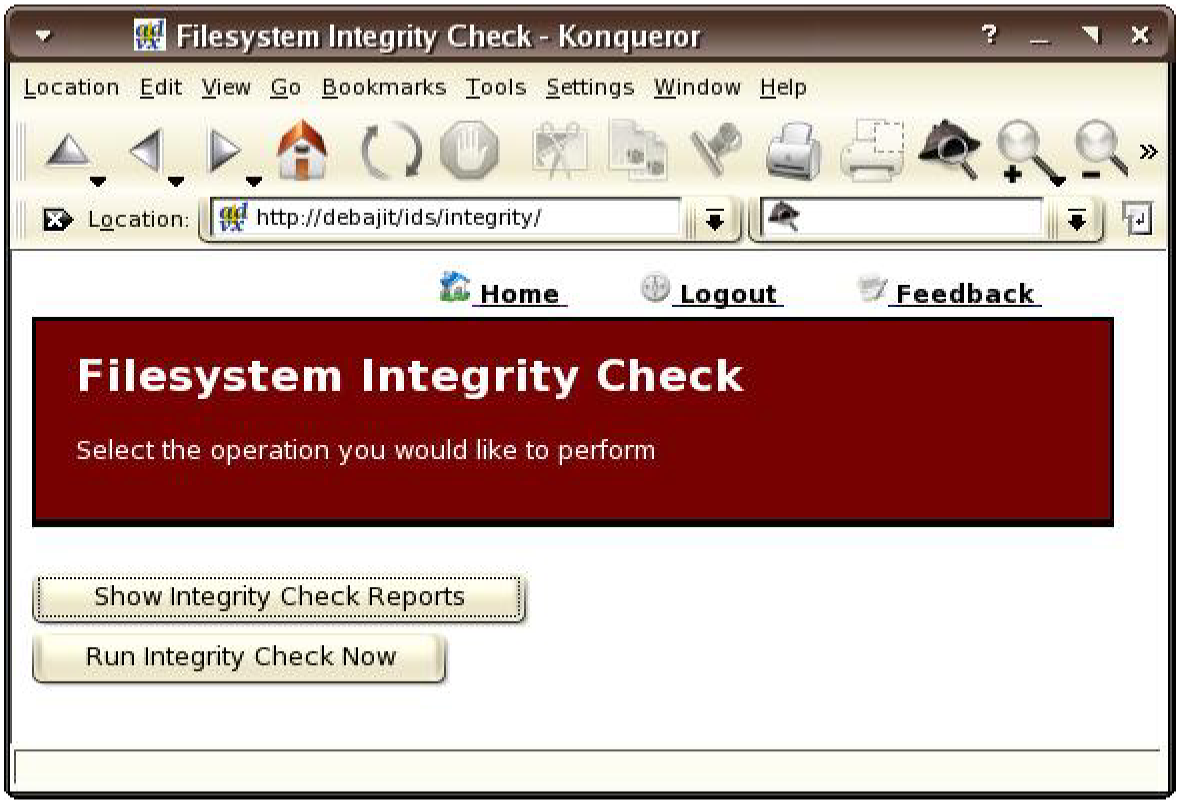
1) The screen give details about the file to be protected.
2) The user enters the file name to be protected along with file type and selects the file attributes which need to be protected for the given file.
3) When user clicks “**Protect**” button, the file gets protected with the specified file type and attribute option.
4) When user clicks “**Basic**” button, he/she is directed to basic file protection scrren.

6.3 FILESYSTEM INTEGRITY CHECKING

To perform any filesystem integrity checking operations, click on

IDS ➡ Filesystem Integrity Checking

This will open the following screen


Filesystem Integrity Check - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/integrity/
Home
Logout
Feedback
Show Integrity Check Reports
Run Integrity Check Now

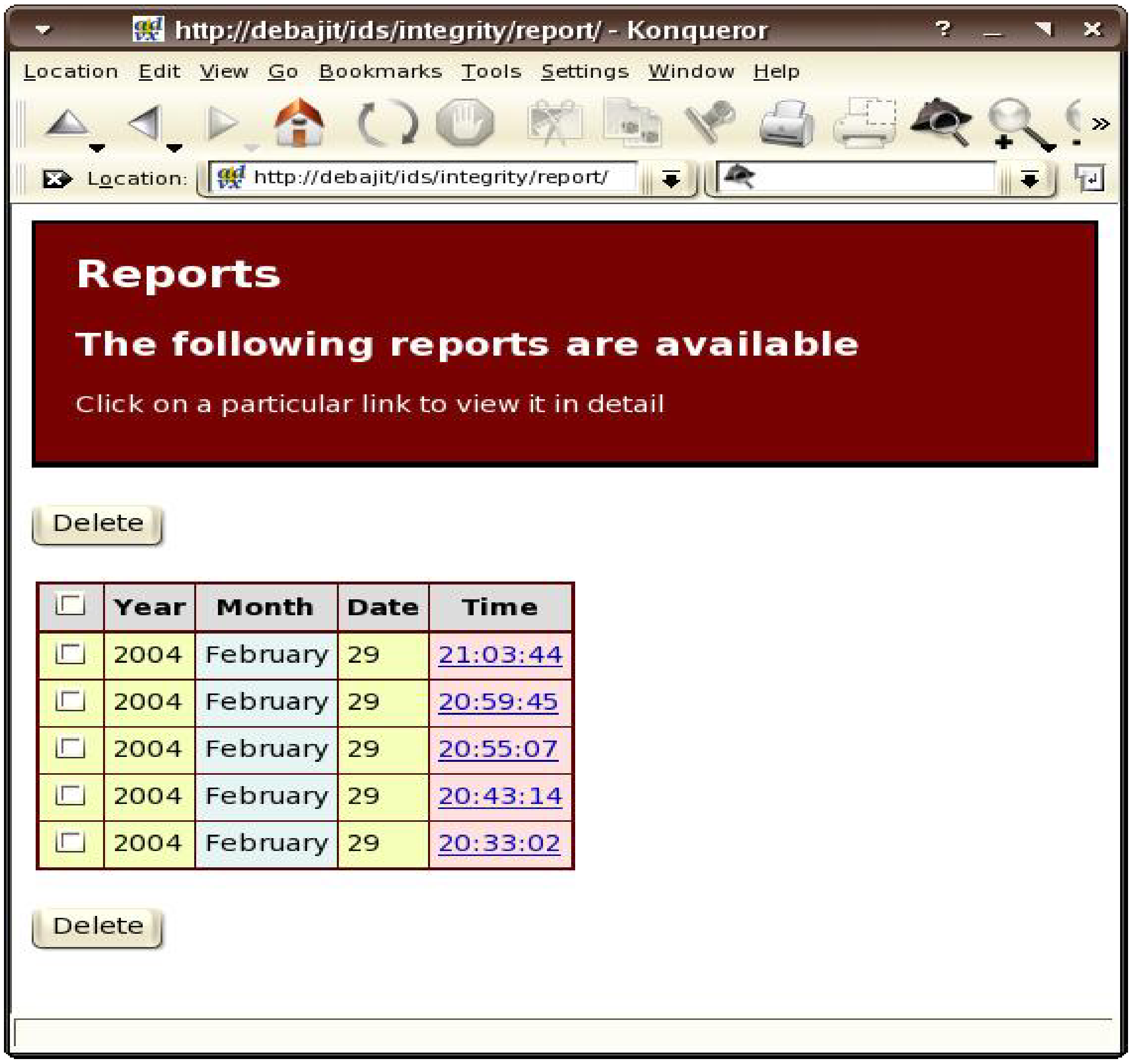
Filesystem Integrity Check

Select the operation you would like to perform

(1) Click on **Show Integrity Check Reports** to view a list of filesystem integrity reports available on the system

(2) Click on **Run Integrity Check Now** to perform a filesystem integrity check immediately and generate a report.

6.4 REPORT


http://debajit/ids/integrity/report/ - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/integrity/report/
Delete
Delete

Reports

The following reports are available

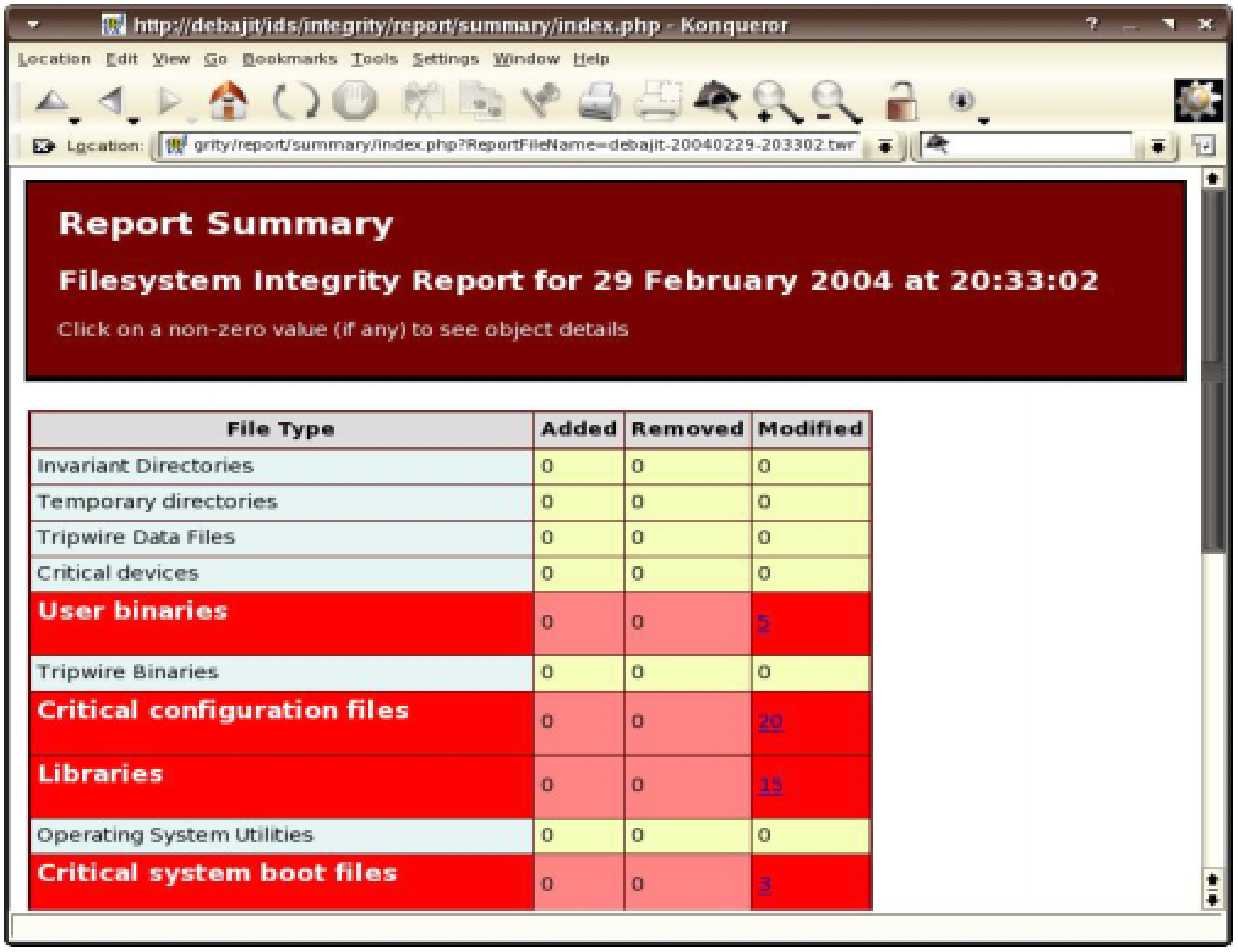
Click on a particular link to view it in detail

☐	Year	Month	Date	Time
☐	2004	February	29	21:03:44
☐	2004	February	29	20:59:45
☐	2004	February	29	20:55:07
☐	2004	February	29	20:43:14
☐	2004	February	29	20:33:02

1) The screen shows information of all generated report on a particular date and time.

2) When the user who is logged in clicks on the "**time**" he/she will be directed to view report page on the particular date and time (timestamp).

REPORT DETAILS


http://debajit/ids/integrity/report/summary/index.php - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: grity/report/summary/index.php?ReportFileName=debajit-20040229-203302.twr

Report Summary

Filesystem Integrity Report for 29 February 2004 at 20:33:02

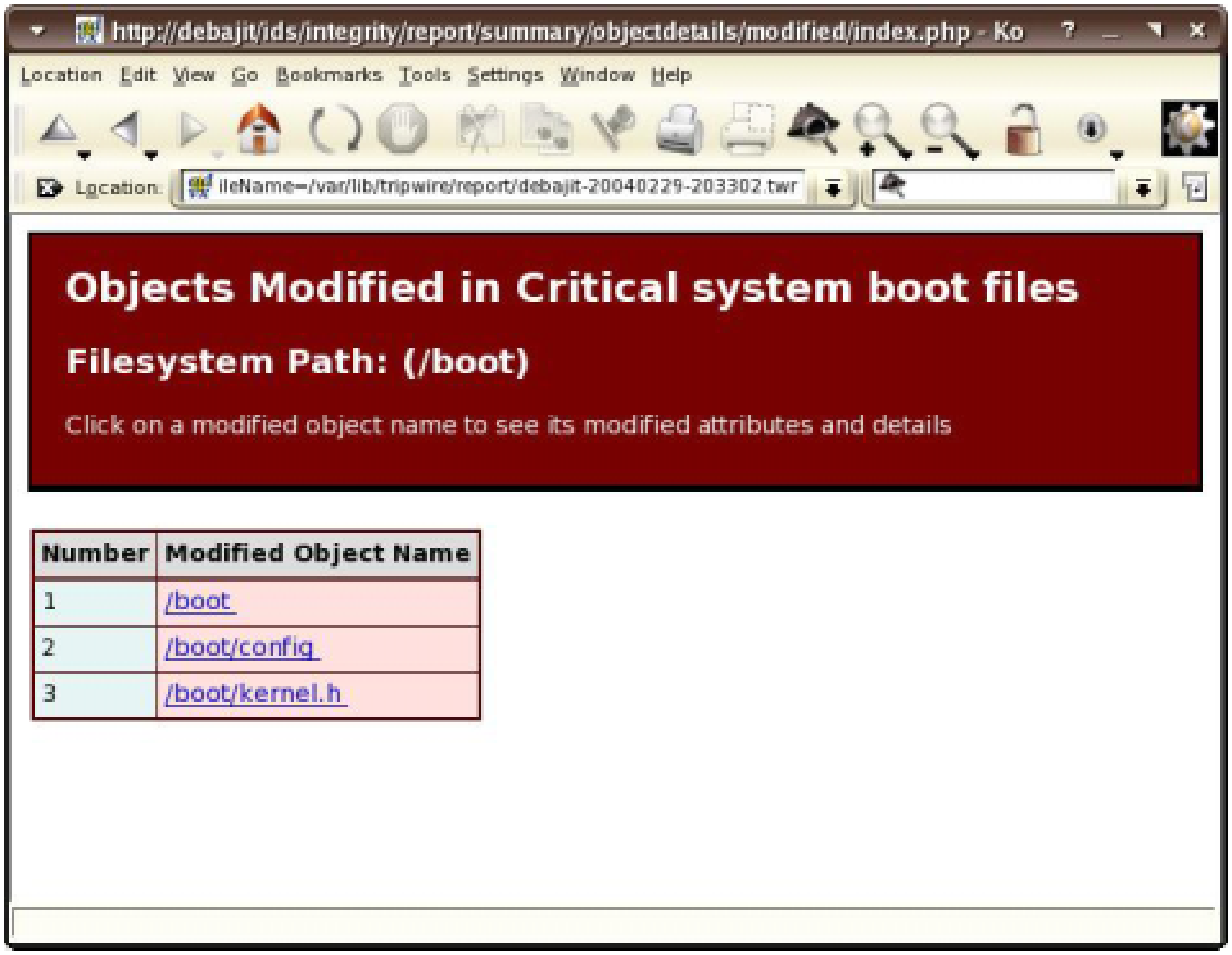
Click on a non-zero value (if any) to see object details

File Type	Added	Removed	Modified
Invariant Directories	0	0	0
Temporary directories	0	0	0
Tripwire Data Files	0	0	0
Critical devices	0	0	0
User binaries	0	0	5
Tripwire Binaries	0	0	0
Critical configuration files	0	0	20
Libraries	0	0	15
Operating System Utilities	0	0	0
Critical system boot files	0	0	3

1) The screen shows the details about the files which are being modified, removed or added in the filesystem .

2) When the user clicks the “**non-zero**” values in objects list of file type, he/she will be directed to file details page which will show the file names which are being removed, added or modified.

FILE DETAILS


http://debajit/ids/integrity/report/summary/objectdetails/modified/index.php - Ko
Location Edit View Go Bookmarks Tools Settings Window Help
Location: ileName=/var/lib/tripwire/report/debajit-20040229-203302.twr

Objects Modified in Critical system boot files

Filesystem Path: (/boot)

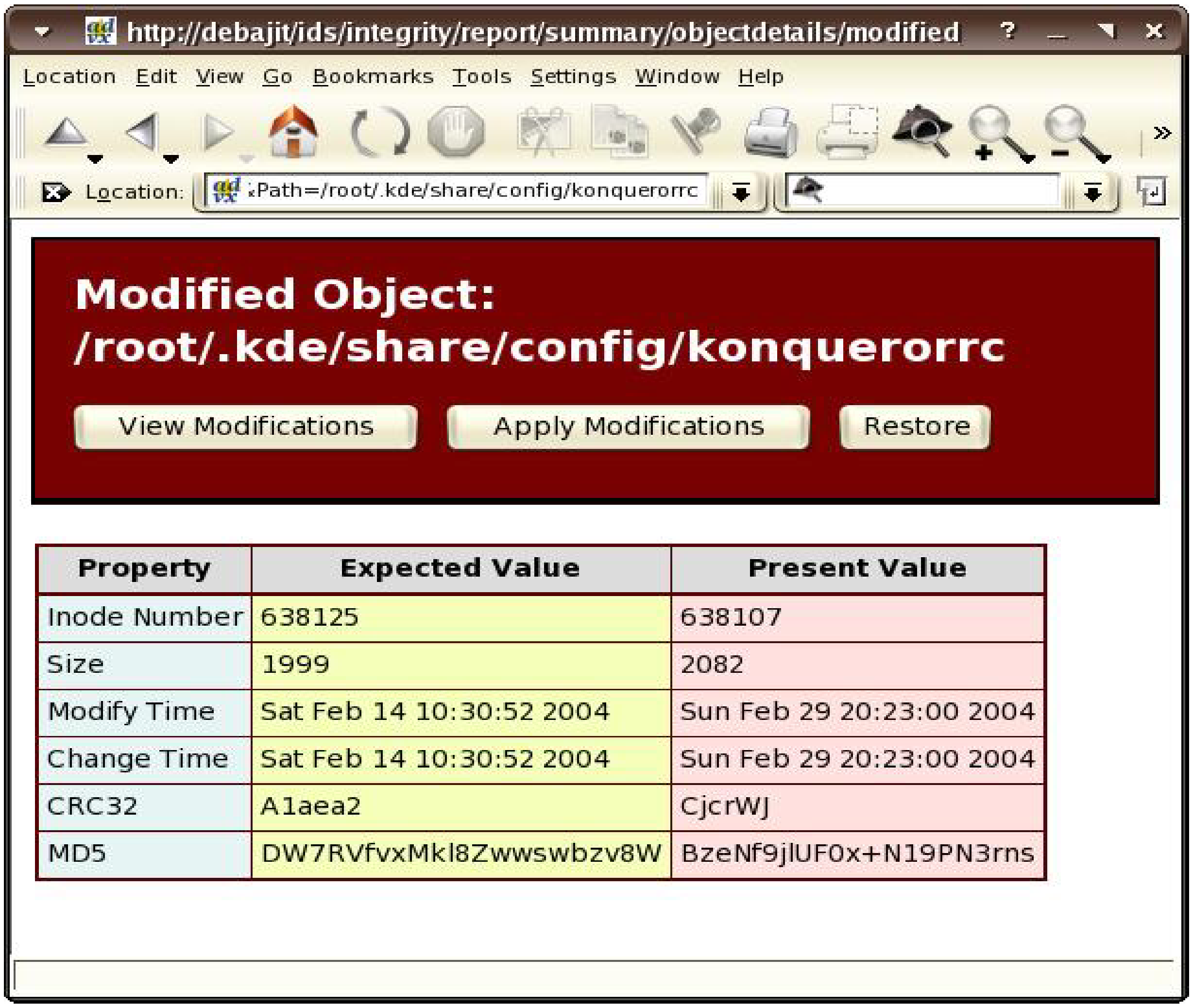
Click on a modified object name to see its modified attributes and details

Number	Modified Object Name
1	/boot
2	/boot/config
3	/boot/kernel.h

1) The screen shows details of the objects name of particular file type.

2) When the user clicks the "**Object name**", he/she will be directed to file attribute page which will show file attributes which are being modified for the current object name.

FILE ATTRIBUTES DETAILS

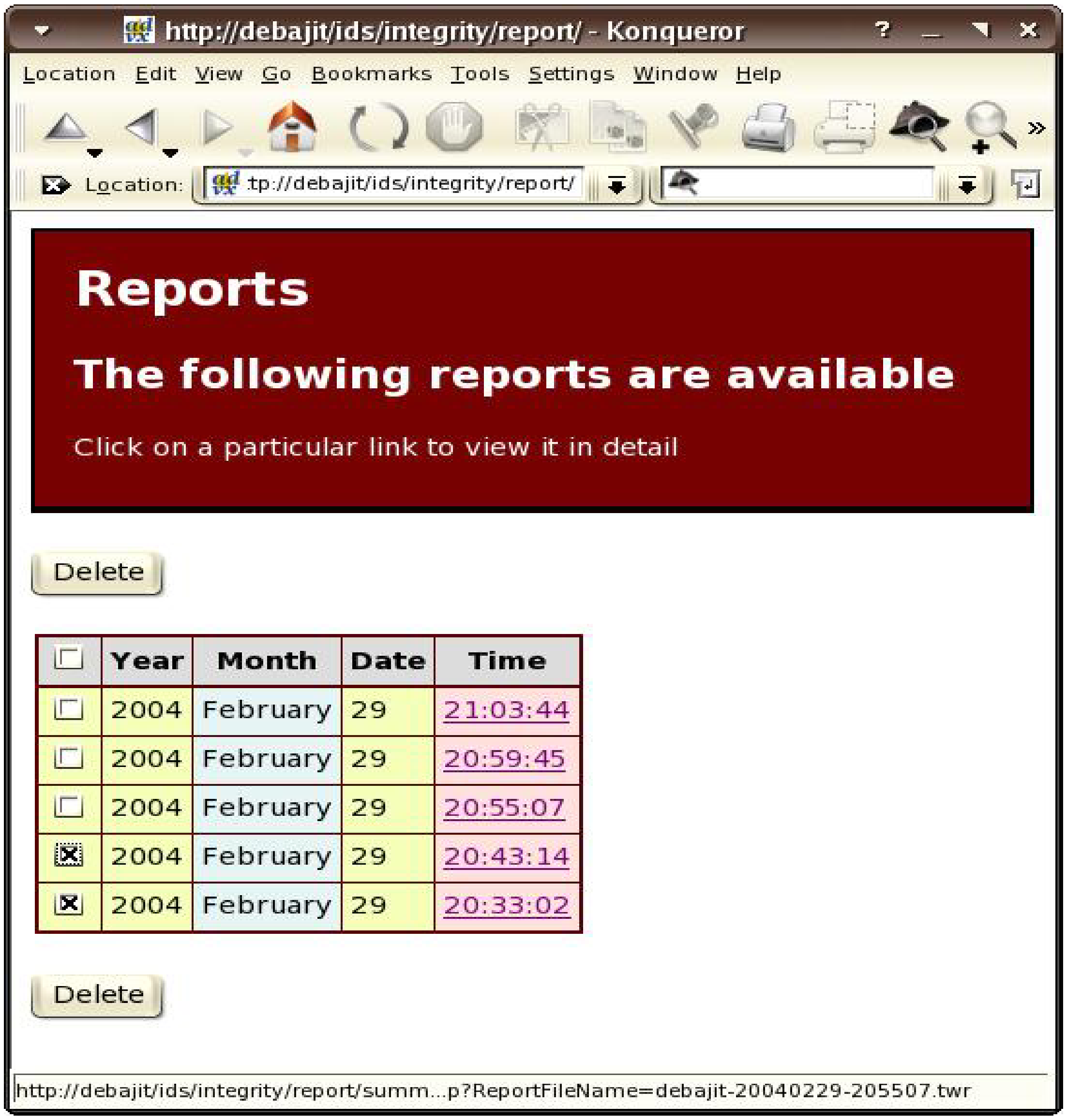

http://debajit/ids/integrity/report/summary/objectdetails/modified
Location Edit View Go Bookmarks Tools Settings Window Help
Location: ;Path=/root/.kde/share/config/konquerorrc
View Modifications
Apply Modifications
Restore

Modified Object:
/root/.kde/share/config/konquerorrc

Property	Expected Value	Present Value
Inode Number	638125	638107
Size	1999	2082
Modify Time	Sat Feb 14 10:30:52 2004	Sun Feb 29 20:23:00 2004
Change Time	Sat Feb 14 10:30:52 2004	Sun Feb 29 20:23:00 2004
CRC32	A1aea2	CjcrWJ
MD5	DW7RVfvxMkl8Zwwswbzv8W	BzeNf9jlUF0x+N19PN3rns

1) The screen shows details of various file attributes which are being modified.
2) When user clicks on "**Apply Modification**" button, the Tripwire database is updated .
3) The user can see modification done to the file by clicking "**View Modification**" button which will show the modification of the file through "**Diff**" command.
4) The user can restore the original file which is being modified or removed by clicking the "**Restore**" button.

REPORT DELETION


http://debajit/ids/integrity/report/ - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: tp://debajit/ids/integrity/report/
Delete
Delete

Reports

The following reports are available

Click on a particular link to view it in detail

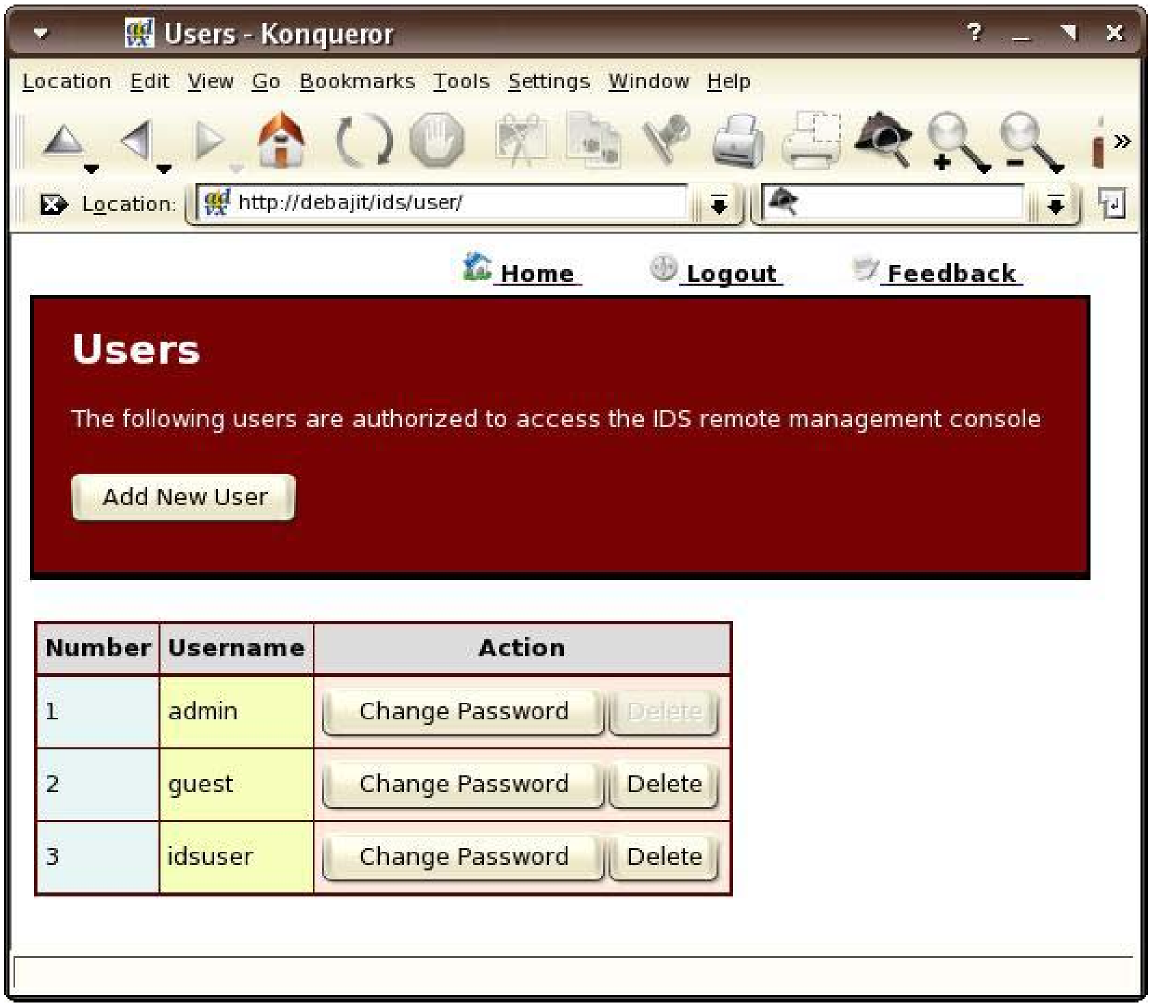
☐	Year	Month	Date	Time
☐	2004	February	29	21:03:44
☐	2004	February	29	20:59:45
☐	2004	February	29	20:55:07
☒	2004	February	29	20:43:14
☒	2004	February	29	20:33:02

http://debajit/ids/integrity/report/summ...p?ReportFileName=debajit-20040229-205507.twr

1) The screen show the information of generated report for particular date and time.

2) The user has the facility to delete a particular report by selecting the report on particular date and time by checking the checkboxes and clicking the delete button.

6.5 USERS


Users - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/user/
Home
Logout
Feedback
Add New User

Users

The following users are authorized to access the IDS remote management console

Number	Username	Action
1	admin	Change Password Delete
2	guest	Change Password Delete
3	idsuser	Change Password Delete

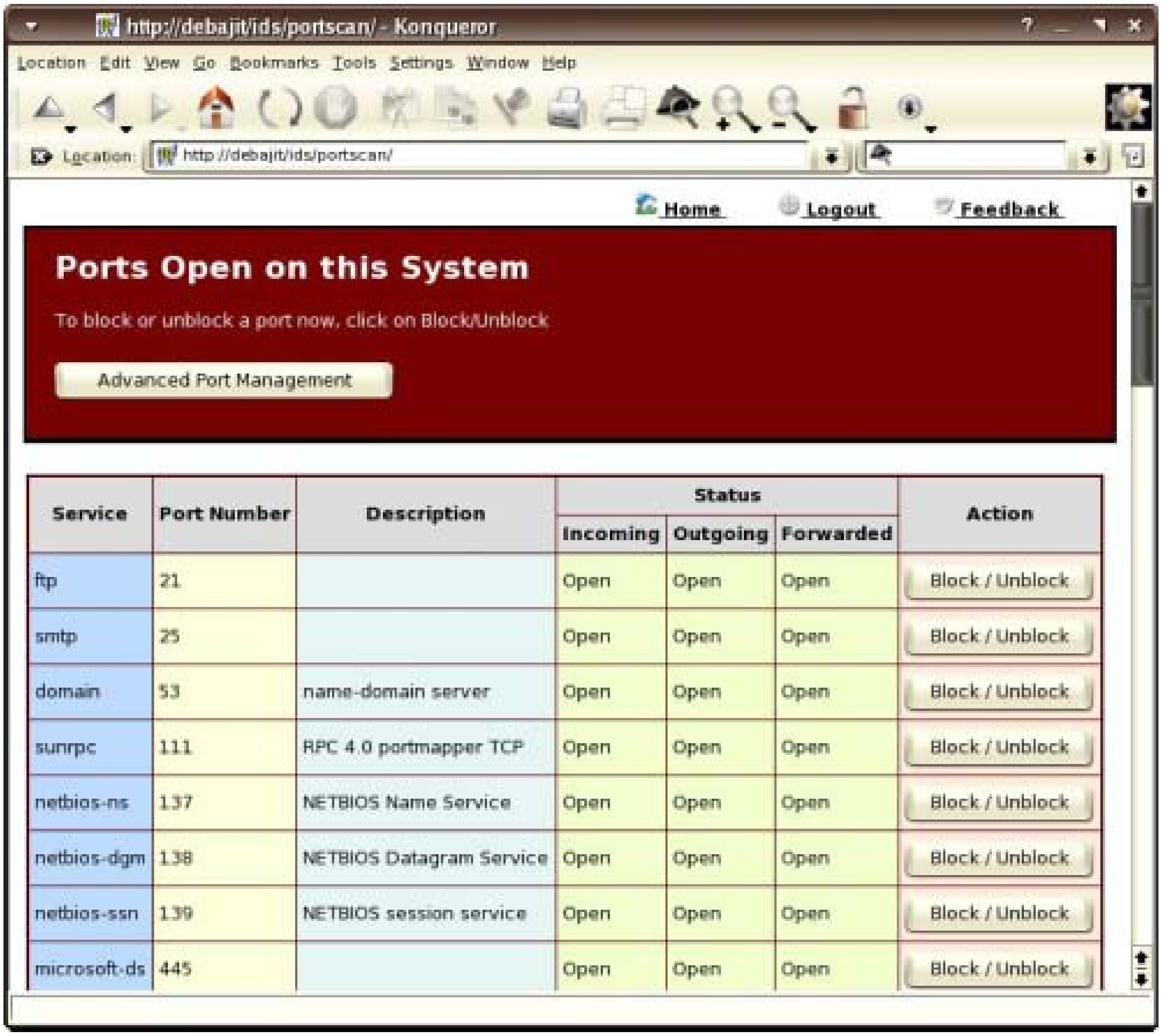
(1) Authorized people to use the IDS Remote management console are listed under '**Username**'.

(2) People not listed but avid to get listed, can be authorized, by clicking '**Add New user**' button.

(3) Password of particular user can be changed by clicking on '**Change Password**' button.

(4) Omnipotent user 'admin' can not be deleted by any user.

6.6 PORT SCANNING


http://debajit/ids/portscan/ - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/portscan/
Home
Logout
Feedback
Advanced Port Management

Ports Open on this System

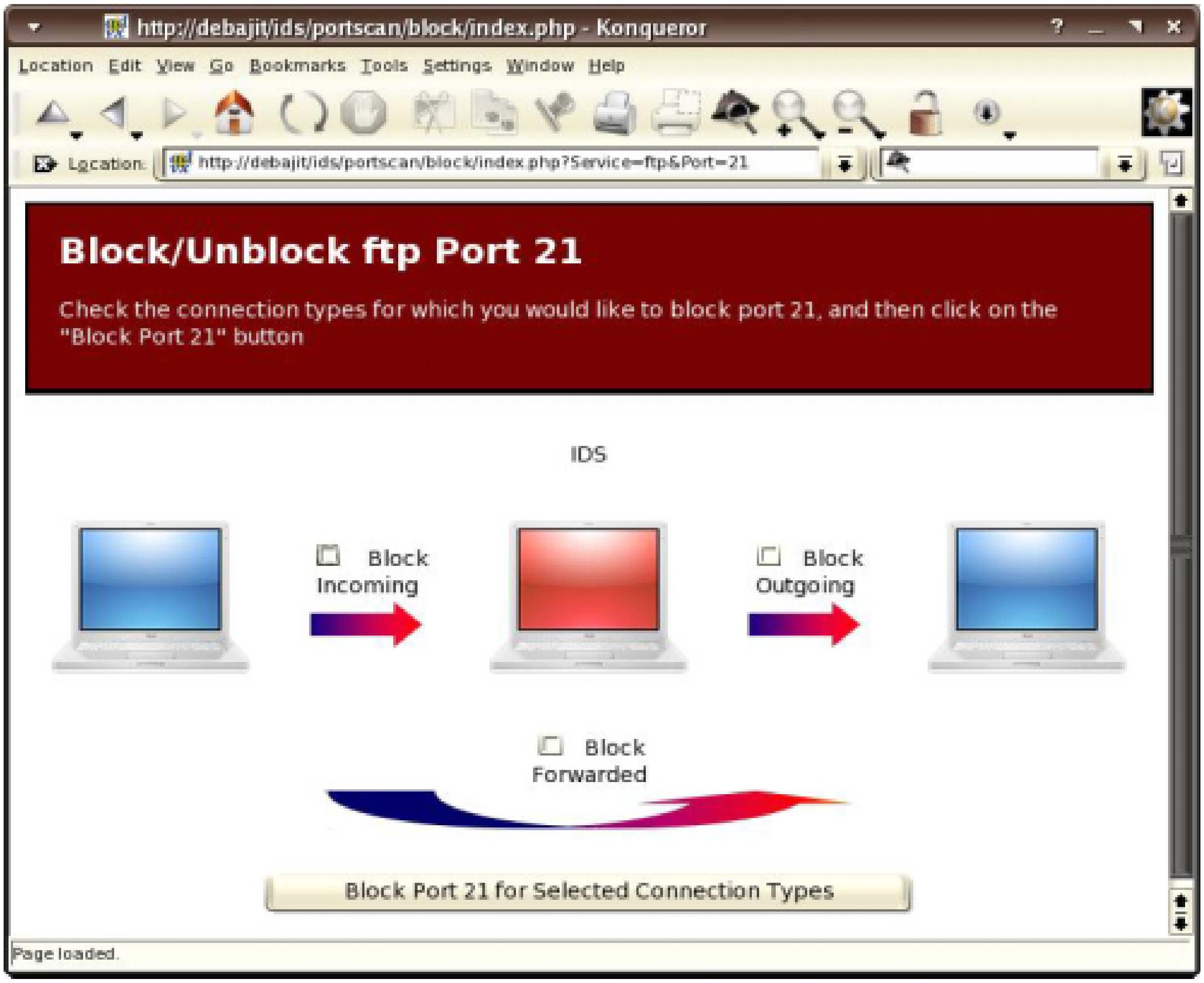
To block or unblock a port now, click on Block/Unblock

Service	Port Number	Description	Status			Action
			Incoming	Outgoing	Forwarded	
ftp	21		Open	Open	Open	Block / Unblock
smtp	25		Open	Open	Open	Block / Unblock
domain	53	name-domain server	Open	Open	Open	Block / Unblock
sunrpc	111	RPC 4.0 portmapper TCP	Open	Open	Open	Block / Unblock
netbios-ns	137	NETBIOS Name Service	Open	Open	Open	Block / Unblock
netbios-dgm	138	NETBIOS Datagram Service	Open	Open	Open	Block / Unblock
netbios-ssn	139	NETBIOS session service	Open	Open	Open	Block / Unblock
microsoft-ds	445		Open	Open	Open	Block / Unblock

The above figure shows the list of all ports which are opened

1) The main port scan page list all the ports along with the services which are opened in the system.
2) The user can block/unblock a particular port by just clicking the block/unblock Button and he/she will be directed to port blocking page.

PORT BLOCKING/UNBLOCKING


http://debajit/ids/portscan/block/index.php - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/portscan/block/index.php?Service=ftp&Port=21

Block/Unblock ftp Port 21

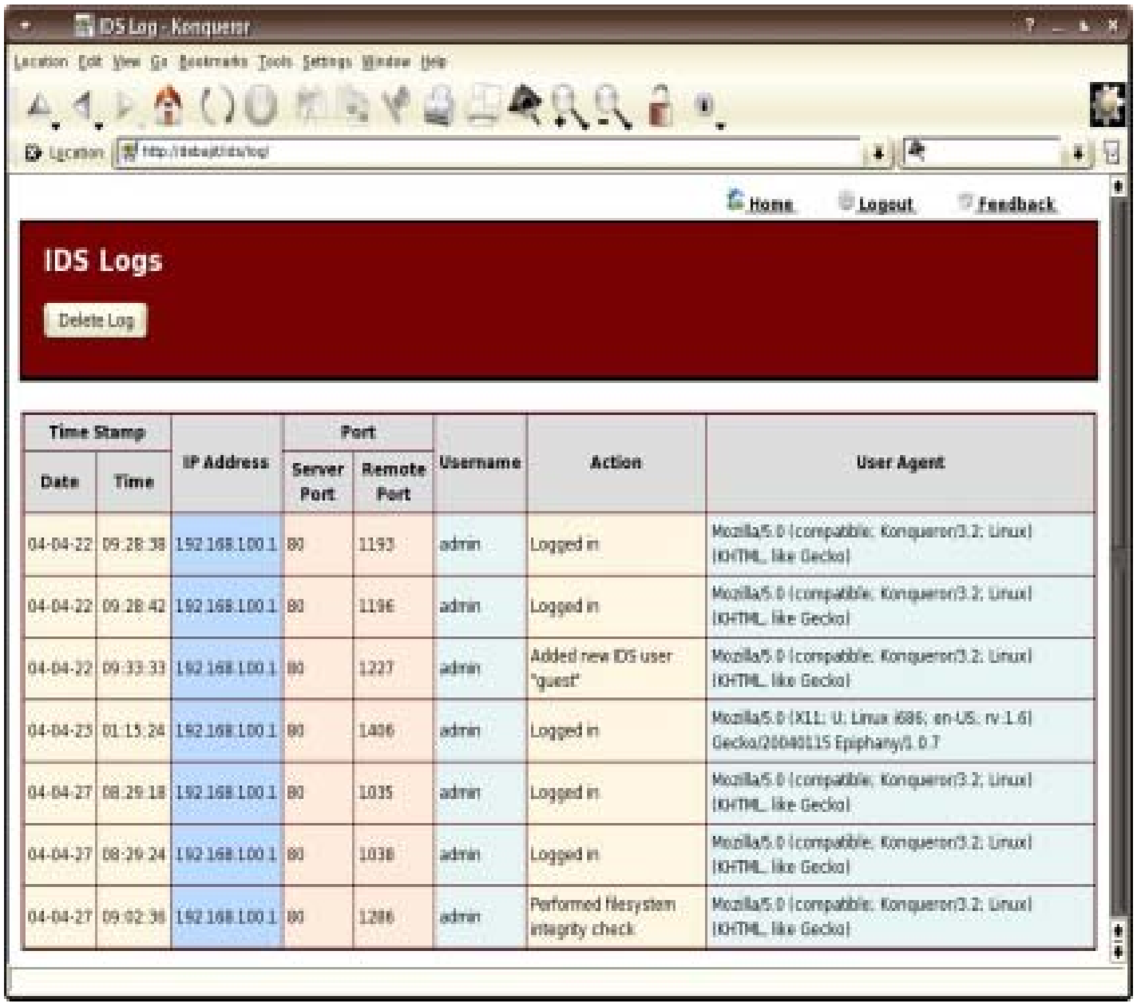
Check the connection types for which you would like to block port 21, and then click on the "Block Port 21" button

IDS
Block Incoming
Block Outgoing
Block Forwarded
Block Port 21 for Selected Connection Types
Page loaded.

The above figure shows how to block/unblock a port

1) The port no 21 which is ftp port has been selected to be blocked/unblocked by the user.
2) Check the connection for which you would like to block/unblock the port.
3) After the connection is being checked just click the block port no button for the selected connection types .
4) You will be directed to main page of port scanning module where u will see the port is blocked/unblocked for the selected connection types.

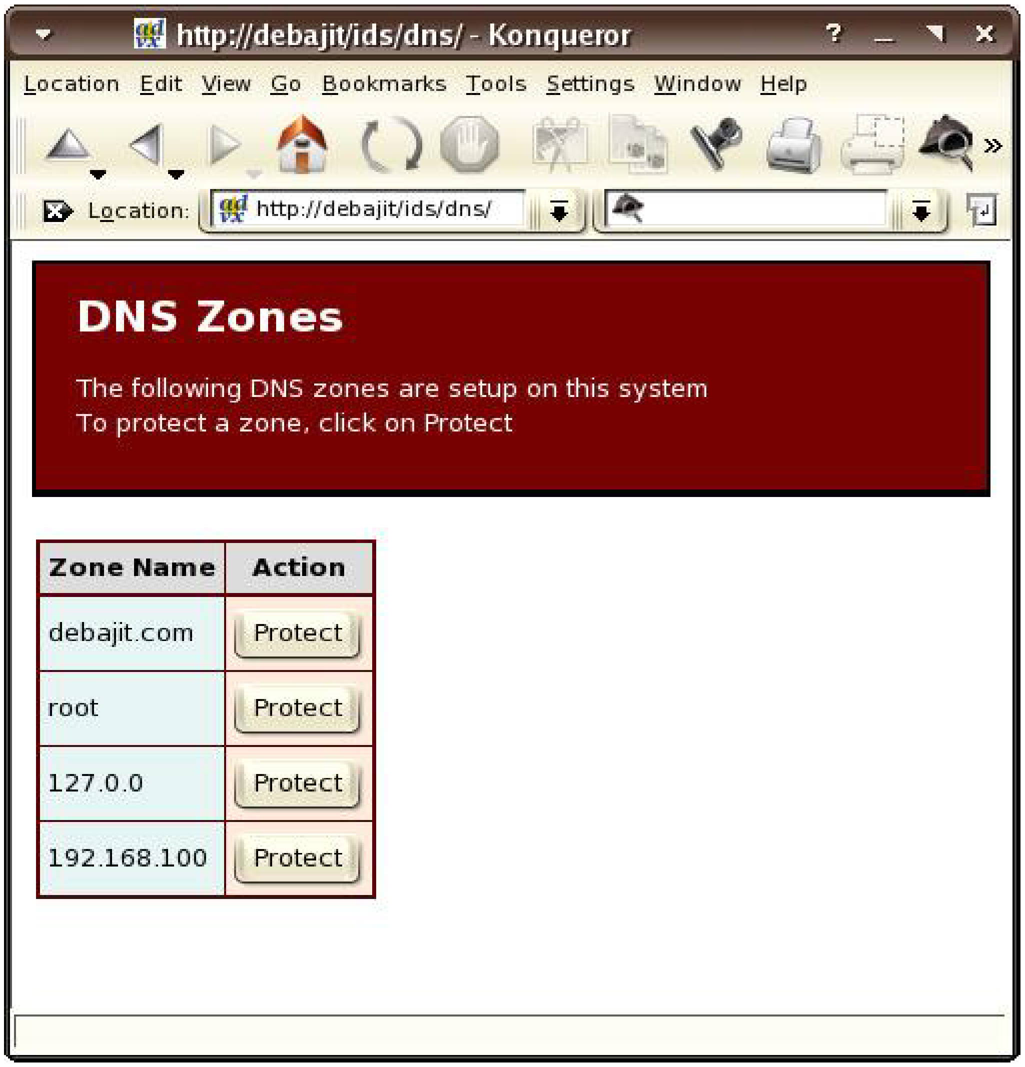
6.7 IDS LOGS


IDS Log - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location http://debajit/ids/log/
Home
Logout
Feedback
IDS Logs
Delete Log

Time Stamp		IP Address	Port		Username	Action	User Agent
Date	Time		Server Port	Remote Port			
04-04-22	09:28:38	192.168.100.1	80	1193	admin	Logged in	Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)
04-04-22	09:28:42	192.168.100.1	80	1196	admin	Logged in	Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)
04-04-22	09:33:33	192.168.100.1	80	1227	admin	Added new IDS user "guest"	Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)
04-04-23	01:15:24	192.168.100.1	80	1406	admin	Logged in	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040115 Epiphany/1.0.7
04-04-27	08:29:18	192.168.100.1	80	1035	admin	Logged in	Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)
04-04-27	08:29:24	192.168.100.1	80	1038	admin	Logged in	Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)
04-04-27	09:02:36	192.168.100.1	80	1286	admin	Performed filesystem integrity check	Mozilla/5.0 (compatible; Konqueror/3.2; Linux) (KHTML, like Gecko)

1) When user clicks logs he or she is directed to this page.
2) The above snapshot shows the log information for a user who is logged in.
3) It gives information the time ,date when a particular user has logged in..
4) Also gives information about the user name and the port through which he/she has logged in.
5) A particular user has facility to delete the logs by clicking the delete button and by doing so logs will be deleted from database.

6.8 DNS


http://debajit/ids/dns/ - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/dns/

DNS Zones

The following DNS zones are setup on this system
To protect a zone, click on Protect

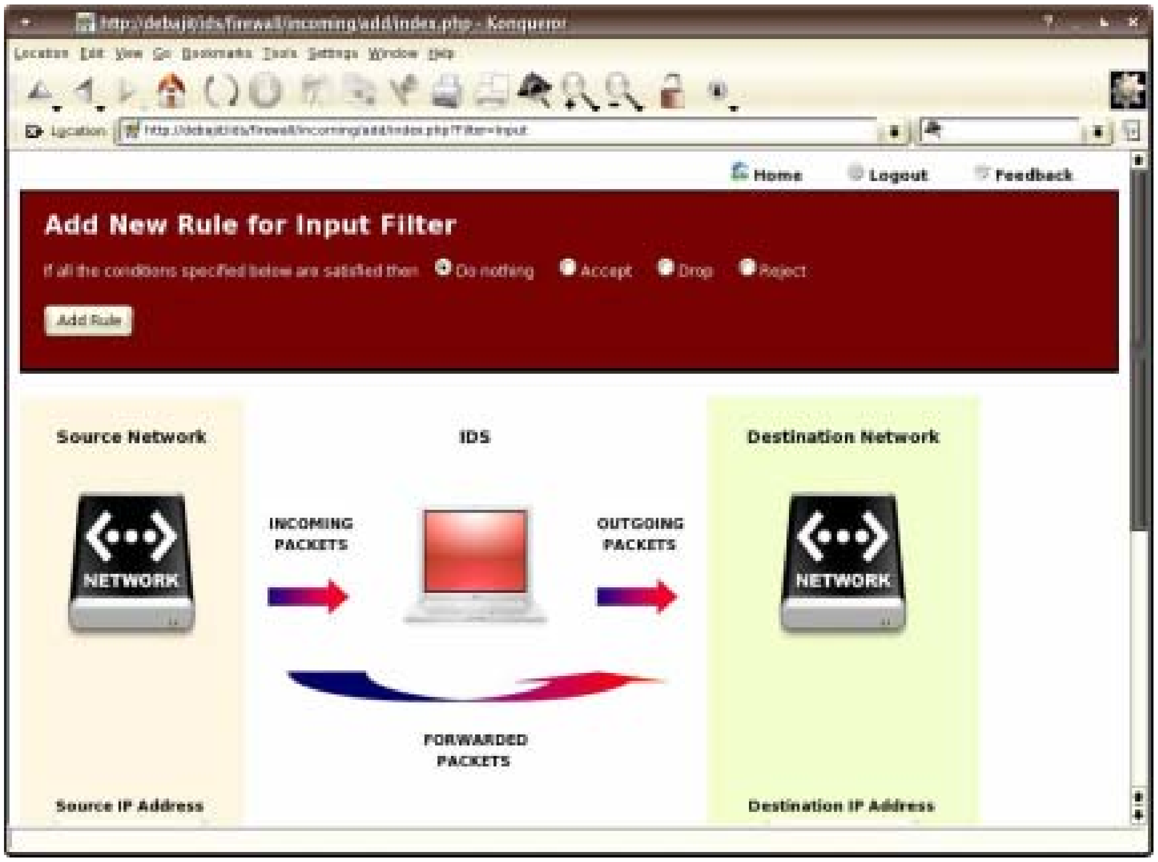
Zone Name	Action
debajit.com	Protect
root	Protect
127.0.0	Protect
192.168.100	Protect

(1) We get redirected to this page when DNS option is clicked by the user under Servers tab.

(2) Under ‘Zone Name’ all the existing zones in our system are listed.

(3) Clicking on ‘Protect’ button redirects us to the webpage where other details like ‘Security Level’ are accepted from the user after which the corresponding entry is done in the Tripwire Policy file.

6.9 FIREWALL



Add New Rule for Input Filter

If all the conditions specified below are satisfied then [✓ Do nothing] Accept Drop Reject

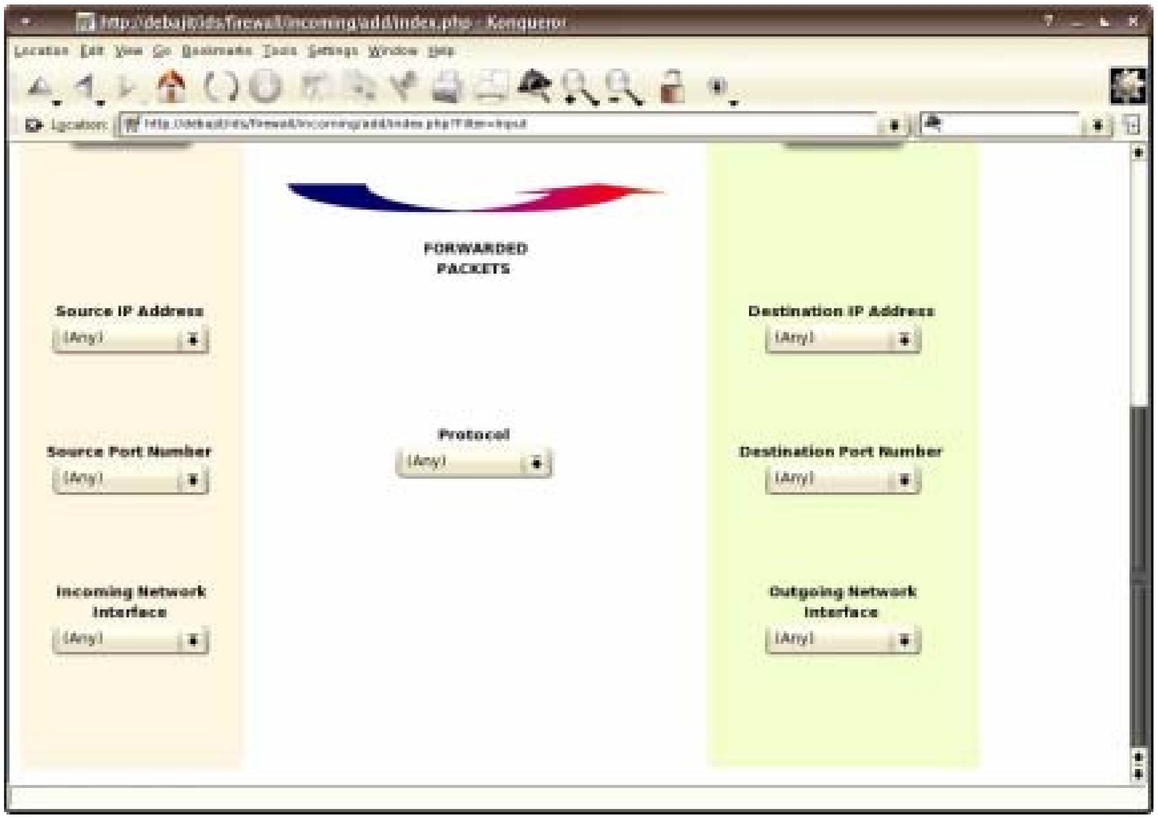
Add Rule

The above picture is the screenshot of the top half of firewall screen, which can be used to add a firewalling rule in the input chain.

(1). The radio buttons provided on top of the page are their to specify the action to be taken with the packets with satisfy all the options specified below.

(2). The user can add the firewalling rule specifying the appropriate options and clicking the Add Rule .

(3). The two combos at the bottom of the page are their to specify the Source IP Address and Destination IP Address.

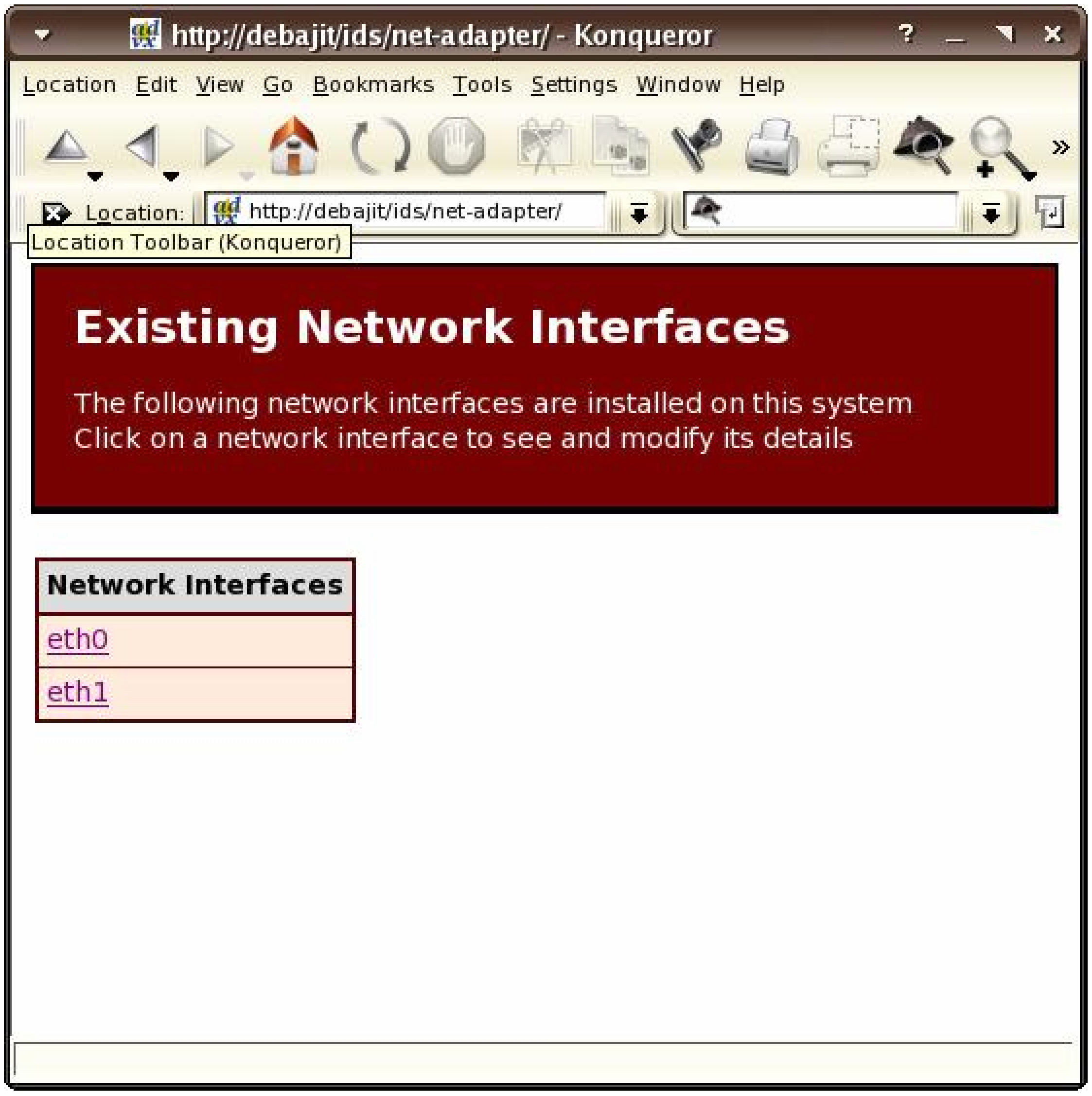

http://debajit/ids/firewall/incoming/add/index.php - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/firewall/incoming/add/index.php?Filter=Input
FORWARDED PACKETS
Source IP Address
(Any)
Destination IP Address
(Any)
Protocol
(Any)
Source Port Number
(Any)
Destination Port Number
(Any)
Incoming Network Interface
(Any)
Outgoing Network Interface
(Any)

The Above screenshot is second half of the above firewalling screen

(4) The user can specify various options through various combos. As soon as the combo is activated to (Equals, Not Equals, Any) the text boxes above the combos become visible in which we can specify the options.

Similar screen are there for other IP chains i.e. Forward or Outgoing chains also.

6.10 NETWORK ADAPTER INFORMATION


http://debajit/ids/net-adapter/ - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: http://debajit/ids/net-adapter/
Location Toolbar (Konqueror)

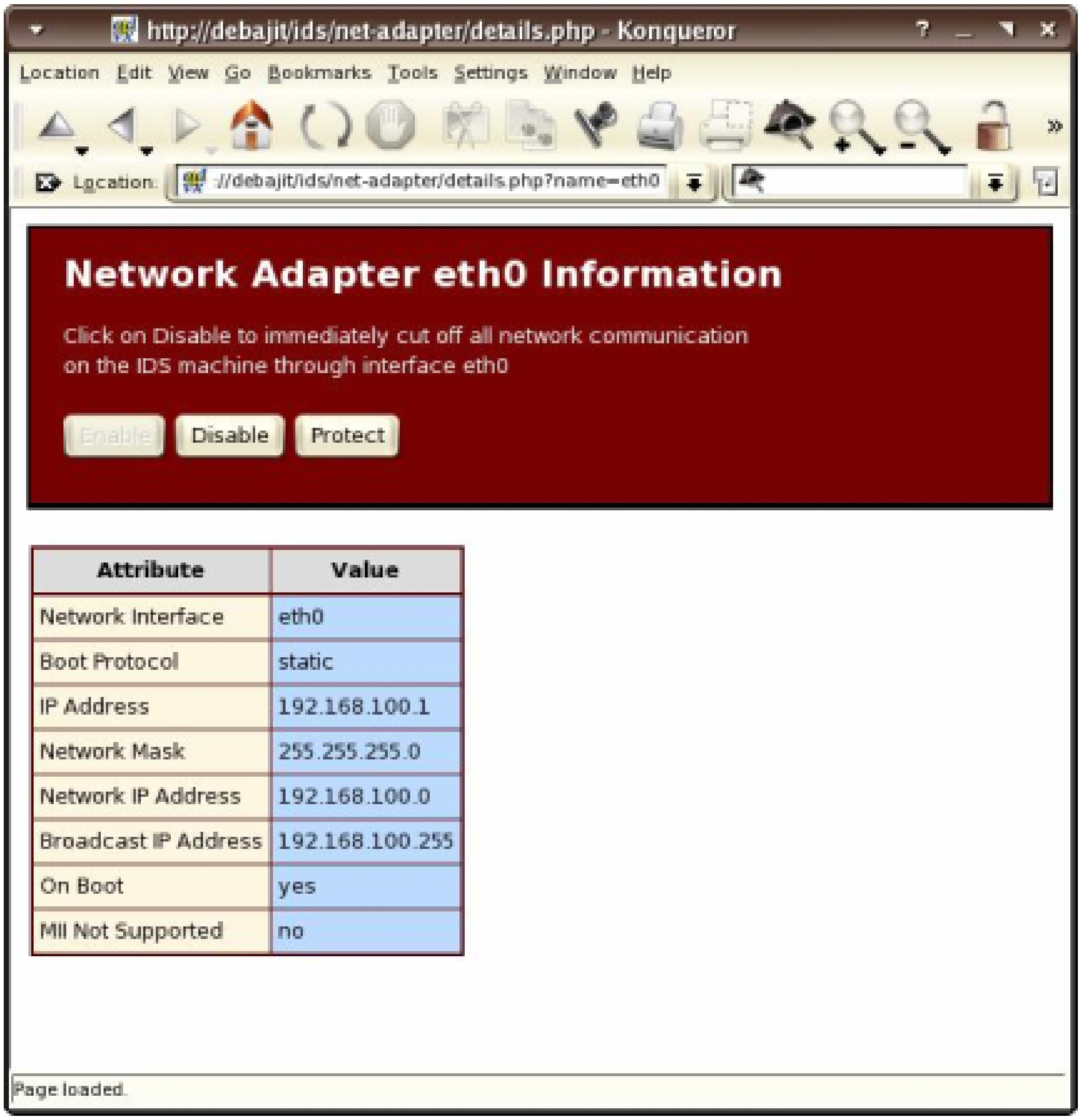
Existing Network Interfaces

The following network interfaces are installed on this system
Click on a network interface to see and modify its details

Network Interfaces
eth0
eth1

When the user selects the Network Adapter Information option then the above screen is displayed

Here the user can select the appropriate network interface to see details of the network interface and even to protect the interface.


http://debajit/ids/net-adapter/details.php - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: //debajit/ids/net-adapter/details.php?name=eth0

Network Adapter eth0 Information

Click on Disable to immediately cut off all network communication on the IDS machine through interface eth0

Attribute	Value
Network Interface	eth0
Boot Protocol	static
IP Address	192.168.100.1
Network Mask	255.255.255.0
Network IP Address	192.168.100.0
Broadcast IP Address	192.168.100.255
On Boot	yes
MII Not Supported	no

(1) This page displays the details of the selected network interface.

(2) The user can click on the disable button to instantly bring down the selected interface. If the user is working on that interface he will be immediately disconnected.

(3) If the network interface is already disabled , then the "**enable**" button is enabled and the user can click on this button to activate or bring up that interface again.

(4) Third button is of protect in which the the corresponding scripts are added to the database .Here user can specify the level of security to be applied to the interface.

CHAPTER 7
TESTING PROCEDURES

7.1 TESTING TRIPWIRE

TEST CASE 1: VERIFYING THE COMPARE MODE OF TRIPWIRE

An initial database is generated during installation of Tripwire.

(1) An arbitrary test file (text file /tmp/abc) is added to the Tripwire policy file. A known security level, $(SEC_CRIT) is used. Some of the attributes which this security level fingerprints are:

- (a) Permission bits
- (b) File type
- (c) File size
- (d) Modification timestamp

(2) A new encrypted Tripwire database is then generated.

(3) An integrity check is now run manually from the command line. This will generate an encrypted filesystem integrity report.

(4) This is then decrypted and viewed through the Tripwire binary *twprint.*

(5) This report does not lists the test file as has not been tampered with.

(6) The content of the file is changed and the following modifications are made:

ATTRIBUTE	EXISTING VALUE	MODIFIED VALUE
Permission bits	-rw-r--r--	-rwxr-xr-x
File size	11 bytes	25 bytes

This will also automatically change the modification timestamp of the test file.

(7) Again an integrity check is run and the generated report is viewed.

(8) The modified attributes are now listed in the report , confirming the proper working of tripwire.

TEST CASE 2: VERIFYING THE UPDATE MODE OF TRIPWIRE

(1) Once the modifications made to the test file are reported, initialize the tripwire update mode using the tripwire binary *tripwire*

(2) Tripwire now prompts the user to either apply or ignore the modifications. We now apply modifications to the test file so they are not reflected in the tripwire database.

(3) A new encrypted Tripwire database is then generated.

(4) An integrity check is now run manually from the command line. This will generate an encrypted filesystem integrity report.

(5) This is then decrypted and viewed through the Tripwire binary *twprint.*

(6) Now in this report test file is not reported to be modified, hence confirming the proper working of tripwire update mode.

7.2 TESTING USER AUTHENTICATION

TEST CASE: BLOCKING ACCESS WITHOUT LOGIN

Despite an initial login screen a malicious user could directly access any other webpage in the remote management console by giving its address. This is prevented by incorporating an authentication check on every page of the console.

(1) The file system integrity checking section is available at the address http://debajit/ids/integrity/ and the IDS login screen can be accessed at http://debajit/ids/

(2) Now we directly access the file system integrity checking section bypassing the initial login screen

(3) An error page is displayed and the user is denied access providing him a link to the login screen.

This test case verifies the proper functioning of user authentication section of the IDS.

7.3 TESTING PORT BLOCKING & UNBLOCKING

TEST CASE: BLOCKING A PORT

(1) We opened the port section of the console which lists all the open ports and their status.

(2) To test the proper functionality we blocked several standard services associated with fixed ports. For instance we blocked port number 21 (FTP) which was initially unblocked for all types of packets.

(3) We blocked only the incoming packets through port 21.

(4) We then tried to ftp the system on which IDS was running (192.168.100.1) from a remote machine (192.168.100.4), but were denied the permission to ftp the IDS machine.

(5) Next we tried to ftp a remote machine (192.168.100.4) from the system on which IDS was running (192.168.100.1), and were successfully able to establish the ftp connection with the remote machine. This confirms that the outgoing packets through port 21 are still unblocked.

(6) Now, we again opened the port section and blocked all (incoming, outgoing, forwarded) packets through port 21.

(7) This time, we were not granted permission to ftp even the remote machine (192.168.100.4).

(8) Then we sent a ftp request from a remote machine on the other network (192.168.200.2) connected to the IDS (acting as a router) to test the forwarded packets and we were denied a ftp connection.

7.4 TESTING FIREWALL MODULE

TEST CASE: CHECKING PACKET FILTERING

(1) We supplied following firewalling options to the firewall module:

a)	Source IP address	192.168.100.2
b)	Destination IP address	192.168.200.2
c)	Source port	any

d)	Destination port	any
e)	Protocol	any
f)	Source Interface	eth0
g)	Destination Interface	eth1

(2) We then tried to send packets from source machine to destination machine.

(3) We tried to ping from either machine to the other and were reach the host as we had denied packets of all protocols through all ports as per our test firewalling options. We were even not able to establish a ftp or telnet or any other sort of connection between the two machines.

(4) We then edited the above test firewalling rule and changed the protocol from ‘any’ to ‘ICMP’.

(5) Now when we tried to establish various connections through ftp, etc, we were successful but were still unable to ping the machines as ICMP packets were blocked.

This test case confirmed that proper packet filtering is achieved.

7.5 TESTING THE FILESYSTEM INTEGRITY CHECK MODULE

TEST CASE 1: CHECKING FILE PROTECTION

(1) We opened the file system integrity check module, selected a file(say /tmp/abc) to protect along with the security level Log Files and the desired file type.

(2) We noted down the attributes fingerprinted for the selected security level (here Log files)

(3) We now click **“run integrity check now“** button and noticed the test file did not figure anywhere in the report as was till now not tampered.

(4) Then we changed the permission bits and the UID of the test file as these attributes were a part of the list of all attributes fingerprinted for the security level.

(5) We then run the integrity check again and notified that the test file was listed in the modified objects list of the given file type section.

(6) When we viewed the modifications we found exactly same changes were listed, Conforming the proper functioning of this module.

(7) We then deleted the test file and immediately the file was listed in the same section's removed objects list.

TEST CASE 2: VERIFYING MODIFICATIONS APPLIED

(1) Again we repeated the same steps (1)-(6) as in the above test case for two test files ,"***/tmp/a***" and "***/tmp/b***".

(2) Now we clicked on the "Apply Modifications" button for the first file only.

(3) Now again we run the integrity check using the given button and viewed the report. This time the first file was missing from the given section's modified objects list. On the other hand the second file was still listed there as we didn't apply the modifications.

This verifies that the tripwire database was actually updated on applying the modifications.

CONCLUSION

Intrusion detection is a process that must be executed by system administrators in order to maintain secure networks. An administrator must understand the importance of protecting his/her network, how exploited vulnerabilities can bring a system to it's knees, and how to react to security incidents. System administrators must stay informed of all system advisories, flaws, and software updates. Not taking appropriate actions to fix known problems can prove to be fatal to network servers. As our society begins depending more on network systems, information security will become more of an issue. If network administrators do not remain informed of software updates and fail to closely monitor their servers, network security will remain to be problematic. Intrusion detection is a necessary process that must be fully understood and executed to maintain network security.

The product like this having a well defined and unified user friendly interface will make it much easier to use, administer and maintain secure networks.

FUTURE SCOPE OF THE PRODUCT

Our project checks the filesystem integrity of the system at a predefined frequency. This product at best detects the intrusion and reports the administrator who can then take preventive cum recovery methods. The basic <missing> characteristic of such an approach is that it might be too late by the time the administrator is reported and hence this product can be extended to automatically block an intruder from getting into the system i.e. inherently act as an IPS also. One possible approach could be monitoring and intercepting system calls like open().

REFERENCES

BIBLIOGRAPHY

(1) Scott Mann, Ellen Mitchell, Mitchell Krell, **Linux System Security**, Pearson Education, 2003

(2) Brian Hatch, James Lee, **Hacking Linux Exposed (Second Edition)**, McGraw-Hill, 2003

(3) Leon Atkinson, Zeev Suraski, **PHP 5.0 Core Programming,** Addison-Wesley

(4) Daniel P. Bovet, Marco Cesati, **Understanding the Linux Kernel**, O'Reilly & Associates, 2002

(5) Sumitabha Das, **Your UNIX: The Ultimate Guide**, Tata McGraw Hill, 2002

(6) Andrew S. Tanenbaum, **Computer Networks**, Pearson Education, 2003

(5) Ankit Fadia, **Network Security**, MacMillan

WEB REFERENCES

(1) http://www.freshmeat.net/

(2) http://www.planet-source-code.com/

(3) http://www.zend.com/

(4) http://www.php.net/

(5) http://www.phpclass.net/

(6) http://www.linuxsecurity.com/

(7) http://www.linuxforum.com/

(8) http://www.ids.org/

(9) http://www.tripwire.org/

(10) http://www.tripwire.com/