Aniruddha P Tekade, Pravin Gurjar, Pankaj R. Ingle, Dr.B.B.Meshram /International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 1, January -February 2013, pp.1854-1860

Ethical Hacking in Linux Environment

Aniruddha P Tekade, Pravin Gurjar, Pankaj R. Ingle, Dr.B.B.Meshram
Department of Computer Engineering Veermata Jijabai Technological Institute
Matunga, Mumbai

Abstract —

"Ethical Hacking" which attempts to pro-actively increase security protection by identifying and patching known security vulnerabilities on systems owned by other parties. Ethical hackers may beta test unreleased software, stress test released software, and scan networks of computers for vulnerabilities. Ethical hacking can be defined as the practice of hacking without no malicious intention, rather evaluate target system with a hackers perspectives.[1][7] Hacking is a process to bypass the security mechanisms of an information system or network. In common usage, hacker is a generic term for a computer criminal. Hacking is an unprivileged usage of computer and network resources. The term "hacker" originally meant a very gifted programmer. In recent years though, with easier access to multiple systems, it now has negative implications.

Keywords **—Ethical standards[2], Penetration, Exploits, Philosophy of Hacking, Emanations, Vandalism.**

I. INTRODUCTION

This paper aims at putting forward the basic concept of ethical hacking and difference between a hacker and cracker, root philosophy of hacking, the approach that differs in the thought processes of hackers and programmers and reveals secretes of hacking under Linux domain. As we all are aware that data and Computer Communications are hot subjects and getting hotter every day[5]. We see it when we turn on our television, cordless or cell phone, or the computer when we get our email. It has provided us lightning speed conveniences that our grandparents could only imagine when they went to the movies to see Buck Rogers or Dick Tracy. However, what they could not have imagined was the "dark side" that comes along with these technological advances. The rapid growth of the Internet has brought many constructive and valued solutions for our lives such as e-commerce, electronic communication, and new areas for research and information sharing. However, like many other technological advancements, there is also an issue of growing number of criminal hackers. [4]Businesses are scared of computer experts who will penetrate into their web server and change their logo, steal their private emails or credit card numbers, or put in software that will quietly transmit their organization's data to somebody in another country.

Hackers are commonly known as bad or terrible people in our society. They are also known as crackers or black hat guys. The reason is that majority of computer users are somehow victim of malicious activities by other users who are outstandingly experts in computers. The important thing to understand is not all the hackers are bad as some people are doing penetration of a system in the limits of **ethical standards** to understand the vulnerabilities in their system or their clients system, also called white hat hackers. Hence the term "**Ethical Standards**" actually refers to the consideration if the person performing hacking has a valid intention or not. If he or she just wants to access the target system with an illegal intention and misuse the data explored, can be termed as the cracker whereas the ethical hacker always intends for test that yields the vulnerabilities of the system as the output through the process of hacking. In ethical hacking, for example, a network administrator might use the encrypted password file and a "cracking" program to determine who has not picked a good password. The need is to train our computer science students with ethical hacking techniques, so that they can fight against criminal hackers. Because ethical hackers believe that one can best protect systems by probing them while causing no damage and subsequently fixing the vulnerabilities found.

II. RELATED WORK

The new generation of hackers are turning open source into a powerful force in today's computing world. They are the heirs to an earlier hacking culture that thrived in the 1960s and 1970s when computers were still new part of community that believed software should be shared and all would benefit as a result. These experts programmers and networking wizards trace their lineage back to the first time-sharing minicomputers and the earliest ARPAnet experiments. The members of this community coined the term "Hacker". Hackers build the internet and made the UNIX operating system what is it today. Hackers run the Usenet and make the World Wide Web work.

1. Hackers sparked the open source revolution-

In 1991, Linus Torvalds sent a posting to an Internet newsgroup, asking for advice on how to make a better operating system. His project was a Hobby, he said, and would never be 'big and professional'[12].

In 1994, the first working version of Linux was distributed. Marleen Wynants and Jan Cornelis, while discussing the economic, social, and cultural impact of Free and Open Source Software in their paper "How Open Source is the Future?" suggest that Linux was more than just a toy of hackers. Propelled by Linux, the open source culture surface from its underground location.

In the spring of 1997, a group of leaders in the free software community assembled in California. This group included Eric Raymond, Tim O'Reilly and VA Research president Larry Augustin, among others. Basically "HACKING" is a loaded term ~ the distinction between hacking and cracking is not universal. The concept of hacking is yielded from the dictionary meaning of "hack" as a verb "to chop or cut roughly, to make rough cuts" as in programming using ad hoc methods based on experience without necessarily having a formal plan or methodology for evaluation. In another perspective and being a little antisocial, hacking is clever but unstructured programming solution to a problem.

2. What is the difference? - In our pyramid of human brain, information is stored in terms of chemicals and genetic substances, as in the same way, two majorly used operating systems viz. Linux and Windows have their owned file system like NTFS and FAT or ext and both of them are considered as robust based on their user's perspectives and specification. But the fact lies in that, every operating system can be cracked and hacked. So what is this difference between the hacks and cracks? Ethical hackers simulate how an attacker with no inside knowledge of a system might penetrate and believe their activities benefit society by exposing system weaknesses – stressing that if they can break these systems so could terrorists. The result is not only enhanced local security for the ethical hacker but also overall operating domain security. The white paper also tries to elaborate about the basic tools and techniques that are widely used by a mass of unexplored group of hackers in the world their methodology of working.

3. Philosophy – The backend thoughts that tempts a person to be either a hacker or a cracker lies in the approach he puts his directions. Henceforth what the hacker digests is this.

- Ethical hackers believe one can best protect systems by probing them while causing no damage and subsequently fixing the vulnerabilities found.
- Ethical hackers simulate how an attacker with no inside knowledge of a system might try to penetrate and believe their activities benefit society by exposing system weaknesses - stressing that if they can break these systems so could terrorists.
- Ethical hackers use their knowledge as risk management techniques.

Whereas influenced with an invalid intentions people think –

- Hacker or cracker are clever but an unstructured programmer and believe the same but with invalid intentions.
- Crackers break into (crack) systems with malicious intent. They are out for personal gain: fame, profit, and even revenge. They modify, delete, and steal critical information, often making other people miserable.
- Hackers have a destructive R&D approach to break different software, systems, and networks policies.

While revealing more and more about the ethical hacking the journey may stuck up at the point where human mind is made thinking about how a hacker and a normal guy differs in approaches when both of them knows same coding patterns and technologies. But the truth turns a little lifeway that turns us to believe how a black or white hat guy breaks the boundaries of the programming.

A typical developer's methodology[12]:

✓ Developers are under pressure to follow standard solutions, or the path of least resistance to "just making it works." As long as a trick works, detailed understanding is often considered optional. Consequently, they might not realize the effects of deviating from the beaten path.

✓ Developers tend to be implicitly trained away from exploring underlying APIs because the extra time investment rarely pays off.

✓ Developers often receive a limited view of the API, with few or hardly any details about its implementation.

✓ Developers are de facto trained to ignore or avoid infrequent border cases and might not understand their effects.

✓ Developers might receive explicit directions to ignore specific problems as being in other developer's domains.

✓ Developers often lack tools for examining the full state of the system, let alone changing it outside of the limited API.

A Hackers methodology:

✓ Hackers tends to treat special and border cases of standards as essential and invest significant time in reading the appropriate documentation (which is not a good survival skill for most industrial or curricular tasks).

✓ Hackers insist on understanding the underlying API's implementation and exploring it to confirm the documentation's claims.

✓ As a matter of course, hackers second-guess the implementer's logic (this is one reason for preferring developer-addressed RFCs to other forms of documentation). Hackers reflect on and explore the effects of deviating from standard tutorials.

✓ Hackers insist on tools that let them examine the full state of the system across interface layers and modify this state, bypassing the standard development API. If such tools do not exist, developing them becomes a top priority.

4. Understanding the need to hack your own system

To catch a thief, think like a thief. That's the basis for ethical hacking. Protecting your systems from the bad guys and not just the generic vulnerabilities that everyone knows about is the need of the hour and is absolutely critical. When you know hacker tricks, you can understand how vulnerable your systems are. Hacking preys on weak security practices and undisclosed vulnerabilities. An *exploit* is a piece of malware code that takes advantage of a newly-announced or otherwise unpatched vulnerability in a software application ex; OS, web browsers, plug-ins etc.

When "ethical" is placed in front of the term hacking it denotes the moral activity. Unethical hacking has no permission to intrude the systems. Ethical hacking includes permissions to intrude such as contracted consulting services, hacking contests, and beta testing of information security or any IT project.

HACKING IN LINUX OPERATING SYSTEMS[7]

In the last decade the open source movement has been a vital source of innovation affecting software development. However, open source community practices have provoked a
Debate on software quality—namely, is open source software's quality better than that of its closed-source counterpart? Studies have attempted to correlate metrics with software performance or validate that metrics can actually predict software systems' fault proneness.

Open Source Software

Where you can define closed-source software as a product created using traditional software development methods, the definition of open source software isn't always straightforward. This is because a software product can take at least three paths to become open source. For example, a collaborating open source community developed the Linux kernel; an individual created PGP (Pretty Good Privacy) and the Mozilla browser were originally developed as proprietary software. One implication of this is that any conclusions about Linux might not hold true for all open source products. But being an initiative taker, open source communities make society Linux strong system software. A hacker always needs to figure out the vulnerabilities in the victim system.

A) Local Access Control in Linux Environment

From a Physical Security (PHYSSEC) perspective, problems do not really begin until attackers have their hands on a machine. Having suitable access controls to prevent direct access and policies in place to prevent social engineering will help ensure that attackers are kept at a safe distance. Linux is a robust OS, but it is still vulnerable to hardware dangers that may lead to damage on its physical drives or power losses that may cause data corruption. Therefore, in addition to access controls, server rooms should include the following items to ensure integrity and availability and provide protections from power outages, power anomalies, floods, and so on[7].

•*Console Access*

Stealing data using a Bootable Linux CD:

1. Reboot the system and configure it to boot from the CD-ROM.
2. Boot into the bootable Linux distro.
3. Open a root command shell.
4. Create a mount point by typing **mkdir mountpoint,** which will create a directory called mount point.
5. This is where the file system will be mounted.
6. Determine the type of hard disks (SCSI or IDE) on the system. [sda, sdb, sdc, and so on for SCSI, & hda, hdb, hdc, and so on for IDE] To determine the disk type, type **fdisk –l** or **look** through the output of the **dmesg** command.
7. Determine the partition on the disk to be mounted. Partitions on the disk are represented as sda1, sda2, sda2, and so on.
8. Identifying the correct partition that contains the ***/etc/shadow*** fi le (always the root "/" partition). It is usually one of the first three partitions.
9. Type **mount /dev/*sda# mountpoint***, where /dev/sda# is your root partition (sda1, sda2, sda3,…), and mountpoint is the directory you created.
10. Change to the /etc directory on your root partition by typing **cd mountpoint/** etc

11. Use your favorite text editor (such as vi) to open the etc/shadow file for editing.
12. Scroll down to the line containing the root's information, which looks something like: root:qDlrwz/E8RSKw:13659:0:99999:7:::
13. Delete everything between the first and second colons, so the line, resembles this one: root::13659:0:99999:7:::
14. Save the file and exit you editor.
15. Type cd to return to the home directory.
16. Type umount mountpoint to unmount the target file system.
17. Type reboot to reboot the system and remove the bootable Linux distribution CD from the drive.
18. Now the system can be accessed as root with no password (or the known password).

B) Chrooting Directory

The amount of work that goes into securing a system can be partially mitigated by taking advantage of the *chrooting abilities built into certain applications or by using the* chroot feature that is included or can be compiled into Linux. Chroot *is a combination of two* words: *change and root.* It creates a sandboxed, virtual directory that is used to provide a user or an application access to only a limited subset of resources. Certain daemons, such as FTP and SSH, have the built-in or add-in ability to sandbox users in a carefully crafted "chrooted" environment.

Identifying Dependencies: The process of identifying and copying application dependencies and configuration files can be painstakingly performed using various Linux tools, such as the following.

Strace: A utility designed to trace all syscalls and executable makes. It will enumerate all files (configuration files, library dependencies, open files, output files) for a given executable. It shows voluminous output as it systematically steps through a binary as it executes.

•Privilege Escalations
•File Permission and Attributes
•Chrooting in system directories
•Hacking Local Passwords
•Disabling Bootable CD's and Bios Password

C) Privilege Escalation

We have described ways that attackers can compromise a system due to lack of physical access controls on or surrounding a system. Instead of aiming only to prevent
physical access to the machine or direct access to its drives, you must also consider how to safely allow semitrusted users some level of access to a machine, but not give them greater permissions than necessary. You must try to prevent users from escalating their privileges themselves and gaining access to unintended resources. Having said that, Linux systems often require a user be able to elevate his or her own privileges from time to time, when executing certain commands. Sudo is a utility that grant granular access to commands that users can run with elevated permissions. When using or administering a Linux box, you frequently need to switch back and forth between performing administrative-type tasks requiring enhanced permissions and
regular-type tasks only needing basic user permissions. It would be ineffective to operate using a basic user account all of the time and unwise to do everything as root. Due to the
restrictions placed on standard user accounts and the number of steps involved in switching back and forth between accounts, not to mention the irritation caused by the path changing every time, the tendency is to just log in to the system as the superuser and perform all the tasks from start to finish. This is very problematic.

D) Restrict System Calls with Systrace Interactive Policies

One of the most powerful system access controls is the Systrace utility that allows enforcement of interactive policies. Proper utilization of this utility can replace other access controls, or be added to them, as part of a defense-in-depth architecture. It essentially creates a virtual chrooted environment where access to system resources can be specifically permitted or denied for a particular application. The Systrace utility has
three primary functions:
• Intrusion detection
• Non-interactive policy enforcement
• Privilege elevation

Intrusion Detection The Systrace utility enables administrative personnel to monitor daemons (especially useful if done on remote machines) and generate warnings for
system calls that identify operations not defined by an existing policy. This allows administrators to create profiles for normal daemon operations on a particular system and generate alerts for any abnormal activity.

Noninteractive Policy Enforcement (aka IPS) Beyond the ability for Systrace to generate alerts for system calls not included in a particular policy, you can also use it to prevent
them. Systrace can be configured to deny any activity not explicitly defined in an active policy.

Privilege Elevation Instead of configuring SetUID/SUID/SGID bits, which can essentially create built-in vulnerabilities, Systrace can be used

to execute an application without persistent permissions, as it only escalates permissions to the desired level when necessary. Furthermore, Systrace only elevates privileges in a precise, fine-grained manner, specifically for the particular operations that require them.

Proposed System

The proposed system explains about how to hack a encrypted wireless computer network (WLAN) The system elaborates about the detailed procedure for hacking and gaining access to any encrypted Wireless Network(WAN) in three steps – preparing Device for attack, crack WEP encryption, crack WPA encryption

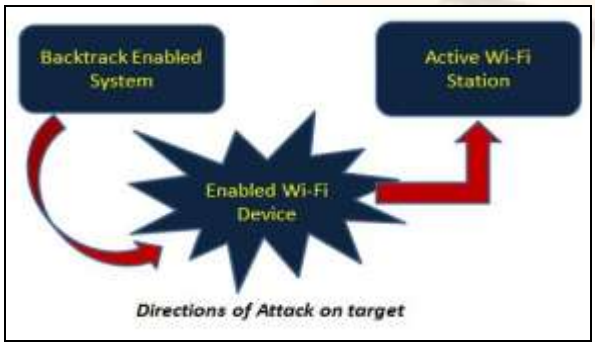
Existing Work

The work that lies in the background of this Wireless Network hacking through Linux flavored OS has not yet been proposed successfully because of the unavailable result statistics. Only the strong probabilities have been drawn out based on some previously proposed mathematical tools and theorems. Along with this many of the tools that come up in build in the Linux like BACKTRACK[7] have been suggested to perform.

Drawbacks in this system

1. Hacking process is totally practical and less theory prerequisites oriented.
2. Every technique of hacking born through the use security of less principled counterparts , here no such case history to consider
3. Hacking cannot be tested over some kind of matrix measures or on any other criteria that are generally considered valid for other kind of security related researches.
4. No analytical statistics are generated to prove the complexity and risk factors involved in this methodology.
5. The suggested tools have not been verified ever via actual implementation
6. Hence, overall risk factor is exponential.

Proposed System Architecture



The proposed system explains about the procedure for hacking and gaining access to any encrypted Wireless Network(WAN).

Requirement

1. Hacking Encrypted Wireless Network
2. Backtrack 5.0 installed hardware machine
3. System assembled with an Wi-Fi device or port
4. A Wi-Fi station

Algorithm for proposed system

This algorithm contains 3 basic steps-

1. Prepare a Wi-Fi device for an attack
2. Cracking WEP Encryption
3. Cracking WPA Encryption

Step I) Prepare the Wi-Fi device for an attack

1. Start with a new terminal
2. Get access of the system in privileged mode "*root*"
3. Check the status of the Wi-Fi device
4. If (Wi-Fi device = installed) Then –
    a. Turn off the "*Monitor Mode*" of W-Fi device
    b. Change the mac address of the computer
    c. Enable *Monitor Mode* of W-Fi device again
    d. Prepare Wi-Fi device for an attack
5. If (Wi-Fi device != installed) Then
    a. Configure a new Wi-Fi device
    b. Repeat Step # 3
6. Halt with exiting the terminal

Step II) Crack WEP (Wireless Encryption Protocol) Encryption

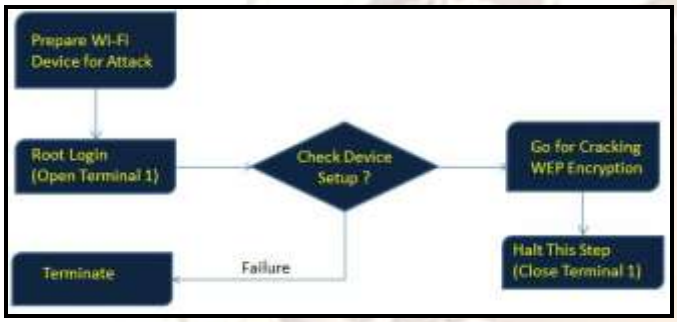
1. Log in a "*root*" user.
2. Discover a target network to attack upon with *airodump-ng.*
3. Terminate the discovery once enough nereby n/w are traced
4. Capture the packets and ARPs on wireless network with *airodump-ng*
5. If (speed of capture is SLOW)
   Use "airoplay-ng -3 –b c" command.
6. If (Error = "Waiting for becon frame")
   Repeat from step # 4 and step # 5
7. Save the captured in a target folder
8. Decrypt the data using *aircrack*
9. Halt

Step III) Cracking WPA (Wi-Fi Protected Access) Encryption

1. Log in a "*root*" user.
2. Open a new terminal to check the *Handshake*
3. Capture the handshakes using *airoplay*
4. Disconnect all the clients in the network by *airoplay*
5. Generate a "*wordlists/wpa.txt*"
6. Brute force the password from *wordlists/wpa.txt*
7. If (COMPUTER DECRYPTION SUCCESSFUL)
   a. Required KEY is achieved
   b. Close the terminal
8. If (COMPUTER DECRYPTION SUCCESSFUL)
   Repeat from step # 2 to step # 7
9. Halt

Procedure for hacking Encrypted WAN

1. **Preparing the Wi-Fi device for an attack**



Open the terminal and log in through the root user. Check status of your Wi-Fi device by typing `iwconfig` command. We get list of all the Wi-Fi devices installed.

```
airmon-ng stop wlan0
```

This command disables the monitor mode of Wi-Fi device.

We must not use our real physical address, hence use

```
macchanger –mac <proxy mac address>
```

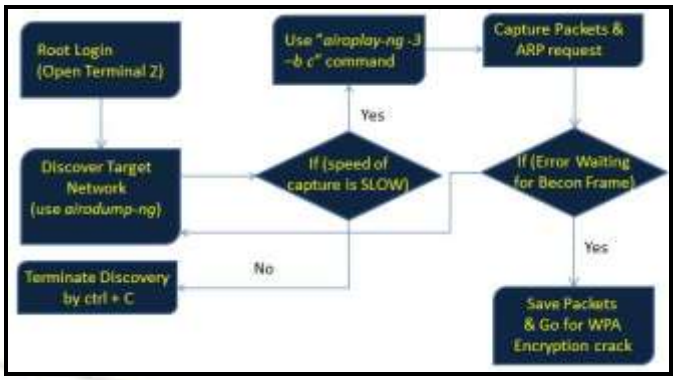
Now use monitor enabling to have full control over device you want to configure by using

```
airmon-ng start wlan0
```

2. **Cracking WEP Encryption**

First of all we need to select the target network to attack. Use "`airodump-ng start wlan0`" This will show all the active station nereby. It also shows their mac addresses, total users, traffics etc etc. Now hit "Ctrl+C" to stop the discovery. Now, it's the time to capture some packets. Use following command:

```
airodump-ng X c w <address where
captured packets should be saved>
bssid <networkaddress> <devicename>
```



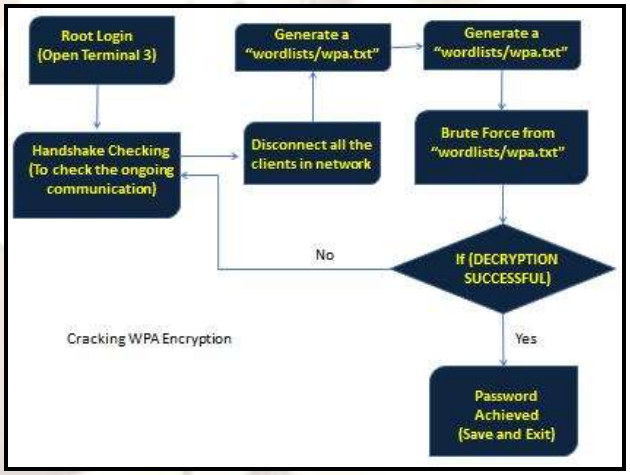
Here X is the network number enlisted in the output of the previous command. This will now capture all the data pakets.

Now open an new terminal and log in through root. Use command

```
airplay-g -3 –b <wirless bssid> -h
<myFakeMacAddress> <device name>
```

It is recommende to capture at least of 2000 packets.

3. **Cracking WEP Encryption**



Lets try to decrypt the captured date to find out the wireless network password. In the other terminal press ls to enlist the data folders. Use `wepcaptured 01.cap`.

```
aircrack-ng –bssid <address of
target> <path where captured data
has been stored> wepcaptured 01.cap.
```

The computer now try to decrypt the data and if successful will show the password. Now we have actually gained the access over the network.Furthermore we can manipulate as we wish. For example I want to disconnect all the clients connected, I use:

```
airplay-ng -0 15 –a <your network
device ID> <device name>
```

We come to know how actual handshaking is done on the other terminal and all the clients are being disconnected.

III. CONCLUSION

The method of testing the system reliability by trying to damage is not new. Whether an automobile company is testing cars by crashing the cars in a controlled environment, or an individual is testing his skills at an army training camp by sparring with another people, is generally accepted as reasonable. Identification of vulnerabilities is useless without regular auditing, persistent intrusion detection, high-quality system administration practice, and computer security knowledge. A simple breakdown can expose an organization to cyber-attacks, loss of income or mind share, or even something worse. Every new technology has its advantages and its risks. Ethical hackers can only assist their clients in better understanding and identifying of their security needs, it is the responsibility of the clients who decide whether to address them or not. Students also enjoyed the new vocabulary that has developed in the Google hacking community.

Using the results of penetration testing requires proper interpretation. Neither testers nor sponsors should assert that the penetration test has found all possible flaws, or that the failure to find flaws means that the system is secure. All types of testing can show only the presence of flaws and never the absence of them. The best that testers can say is that the specific flaws they looked for and failed to find aren't present; this can give some idea of the overall security of the system's design and implementation. Penetration testing is effective because it lets us consider a system as it's actually used, rather than as it's expected to be used. It's something to which all computer security students should be exposed.

REFERENCES

1) Packet Sniffing:A Brief Introduction DECEMBER 2002/JANUARY 2003 0278-6648/02/$17.00 © 2002 IEEE
2) PERVASIVE computing Published by the IEEE CS n 1536-1268/08/$25.00 © 2008 IEEE
3) Teaching Students to Hack: Ethical Implications in Teaching Students to hack at university level
4) "What Hackers Learn that the Rest of Us Don't", THE IEEE COMPUTER SOCIETY, 1540-7993/07/$25.00 © 2007 IEEE.
5) Ethical Hacking: The Security Justification Redux by Bryan Smith William Yurcik David Doss Illinois State University, 0-7803-7824-0/02/%10.00 62002 IEEE.
6) Network Viruses: Their Working Principles and Marriages with Hacking Programs by Yanjun Zuo and Brajendra Panda Computer Science & Computer Engineering Department University of Arkansas, ISBN 0-7803-7808-3/03/$17.00 0 2003 IEEE.
7) Hacking Exposed in Linux 3rd Edition by ISECOMM Tata McGrawHill Publication
8) Linux 101 Hacks: Practical Foundation to built strong foundation in Linux by Ramesh Natarajan.
9) Beaver, Kevin, CISSP (2003), Ethical Hacking: Ten crucial lesson. Retrieved on June 21, 2006.
10) Ethical Hacking and Cybercrime by Nataliya B. Sukhai 6675 Williamson Drive Atlanta, Georgia 30328
11) Legal Policy and Digital Rights Management by RICHARD OWENS AND RAJEN AKALU, PROCEEDINGS OF THE IEEE, VOL. 92, NO. 6, JUNE 2004.
12) Hacking Risk Analysis of Web Trojan in Electric Power System by Yong Wang , Dawu Gu, Dept. of computer Science and Engineering, Shanghai Jiao Tong University Shanghai, China
13) Hacking Competitions and Their Untapped Potential for Security Education by Gregory Conti, Thomas Babb itt, and John Nelson, US Military Academy, COPUBLISHED BY THE IEEE COMPUTER AND RELIABILITY SOCIETIES
14) Be Protected and Feel Secure – Ethical Hacking Seminar by KARMA Cyber Intel, Mumbai.