

Information Security: Level #4
Prof. V. Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 08
History of Kali Linux

In this particular session, we will talk about the history of Kali Linux. So, as I mentioned in the beginning of this course, that we are going to teach you cyber forensic, some incident has happened; how do we trace this? So, as a part of this we also want to prevent these incidents happening. So, for that each of the industry that is having an IT based infrastructure, information technology-based infrastructure we do something called penetration testing, it will try to test the entire system to see if it is vulnerable to attacks. So, vulnerability and penetration testing VAPT, is something generally done by all the T organizations, right.

So, and when we do that, basically we can find out all the vulnerabilities and loopholes and plug those loopholes. So, that attack based on those loopholes, will never occur that is one way of looking at this, another thing another ways that that then attack has happened, what are the type of logs that we maintain? So, that we could pinpoint and find out who has done it. So, these 2 are very, very important and now Kali Linux is something which you can use, for your cyber forensics and it gives, it comes with a package of tools, by which you can basically analyze the security of your network and if you find a loophole, you can basically you know rather than saying that this is a loophole.

You can basically exploit that loophole and do an attack and then ethically go and say that, yeah this is the problem and this is how somebody can attack, then your organization will certainly take it in their head and basically plug this hole. So, that is very, very important in today's context.

(Refer Slide Time: 02:05)

UNIX to Kali Timeline

- ▶ 1968: E.W Dijkstra develops MULTICS (Multiplexed Information and Computing Service) in the Netherlands
- ▶ 1969: Bell telephone (AT&T) lab researcher Ken Thompson developed a new system using MULTICS as part of a team. His coworker Brian Kernighan dubbed it UNICS (UNiplexed Information and Computing Service). It was later changed to UNIX.
- ▶ Milestone: The UNIX operating system was born.



Now, what we will do is that we will quickly look at this, Kali Linux what is Kali Linux? It is based on the name of a Hindu God Kali right? So, what is Kali Linux after all does it. So, let me just quickly go through the history of the transformation from UNIX to Kali, whatever what did happen? 1968 Dijkstra actually develop this multics operating system, in the Netherlands 1969 bell AT and T bell labs, Ken Thompson was the name of the researcher in AT and T bell labs developed a new system using multics, as part of a team.

So, his co-worker Brian Kernighan dubbed test UNIX, that is uniplexed information and computing service, it was later changed to the UNIX. So, one of the mile stone that happened between 1968 to 70, 70 early 70 was the UNIX system operating system was indeed one, not just one as I ate another operating system, but it had specific ways of addressing it has specific way of addressing some very, very important issues right it was just not an you know another operating system ah, but it was specific to addressing some of the vulnerability and penetration testing, and the most important thing that was that this UNIX was born in this range 1968 to 70; early 70. 1969 to 1973, bell telephone researchers Dennis Richie Cunningham Richie develop the C language as a systems programming language for UNIX, right?

(Refer Slide Time: 03:33)

UNIX to Kali Timeline

- ▶ 1969-1973: Bell Telephone researcher Dennis Richie develops the C language as a systems programming language for UNIX.
- ▶ 1970s: UNIX versions 6 and 7 were developed, first in B and Assembly than C. Originally for academic use, later sold to vendors.
- ▶ 1987: A Unix-like system based on microkernel design known as MINIX was developed.

- ▶ Milestone: C language developed.



So, this also became very crucial right if this was not that then they are probably your UNIX would have been written in scientific computing languages in 1970s, UNIX version 6 and 7 were developed first in B and assembly then C and originally for academic use later it was sold to vendors. So, the UNIX version 6 and 7 was first in being since origin binary and assembly rather than in C in 1987, a UNIX like system based on microkernel design known as MINIX was developed. So, at the end this period between 1969 to 87 in the UNIX to Kali timeline, the milestone there was that the C language got developed.

(Refer Slide Time: 04:27)

UNIX to Kali Timeline

- ▶ 1980s-1990s: The “UNIX Wars” occur, vendors struggle to standardize UNIX.
- ▶ 1991: Linus Torvalds developed a new operating system called Linux, which is similar to MINIX.
- ▶ 1990s-Today: Various UNIX and UNIX/Linux-like distributions are released, such as: GNU, OS X, Debian, and Ubuntu.

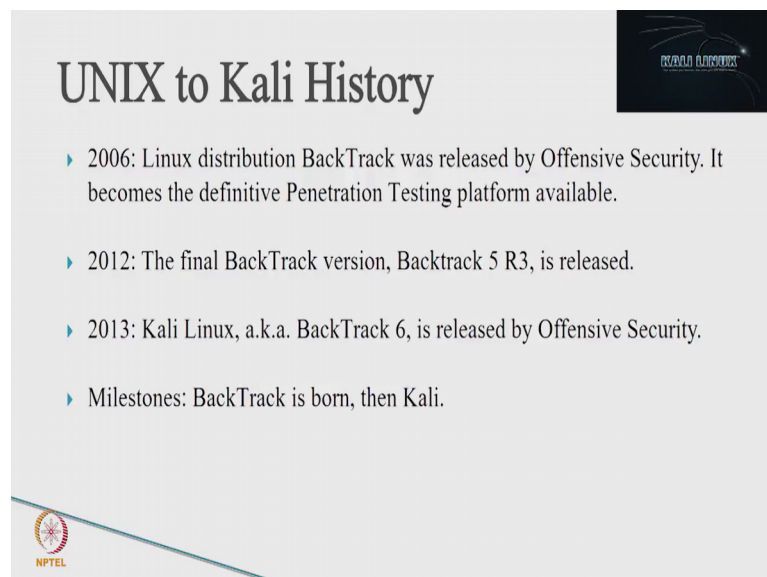
- ▶ Milestone: Linux was born.



In 1980 to 90, the UNIX was continued occur vendors, struggled to standardize UNIX. So, Linus Torvalds actually, developed a new operating system called Linux which is similar to the MINIX. And 1990 to today the various nix and UNIX slash Linux like distributions are released such as GNU, OS X, Debian and Ubuntu. So, the 1980 to 19 till to date one of the major achievements or milestone was Linux was born and this helped lot more of you know careful diagnostics that could be done on the system. So, just to go back from 1968 and 69 there was a UNIX operating system, that was standardized between 1969 to 73 C language as a as a systems programming language for UNIX was formed.

1970s UNIX version 6 and 7 not in C, but in some other assembly language and binary was actually developed. In 1987 a UNIX like system based on microkernel design known as MINIX was developed. The entire in this period C language development was a major milestone and between 1980 to today, people have actually started moving towards Linux and Linux was actually born in this framework and people have started moving towards Linux.

(Refer Slide Time: 05:49)



The slide features a title 'UNIX to Kali History' in a serif font. To the right of the title is a small dark square with a network diagram and the text 'REWARD HACKERS'. Below the title is a bulleted list of milestones. In the bottom left corner, there is a circular logo with a gear and a star, labeled 'NPTEL'.

UNIX to Kali History

- ▶ 2006: Linux distribution BackTrack was released by Offensive Security. It becomes the definitive Penetration Testing platform available.
- ▶ 2012: The final BackTrack version, Backtrack 5 R3, is released.
- ▶ 2013: Kali Linux, a.k.a. BackTrack 6, is released by Offensive Security.
- ▶ Milestones: BackTrack is born, then Kali.

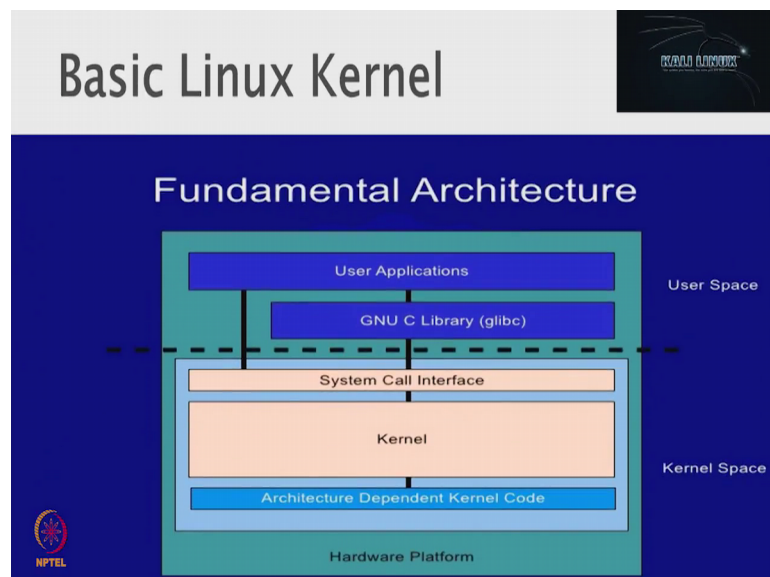
So, in 2006 a Linux distribution called BackTrack was released, by offensive security, it becomes the definitive penetration testing platform. So, I wanted to test the operating system for vulnerability, I put Kali Linux as a testing utility which basically can review code and give you potentially you can basically execute this test and potentially give you

how safe is your operating system? In 2012 the final BackTrack version BackTrack 5 released 3 was released. In 2013 Kali Linux also known as BackTrack 6 as is was released by offensive security. And so, the milestone here, is again, then BackTrack was born and that was followed by Kali.

(Refer Slide Time: 06:34)



(Refer Slide Time: 06:41)

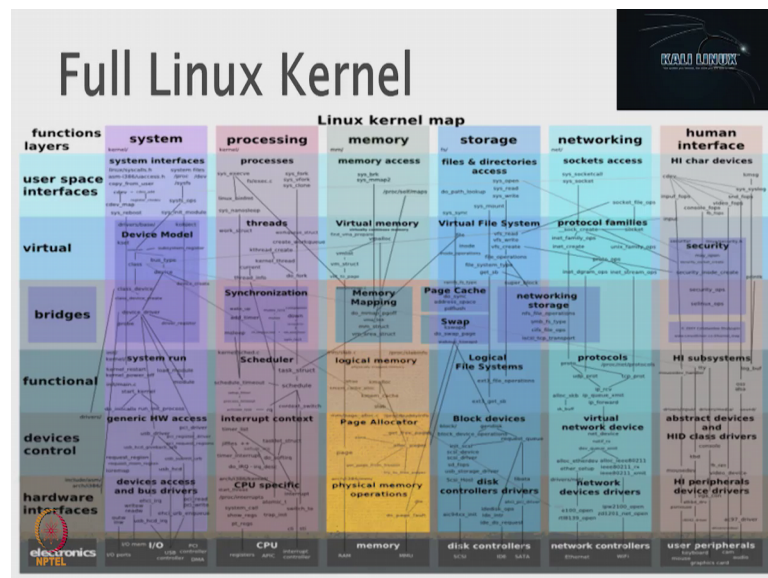


So, we will just go into what is this Linux OS architecture and see how Kali actually fits into this. The basic Linux architecture is as follows there is a user application at the top layer, then there is a GNU C library and all these things the user application can directly

talk the operating system, well that GNU C library can talk only to the system call to t not about that. So, from the UNIX from the user applications, basically you come to an environment where you call system called directly go to the kernel as seen on the left-hand side, that black line or from a user application you go through a GNU C library and then go to some system call interface and then execute on the kernel.

So, this is this is the basic call Linux kernel, as you see here. But please note that, there are applications that will bypass the GNU C library and start executing through the system call interface, not every request has to come to the GNU C library interface as seen. So, the users space as multiple way of accessing the kernals, well the kernel space also has opened multiple ways by which and that users space can basically access them.

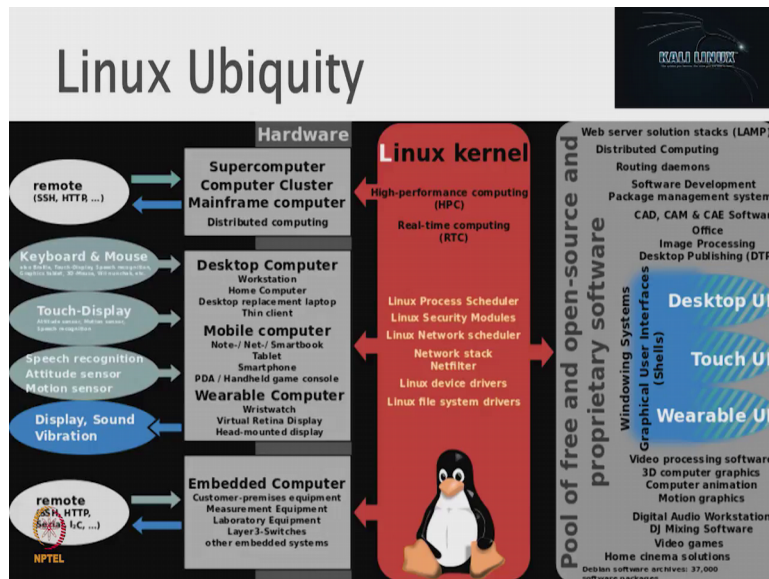
(Refer Slide Time: 07:48)



So, this is a full Linux kernel, I am just showing this slide just to tell you this is a big thing. So, you have modules for IO, you are modules for CPU, your modules for memory, disk controllers, network controllers, user peripherals and many, many more.

So, at least to start with we have 7 different kernel issues, the kernel modules which we need to basically address, when we want to start off talking of a secure operating system and put it into practice.

(Refer Slide Time: 08:25)



And the Linux has been a very, very common platform, leave alone from say some supercomputer to embedded computer, supercomputer, computer cluster, mainframe computer, distributed computer, desktop computer, mobile computer, wearable computer. So, for all these things and embedded computers for all these things there is a version of UNIX which will be it. And that is why, the next itself calls it as a you know a very, very took it as environment, because right from high performance computing to embedded computing the Linux can basically take care of this version of Linux that supports it.

(Refer Slide Time: 09:09)



So, we will get to Kali already. So, we will start looking at Kali. So, before we go to the next section, what I basically want to tell you is that, monitoring the operating system,

monitoring the network are 2 ends of the game and all network monitoring basically, underlying, you know assumes some amount of sanctity from the underlying operating system, right?

So, suppose I want to do a network level complete scan and see, the operating system should cooperate in the sense that operating system should basically give you a very secure execution environment etc, unless we have that, we really do not know whether the problem is it the cash sorry the problem is with the you know network or if the operating system. How the problem is with the people process, etcetera, with this this is very, very important. So, that is why we are I have spent some time on network security, now I have spent some time on operating system level security and we will take it forward from here and look at more on the operating system.