

Information Security - II
Prof. V. Kamakoti
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 07
Architectural Aid to Secure Systems Engineering
Session – 6: Security and Architecture

Now, we really go in to the main business of this course Security and Architecture. Now, I strongly believe that every course at least in the systems side from computer organization, computer architecture, operating system, compilers, all should be thought from a secure and networking hardware at least apart should be thought from a security point of view, then only it makes sense for the next era.

All the other stories are already done, I think the way we should form the next generation syllabus should be that all the other stories we should learn from all video lectures which we host in the NPTEL and other things. The real course that we teach in a curriculum should be more focused toward security, and that makes at least at a post graduate level so that you know we address one of the most important problem. We need several lakhs of engineers to handle these issues and we produce in 1000's, so there is a big gap and there is a lot of market value also for this. I think, it is only of reason that people even talk at even a big management decisions in a bigger organization especially, financial organizations. The need for information security is slowly coming in they are realizing the need first information security. And so, within next 4 to 5 years there will be a huge demand so there will be you know presentation to top managements where they start asking you about intricate questions about security, right, intricate questions about hardware's.

So, there will be force to learn all these things the moment you know the systems take same hard punch you know, in terms of security vulnerabilities. If you start feeling the pinch then you start reacting that will start happening, and that is my opinion. So, this course becomes extremely important from that point view. Again, as I told you in the previous session that there is no connection between you know, what was the root cause

and what was the actual action that effect at you see. The caution effect relation is really far spread.

Similarly, here also I will be teaching lot of computer organization, I will bring in some security point of view, but for you to realized where it is going to really useful, you have to go through all the 4 courses and then finally, you realize where you can use it in full perspective. This deep dive that we have going to do it architecture today will be finally appreciated, when you start building or writing some secure solutions or start building secure engineering. At least for you to appreciate, why certain problem has occurred? Right? That itself unique, lot more intelligent and there also you need to diagnoses at a security vulnerability, you need to have this much amount of architectural understanding. So, with this I will start here.

(Refer Slide Time: 03:10)



The slide is titled "Objectives of this course" and is framed by two vertical black bars on the left and right sides. The text is centered and includes a bulleted list of objectives.

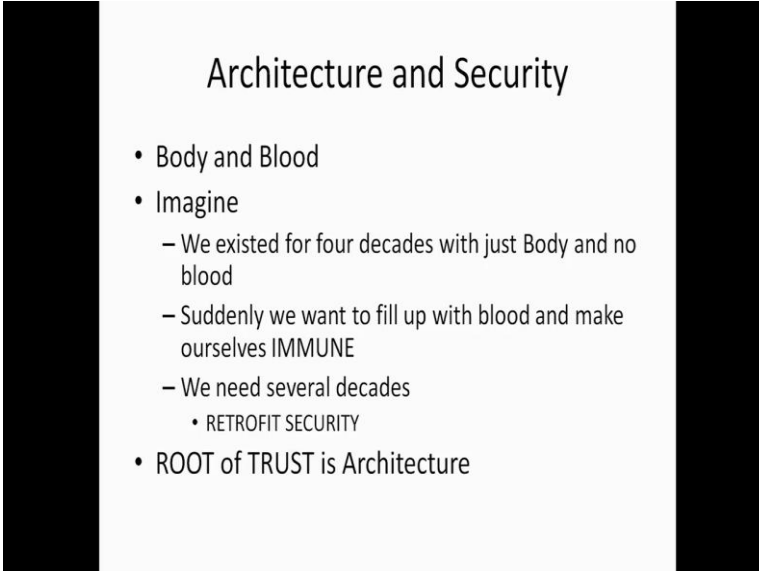
Objectives of this course

- A solution to these issues
- A collective effort of Architecture, OS, Compiler and Application Developers
- Discuss Architecture structure, roles and responsibilities
- Rest in subsequent courses
 - We have four more in the Information Security series.

All the issues that I have talked, we need to find a solution there is no going back on this and that solution will be certainly a collective effort of architecture, operating system, compiler and application developers. So, what we are going to do in this course is we are going to discuss, what is the role of the architecture? What should be the structure in that architecture? What are the roles and responsibilities of the architecture in this collective

effort? The subsequent courses we will see, what is it that the operative system should do? What is this the compiler should do? And what should an application developer should do? We will have 4 more course in that series to complete this.

(Refer Slide Time: 03:52)



The slide is titled "Architecture and Security" and contains the following bulleted list:

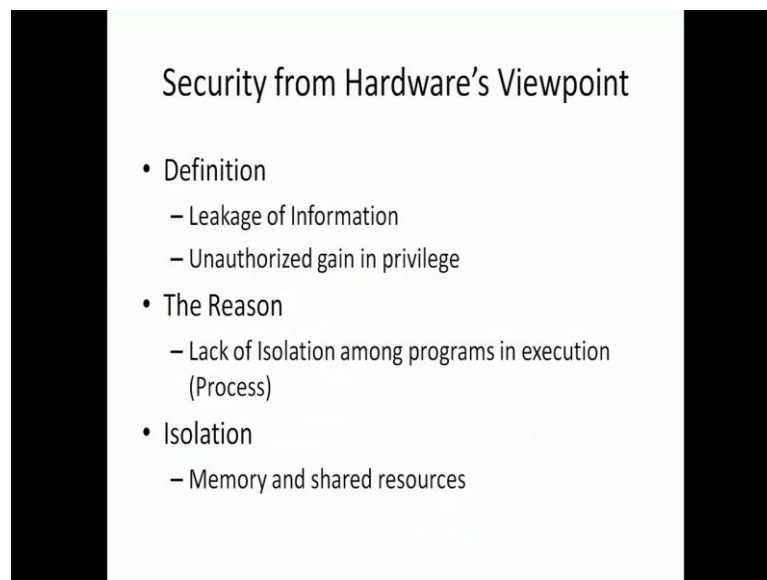
- Body and Blood
- Imagine
 - We existed for four decades with just Body and no blood
 - Suddenly we want to fill up with blood and make ourselves IMMUNE
 - We need several decades
 - RETROFIT SECURITY
- ROOT of TRUST is Architecture

Now, this architecture is like the body, security is like the blood. So, let us take this imaginary situation that you see on the slide, suppose we existed for 4 decades with just body and no blood and suddenly, we want to fill up with blood and make ourselves immune, this we need several decades. That is what I have been telling next 3 decades do not worry we will have security related issues, and this is because we are now retrofitting security. When you see the diagram which I have shown quite now, a very nicely I think you will all memorize, I will keep on repeating certain things, I believe that if you repeat 7 times it will at least 1 time it will sit in your head. I will do that and so it is very important. So, what you have is first the hardware on top of it is your micro architecture and top if it operating system, etcetera.

All the other things are excellent no problem there, but then you have this hardware, there are very flimsy lousy hardware you are gone, all your excellence goes off because another finally, this is the fellow who is executing your programs and if there is some

stupidity here where anybody can come inside and do something, all however good or however best your operating system and compiler and at a layers be, if your hardware is flimsy everything is gone. So, hardware is the root of trust. The root of trust is your architecture and that we have to be extremely you know, solid so that you build a secure system. So, the foundation of your secure system is in hardware. That is why if you start working on secure hardware, already if you work on security will be boss and if you work on secure hardware you will be really a boss. Let us go in to this, what is security from a hardware's view point? Like what is lack of security? Here, it is leakage of information.

(Refer Slide Time: 05:57).



The slide is titled "Security from Hardware's Viewpoint" and contains the following content:

- Definition
 - Leakage of Information
 - Unauthorized gain in privilege
- The Reason
 - Lack of Isolation among programs in execution (Process)
- Isolation
 - Memory and shared resources

I have some private information I leak that is a breach of security. Other thing is there an unauthorized gain in privilege. Now you are user you get into super user, after becoming super user you will do lot of things. So, the 2 definitions of security today is I should, what is a secure hardware? Hardware, which will prevent leakage of information, because I could leak I could gain information even before getting a privilege, I could be an external fellow. So, I should have a clear define mechanism which will prevent me from leakage of information. And there should be a clear mechanism which will ensure that there is no unauthorized gain in privilege. If I am going to ensure these 2 things, I

think we have almost a good secure hardware. All that my responsibility to the operating system to the compiler that are going to execute on top of me is that, I see that there is no leakage of information, I also see that there is no unauthorized gain in privilege.

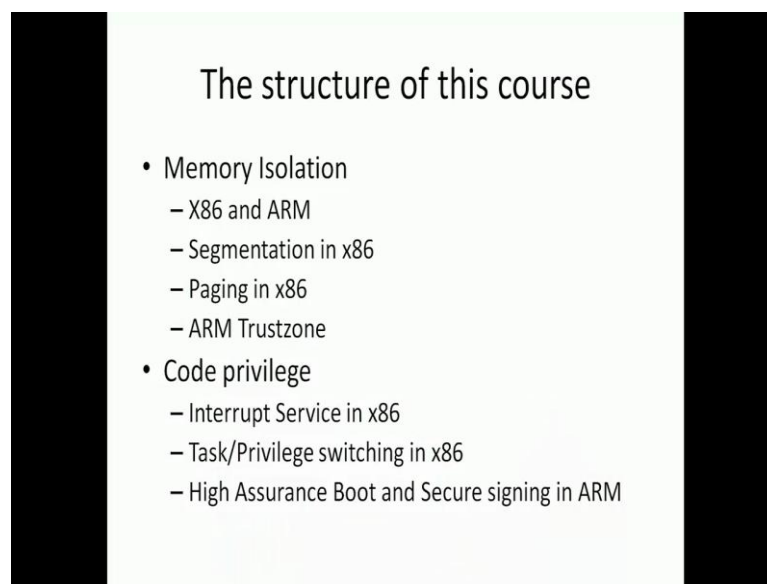
See, how can information leak? How can I get unauthorized gain in privilege? The main reason is the processes that are execution, what is a process? A process is a programming execution. The programs that are executed, they do not have what you call as perfect isolation. What do you mean by isolation? I share certain resources; for example, why did stack smashing happened? Stack smashing happened because I shared the same stack. The calling function and the called functions shared the same stack, they were not isolated because they were still sharing the same stack. If I take any 2 processes in the system, what will happen? I share the same registers. In an architecture, there are general purpose registers, right? When I am executing I am using the registers, after I finish or suppose I have a scheduling algorithm where 10 programs are scheduled on this same 10 processes are scheduled on the same hardware. I execute now, I use the general purpose registers, I move out somebody else comes in, he also uses the same hardware registers. When I am moving out if I did not erase those registers, the other fellow who comes in can keep reading this registers. And by reading this registers the information about me can go to the program who is going to follow.

Why I cannot erase all the registers? I cannot erase all the registers because I do not know when I will be moving out. I am a program who has to execute for 100 seconds, I will be given 100 milliseconds some 100 times or 100 milliseconds 1000 times, if I have a round robin scheduling. What is the round robin scheduling? 100 process is sharing 1 fellow each will be given sometime pulled out next fellow will be put in, pulled out it is at robin hood type of thing. So, I execute now, I do not know when I am going to be pulled to out, I will be pulled out by whom? Some interrupt will come, and pull me out. The operating system then loads a next fellow. Now, who is responsibilities to erase the registers operating systems responsibility, but if it does not do then what will happen, then the next fellow who comes in can access these registers and get back. You are getting my point? So, we cannot ensure perfect 100 percent isolation between processes, as of today in the contemporary operating systems. And since, I cannot ensure that there

is a sharing of resources and that basically results in leakage of information. Are you getting this?

The main reason is memory because memory is one source, memory in different forms it can be cash, it can be you know registers, it can be ram main memory. It can be disk, but memory in any form is the one source, where one violate the property of isolating individual process is that need not communicate. And that is one of the major reasons for a leakage of information. There is also a major reason for unauthorized gain in privilege. There also the calling function and the called function shared memory stack and that was the reason for that is it.

(Refer Slide Time: 10:56)



Now, let us go. What are we going to do here? We are taking about Memory Isolation and we are going to take about to code privilege. Privilege in the sense that I can do certain things I cannot do certain things, if I have some privilege I can do certain things, if I do not have that privilege I cannot do certain things. So, how is that implemented? What is it that the architecture can do to get you that code privilege? What is it that the architecture can do to get you that memory isolation? These 2 aspects in my opinion if you are thorough if you have an understanding I thing you can start appreciating security

information security in a better perspective. So, what we are doing to do in the area of Memory Isolation is we are going to look at 2 very, very interesting architecture the x86 you just clean seen in every laptop, every many 99 percent of your laptop and desktop systems do you use x86. Arm with 99 percent of the computers today a mobile phones has ARM process.

We will look at Memory Isolation from the x86 perspective and the ARM perspective I think. And then, what is it that we are going to see in x86? What is interesting about x86? Why we are all excited about x86? Is that we have seen 5 generations, it was exciting in many of your previous birth, even you have before you have born x86 came in. And probably if it exists for another some more time it can see 2, 3 generations of the same soul coming back to a fore from the earth. It is a very interesting thing, so if anybody wants to understand how architecture as evolved you have to go and look at you know history of Intel and AMD its very, very important fascinating things. We will deal about 2 important memory management concepts in x86 architecture which is the segmentation and virtual memory paging. I will explain those details those things in very much detail. And from ARM we have something called the Trust Zone and we will have an ARM Trust Zone. We will have a demo of a product that we are made in IIT, Madras and we will show you some of the Trust Zone features there. This is what we will talk about in the Memory Isolation in this course. And then in the Code privilege section we will talk about Interrupt Service.

Interrupt Service is one very interesting and most important thing, so whenever we talk of process isolation, interrupts also come in to your place. For example, in a Round Robin scheduling one process is pulled out and another process is put it and this pulling out is done by an interrupt service routine correct, there is a timer interrupt which is stop the execution of the current program and that interrupt service routine is responsible for putting something in. So, I can be a least privilege program and there can be an interrupt right which you will be a higher privilege program. There is a privilege transition when a program is executing and something happens there and then an interrupt service routine is executing after that. So, there is a shift of privilege limits there and that is why whenever we study anything interrupt is very, very important. Any certification we go to prove a system is secure interrupt plays a very, very important role.

Another important thing about interrupt, is that it is very asynchronous we do not know when the interrupt will come as I explained you in that when I move out another code comes in why cannot they erase the registers and go I cannot do that because I do not know when I will be pulled out. The interrupts are also asynchronous events, so that make interrupt service a very, very important thing. Whenever we do interrupt service there are very settle things in the Intel architecture, in the AMD architecture, in the x86 architecture general which will ensure you these type of security and that needs quite a bit of deep dive in to the architecture no normal text books or conventional computer organization course does not teach you that.

Of course, switching off task so I am a privilege 0 high privilege code then I start executing the low privilege code, what it means, how can I go from a high privilege to your lower privilege or lower privilege to high privilege. So, what are the basic safe guards I need to take before I do that such type of switching. And then we will also talk about 2 interesting technologies that I have come up, 1 is the High Assurance Boot and secure signing in Random Access Memory we will do it in very great detail at the end of this course which will stop you from concepts like BMBR. The virtual machine based suit case how can I handle that is to this.

(Refer Slide Time: 16:02).

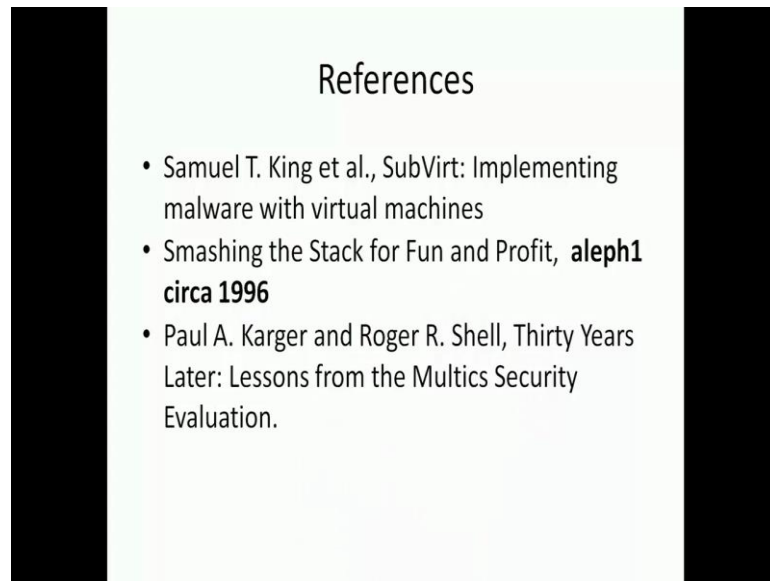
Practical Aspects

- Security is no more theory – it is full of practice
- In this course
 - Develop a Mini Kernel for x86 environment in assembly that help explore the security features provided by x86 hardware.
 - Live demonstration of certain hardware security features in ARM

So, when we study computer organization here, it is not just going to be, we are not just going to talk about you know hardware. We are going to talk about lot more of operating systems and tell you what is it that is need in the hardware and how its is existing so that it handles (Refer Time: 16:22). This is an issue based you know, course rather than trying to teach what the hardware is, right? So very importantly, please at understand security is no more theory it is full of practice. This course we will try, we are going to give you lot of programming exercises and we do live demo of these exercises. At the end, we are going to develop a small Mini Kernel for x86 environment and that Mini Kernel will be return in an assembly. That we are going to do assembly level programming and that will help explore the security features provided by x86 hardware. The best way to understand hardware security is to write assembly programs and realize this. Unless you go and work at the assembly level you cannot have an intricate understanding of the hardware.

If you write C Programs, then there is somebody intermediate between this C Program and this you cannot even understand how those things, you are one layer above. You have to be very close to the hardware, close to heart of that hardware and then find out. So, we will be giving you assembly program based exercises in the environment that would have created in your laptop and we will do that. There will be a live demonstration of certain hardware security features in ARM, so that is what we will do. And this course essentially is sufficient enough in my opinion to take you to the operating system and appreciate operating system level security. So, when we start talking about the operating system in subsequent courses and we say that, OK this is how we should realize. Now, you will have immediately mapping on, how we can realize it on conventional contemporary hardware's like your x86 and ARM right, so you will have that mapping. Some (Refer Time: 18:08) security concept, how do you realize it? What will be the hardware support? You know how the hardware can support you, and that is would be the contribution of this course.

(Refer Slide Time: 18:19)



So, these are the references Samuel T. King et al., SubVirt: Implementing malware with virtual machines. Smashing the Stack for Fun and Profit, just search will get this. And this is the Paul A. Karger and Roger R. Shells paper, 30 years later: Lessons from the Multics Security Evaluation; these 3 are very important things which will stand as a motivation for you to undergo this course.

So, any doubt in this session I will now take up, else I will go to the next. Yes, doubt.

Exactly, so the question is that, if I have at a higher level of abstraction like your operating systems or compilers at application level, if there is the breach of security can the hardware catch you? The hardware is supposed to catch. Why are you asking this question, so the answer I should give you is that how is that related to this course, correct? So, for the hardware to catch this is not the hardware that is catching the hardware will give you some features, now we should use those features intelligently so that you catch it. Hardware will only build your dam, now you have to fill it with good water. Now, for you to catch those vulnerability you have an infra structure to catch that vulnerability. So, we have put a road and put a gate, now it is for you to how to lock and

open that gate so that trespassers do not enter, you followed? It is for you to put that watchman; we are not putting the watchman for you.

One thing is to understand, what the hardware can do for you; the second thing is, how you exploit that feature to basically get those things done. For example, there will be notion of a Mini Kernel there is something called a G-Node. A project that is we are also would do at IIT, Madras and some of funding partners are also involved in that which is a Micro Kernel base approach. What it does is, it gives you a very good isolation between a different processes and the hardware. The G-Node is called a Separation Kernel in all all practical aspects. And what will this Separation Kernel ensure you? It will enumerate a list of objects and it will also give you a way by which you can impose rules on this access to these objects. Vulnerability, basically comes when I am having unauthorized access to an object, you are getting this? And when I got, get an unauthorized object access to an object. Basically, I am violating some access rules.

Now, this Micro Kernel will give you a way by which you can enforce such rules so that you do not get an unauthorized access to an object, by that I could if somebody tries to do it; yes, you can catch it. In the course itself, you will be doing some assignments where you try to go and access a memory which you are not suppose to access and you will be caught. So, there is simple thing I can tell you that will happen in this course, in the assignment that you are going to do, the programming assignment you are going to do, that you are going to demonstrate it, correct. But then, beyond that we give you the infra structure and it is up to your intelligence to build it up and use it effectively.

Any other doubts? No, but you do not know whether he is an attacker or not an attacker, right? If I know that you are an attacker, I stop you there itself I will not even allow you beyond the ethernet card. I do not know whether you are an attacker or not an attacker and that is the problem, I do not know you are genuine or not genuine because you come with credentials, correct? Followed what I am saying? For example, even in the case of stack smashing which will be using quite throughout, I do not know what sort of code you wrote, I do not have any way by which I whether you are writing normal data or writing an assembly program, but you actually wrote an assembly program and got access through that assembly program.