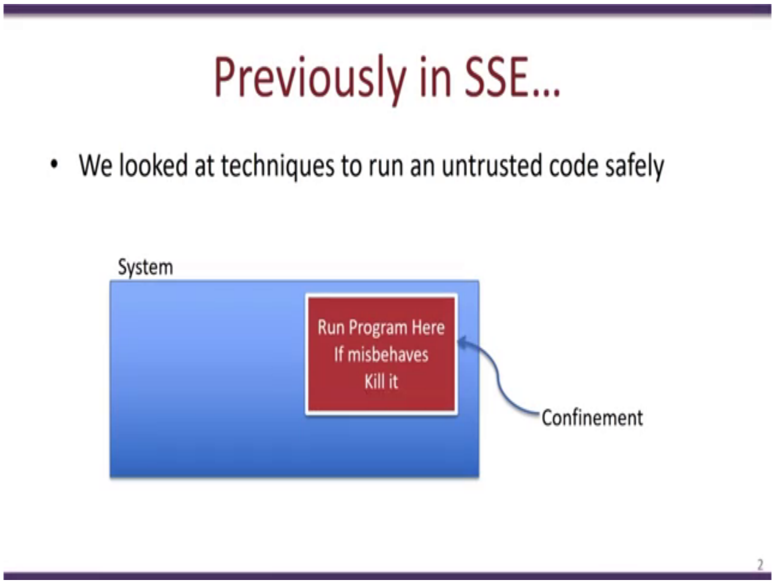
Information Security - 5 - Secure Systems Engineering
Professor Chester Rebeiro
Indian Institute of Technology, Madras

Trusted Execution Environments

Hello and welcome to this lecture in a course for secure systems engineering. So in the previous couple of lectures we had looked at a problem of confinement. In this lecture another lectures that follow we will take a new topic know as trusted execution environments.

(Refer Slide Time: 00:35)



Previously in SSE...

- We looked at techniques to run an untrusted code safely

System

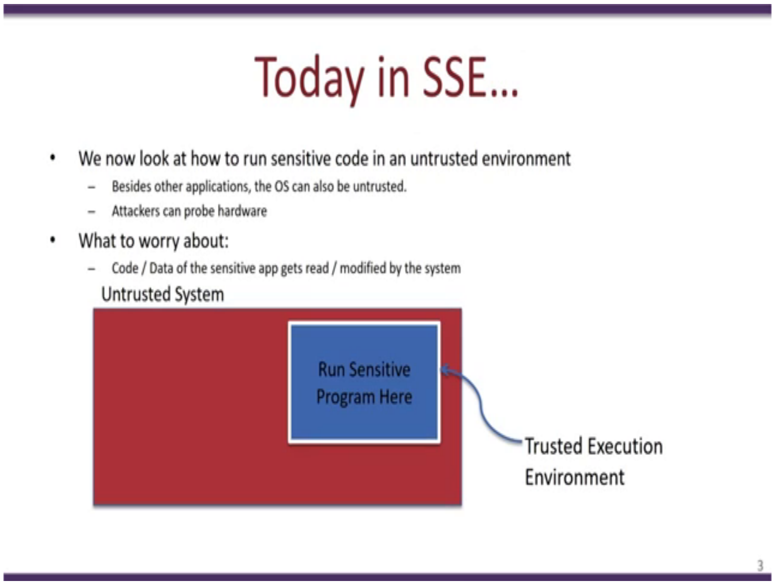
Run Program Here
If misbehaves
Kill it

Confinement

We first start of it is in how this particular trusted execution environments is different from confinement. So with confinement we had looked at something like this, we had a system over here and within this particular system we wanted to run a particular program and this program may be malicious. So what we needed to ensure was that we needed to ensure that this particular program is confined to a specific area. So we will start off with how trusted execution environment is different from confinement or the topics that we have studied in the previous lectures.

With the confinement we had considered a system like this, we had considered that we have system and in the system we wanted to run a program which may be malicious. So what we did through multiple techniques such as lease privileges or the OKWS or the software fault isolation we had created confined areas which executed the possibly malicious programs. So what the confined areas achieved was that any misbehavior is always restricted to this confinement. Now this is very different from trusted Execution Environment where we actually look at the inverse of this.

(Refer Slide Time: 02:06)



Today in SSE…

- We now look at how to run sensitive code in an untrusted environment
  - Besides other applications, the OS can also be untrusted.
  - Attackers can probe hardware
- What to worry about:
  - Code / Data of the sensitive app gets read / modified by the system

Untrusted System

Run Sensitive Program Here

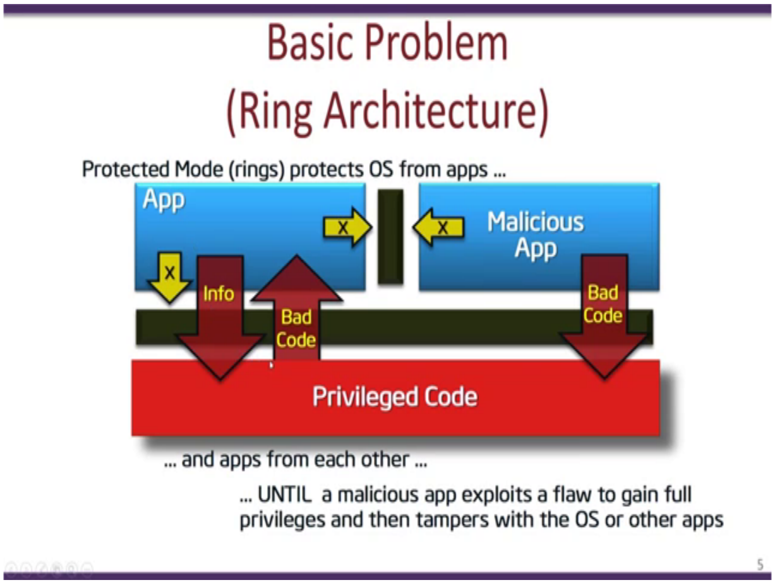
Trusted Execution Environment

3

With trusted Execution Environment we have a very sensitive program that we want to run. It is so sensitive that we do not even trust the system that it is running on. So in other words what we actually discuss in these lectures is that we have a very sensitive program and you could consider this sensitive program for example say a banking application or this program wants to do encryption of very critical data and we want to actually run this particular program in an environment which is untrusted, so for example you may have a malware or any other malicious applications running in your system which has compromise the entire system.

Now in spite of all of these malicious program running on your system we would still want to run our sensitive program without loss of any information. In spite of this untrusted system we would want to run our sensitive program in a safe and secure manner.

(Refer Slide Time: 03:17)



Basic Problem
(Ring Architecture)

Protected Mode (rings) protects OS from apps ...

... and apps from each other ...

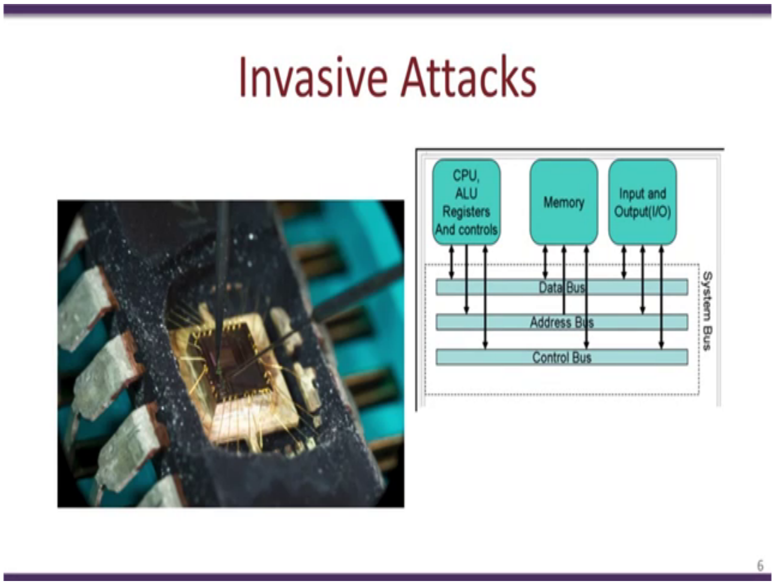
... UNTIL a malicious app exploits a flaw to gain full privileges and then tampers with the OS or other apps

The heart of the problem is the ring architecture that is adopted by all processors and operating systems. In the ring architecture for example X86 ring architecture, there are multiple rings, ring 0 to ring 3, the operating system runs in the ring 0 which is the privileged code and it creates an environment for various applications to run. So all the applications are running in ring 3.

Now the entire system is designed in such a way so that the privileged code or operating system is able to access the memory and data of all other applications. However, one application is isolated from another application that is one application cannot invoke or access data of another applications. These restrictions are enforced by the virtual memory and page tables setting while the partitions between the applications and privileged codes are achieved by the ring architecture. Now problem occurs when one of these applications becomes malicious and finds a bug in the operation system or the privileged code and has compromised the operating system. So this occurs quite often what we see is that for example for this program is malicious it has found a bug in the operating system and it has created and exploit and has compromise the entire operating system.

Now the problem over here is that since the operating system controls the entire system and can monitor or modify all applications present in that system. The result of the OS or the privileged code getting compromised is that all other applications present in that system will also get compromised, in other word the entire system is compromised. So this is the basic problem in today’s system and therefore solving this problem where you run a sensitive application in the system in spite of a compromised operating system is extremely difficult.

(Refer Slide Time: 05:46)



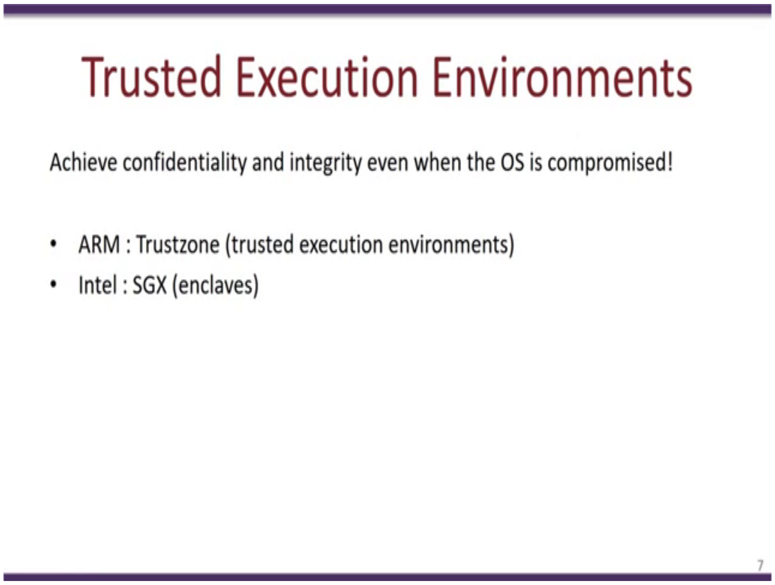
Invasive Attacks

Now another possible attack scenario is due to invasive attack, so what happened over here is that attackers may be able to de-package the processor IC and will be able to monitor internal signals within the IC. For example if you look at these particular figure where you have the CPU, memory and the various input output and all of them share the same data and address bus. What would happen in a very typical invasive attack is that the attacker would be able to monitor the address bus and the dada bus and thus gain access to may be sensitive information that is executing in the processor.

So another recent attack is the Cold Boot Attack, so this particular attack use the fact that the memory is made out D-RAM or the dynamic RAM and the fundamental component in the D-RAM is the capacitor. Now the capacitor holds charge and to indicate that a 1 is stored in that particular memory bit or a capacitor discharges when a 0 is present in that particular bit. Now the problem here is that people have found that even when the power is turned off the state of this capacitors or many of these capacitors is still detained for certain amount of time. So what researchers have been able to do is to actually pick a D-RAM chip from a system plug it into another system and then scan that particular D-RAM and read all the contents of that D-RAM.

Since a large portion of the D-RAM content is still available a lot of information present in the D-RAM can be retrieved by the attacker.

(Refer Slide Time: 07:37)



Trusted Execution Environments

Achieve confidentiality and integrity even when the OS is compromised!

- ARM : Trustzone (trusted execution environments)
- Intel : SGX (enclaves)

7

In the lectures that follow we will be looking at trusted execution environments which can handle these kind of attacks. So in a typical trusted execution environment we assume that the operating system itself is compromised and thus the entire system is (comp) may be compromised. In spite of this what we would what a trusted execution environment would provide is an environment where you can run sensitive applications in spite of being OS compromised. So trusted execution environments are becoming quite common in the recent years and finding various applications in a multiple domains ranging from embedded system to high performing computing.

Both Intel and AMD have actually created trusted execution environments in for the processors and in our later lecture we be looking at the INTELS SGX which is INTELS Trusted Execution Environment where enclaves are created in order to run sensitive codes in spite of the entire system being untrusted. In the embedded world arm has introduced this trust zoon architecture which is stands for Trusted Execution Environment which could achieve the same thing. So in the lectures that follow, we will be first looking at the ARM trustzone and then we will be looking at INTEL SGX, thank you.