Mandatory Access Control
Lesson Introduction

- Understand need for **mandatory access control (MAC) and multi-level security**
- Explore several **MAC models**
- Understand **assurance techniques for a trusted computing base (TCB)**

We discussed access control in the last lesson. But that was only one kind of access control called discretionary access control, or DAC. That allows you to decide who you want to share a resource with when you create the resource and own it. If you work for a company, or a defense or intelligence organization, you don't get to decide how information that you access can be shared. The company or the agency will mandate how various types of information is accessed by different users. For this, we're going to develop a number of mandatory access control, also called MAC, models. Once we're done with, this we're going to revisit the idea of a trusted computing base, or a TCB, and explore how we can evaluate the level of assurance that is provided by such a TCB.

Discretionary Access Control

In discretionary access control (DAC), **owner of a resource decides** how it can be shared

- Owner can choose to give **read or write access to other users**

In the last lesson, we focused on access control, but in particular what we called discretionary access control.

The idea there is that access control is about checking if the source of a request should be able to gain access to a resource. We know that who the user is because user comes and authenticates himself or herself. So in discretionary access control, we said whoever creates the resource is the owner of that resource. So the owner can also selectively grant access to other users. For example, Bob may both be able to read and write, and Charlie can either read or write or maybe just has read access or something like that.

So the idea of a discretionary access control model is that resources have owners who are users. And these owners get to decide, it's at their discretion, who else should be able to share these resources.

In this lesson, we're going to examine if this model has any limitations or any problems. Are there situations where it's not going to do the job for us? And if that is the case, then what other kind of models do we need to develop to meet those applications, or those needs that arise when we talk about access control?

Discretionary Access Control

Two problems with DAC:

- You cannot control if someone you share a file with will not further share the data contained in it
  - **Cannot control "information flow"**
- In many organizations, **a user does not get to decide how certain type of data can be shared**
  - Typically the employer may mandate how to share various types of sensitive data
- Mandatory Access Control (MAC) helps address these problems

In particular we're going to focus on two problems that DAC has.

The first problem is that although the owner of a resource, owner of a file let's say, owner happens to be Alice, controls who this file can be shared with, and Alice gives Bob access to read this file. The problem arises because Alice is not going to be able to control if the information is going to stay with her and Bob only. So the problem is, can Bob share this information further with somebody else?

One problem that we have here with the DAC model is that although we can control direct access to our files, but how that information can be shared with others we're not able to fully control that. In particular, this problem is called the information flow control problem. So information was shared with Bob, flows from Alice to Bob, but can it flow further? And in the Discretionary Access Control model we're not able to control this flow going from Bob to Charlie.

There is another problem that Discretion Access Control has and this is because of the idea that resource is owned by the user and it is at his discretion who it can be shared with. Well, in many organizations at companies it's really not at your discretion. You are an employee of a company and the company actually controls how certain type of data should be shared in that company or outside of that. In other words, the employer of the company where the user works actually may be able to mandate or will mandate actually how certain kinds of sensitive data can be shared.

So to address these two problems, we actually going to explore another model for access control. It's called the Mandatory Access Control model and it's called Mandatory Access Control because it is not at the discretion of the user. Discretionary Access Control is at the discretion of the owner of a resource. But here organization is going to mandate how data can be shared and that's why it's called Mandatory Access Control.

DAC Quiz

Check the best answer:

In a certain company, payroll data is sensitive. A file that stores payroll data is created by a certain user who is an employee of the company. Access to this file should be controlled with a...

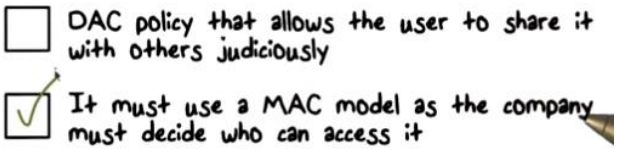
- ☐ DAC policy that allows the user to share it with others judiciously
- ☐ It must use a MAC model as the company must decide who can access it

In this one we're talking about a company, in particular payroll data, who gets paid how much. And so we have a file that stores data that is created by a user who works for the payroll department. And obviously this user is an employee of the company. But let's say it has a compensation information.

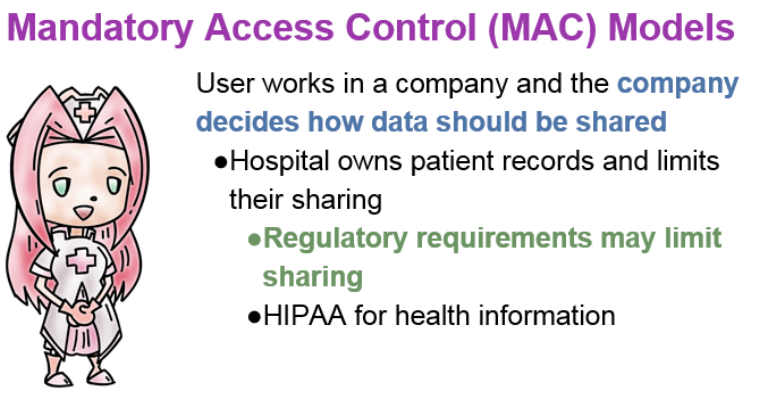
By the way, there is a large company right next to the

Georgia Tech campus had a situation where people's salaries were leaked. And that happens when somebody shares the information, either by mistake or by design, with others of whom this information should be shared.

So here we're talking information of that kind. Access to it has to be controlled. So what kind of policy is going to help us provide the right kind of access control for it. There are two options. It could either be DAC or it has to be MAC. So I want you to think about it and then we'll discuss which answer is the right one.



Here we talked about the nature of the data, okay? It's sensitive data who gets paid how much. Company obviously doesn't want to disclose that to people for whatever reasons, even within the company or outside. So the user who is creating this file is obviously should not be able to share it with whoever he or she feels like. Even if they try to be cautious and careful, remember, they can make a mistake or they may not be. We talk about going back to threat models, talk about insider threat. Someone can be unhappy and do something that could harm the company. So it shouldn't be at the discretion of this user who is creating a file. This kind of data really needs to, the company needs to, not just mandate but control how it can be shared. So, really, this is an example of where a mandatory access control model would be the right one for controlling access to the data that is in this file that has people's salaries. DAC and MAC, these are two different models. And this is a scenario where DAC is not going to be able to address or meet the needs that we have for how this data should be accessed.



Well, what exactly is mandatory access control? Why do we need it?

I think we already talked about that, although the users, its people who actually create information and run applications that manipulate such information. But people work for a company or a ~~garment~~ government agency, we'll talk about that. And it's really their employer and somebody that's trusted by them already acting on behalf of them, should really be able to specify how certain kind of data can be shared.

First of all, in taking the power of deciding how the data should be shared from the user, and, essentially, you can think about the user being told the right way to share this data is based on this policy that the company has. And this policy could be, it may depend on the kind of company that we have.

So, for example, hospitals with electronic medical records and stuff like that clearly have patient records. And we know that patient health information, their health conditions, information about them is highly sensitive, and who can see it has to be limited, and how it can be shared has to be limited.

- **Regulatory requirements may limit sharing**
- HIPAA for health information

And it's not really up to just the hospitals, they would want to do it. But there are regulations that force them to do this or force them to control how patient records can be accessed. So in the context of so are held information, we have HIPAA, which is the regulation the government imposes on anyone, entities like hospitals that collect information that goes into patient records, medical records. And of course, there's a need to share that information. If you go see a different doctor, they need to have access your medical information. But how access to it should be created? It's not the particular person who does data entry for a patient's record. They don't get to decide who else can see. This has to be decided by the hospital that's informed by this regulatory policy called HIPAA. This is protecting people's health information, privacy, and things like that.

We said in mandatory access control, it's not just the user owning a resource and being able to control, but it's really the user's organizational company. Hopefully, this example information, being medical records, convinces you that sharing of information has to be mandated by someone other than the user who's creating it.

Implementing MAC

Labels: A Key Requirement for Implementing MAC

- indicate sensitivity/category of data or clearance/need-to-know requirements of users
- TCB associates **labels with each user and object and checks them when access requests are made**
  - Need to relate labels to be able to compare them
- Exact nature of labels **depends on what kind of model/policy is implemented**
  - DoD models include classification/clearance level and a compartment in the label
  - Commercial policies are different but use labels to deal with conflict-of-interest, separation-of-duty etc.

So, before we talk about this different kind of models, and we will talk about a number of them, let's sort of explore what is new thing that we need for implementing them. So, we're obviously, I said, going beyond discretionary access control. Well, implementation of this mandatory access control model is going to require some new functionality, some new mechanisms the system has to now include.

So one of the new things that we're going to talk about all the time is, what is called sort of these labels that we're going to associate. That's a key new implementation requirement for defining these models and for implementing these models. But idea is that we're going to see that both users and resources of documents are going to have certain labels attached or associated with them. And they're going to get used in the way we make access control decisions.

So what's the purpose of a label that you want to attach to a document or associate with a user? The label is actually going to tell us how sensitive certain information may be. Labels also actually have something that describes the nature of the data. What topic, what area does the data come from? We talked on payroll as another example, before. So labels could also include sort of a category of the nature of the data that we have here. And that's sort of useful in answering this question, who needs to know or needs to have accessed? If you're in the payroll department, of course you need access to payroll information. Labels with documents, or data files, or users, are going to capture something about the information that's contained in those documents and the users who need to access them.

So these labels are actually going to get manipulated each time access to an object or document is requested. So TCB is actually not going to associate, but it's going to use these labels associated with user and the object any time the request is made.

- TCB associates **labels with each user and object and checks them when access requests are made**

So when I say the labels look right, of course we have to sort of say, here's the label for the user, and here's the label for the document. Should the user be able to access this document, to be able to read it? So we have to sort of relate or compare the labels, and we're going to see results of that comparison. But you're going to say, here is user with this label, here is a document with this other label. Does the user's level imply that he can access this document? So we have to relate, compare, the labels that we have.

- Need to relate labels to be able to compare them

Now exact nature of what the labels look like, and how we are going to be able to compare them, and what the result of the comparison is, well, that's going to depend on the particular model or policy that we're going to implement. We're going to look at several different examples, and each one of them has a very different kind of a label, and how you make use of that label. A label goes with, not just the document, but also with a user. And we'll see examples of those.

- Exact nature of labels **depends on what kind of model/policy is implemented**

So in the Department of Defense, we're talking about the labels actually going to include. For a user, it'll be their clearance level. For a document, at what level the document is classified. And compartment is this category that we're talking about. What kind of information is contained in this document?

- DoD models include classification/clearance level and a compartment in the label

Of course, in the commercial world, we don't have clearances and classifications. So these labels have to look different. And the concerns are going to be of a different kind, also. For example, we may have conflict-of-interest concern. So access control must be such that when somebody has a conflict, they don't get to access a certain document where a conflict-of-interest, or COI, may arise. Similarly, sometimes we may have separation-of-duty requirements. The same person can do two different things, when allowing them to do that requires a level of trust or potential for fraud and things like that. So, the needs would be different in commercial versus DoD, or garment DoD intelligence agency.

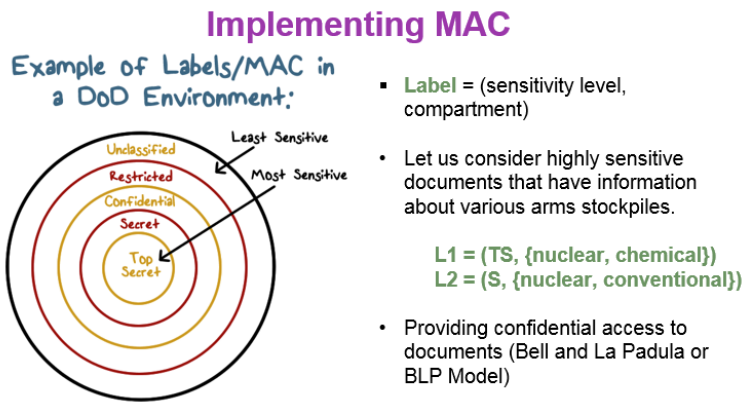
- Commercial policies are different but use labels to deal with conflict-of-interest, separation-of-duty etc.

Mandatory Access Control (MAC) Models

Military and intelligence agencies:

Data has **associated classification level** and users are cleared at various levels

- Top secret, secret, confidential etc.
- Limits on **who can access data at a certain level**
  - User cleared only at secret level should not be able to access top secret data
- Also called **multilevel security (MLS)**



So we actually going to explore this idea of labels that a mandatory access control policy would use in a DOD, Department of Defense kind of environment. A little bit more by looking at some examples.

So information is going to have classification and users going to have clearances, okay? So the highest sensitivity typically is top secret, that's the innermost one then secret, confidential and going out we have this unclassified we said this information anyone can see. So most sensitive, least sensitive, this is. We said, the level at which has somebody is cleared or the classification level of the document.

So we said label is going to be a sensitivity level. So for example, it could be top secret here and a compartment that describes what kind of data is contained in the document. So are we going to, let's say look at documents one, that is top secret. And let's say these documents have information about various arm stock piles. So this document let us talks about nuclear and chemical weapons. You hear about them in the news, so this is clearly extremely sensitive. So we going to say that is top secret and the document contains information about both nuclear and chemical weapons. We have another document that has information about let's say conventional. So maybe it's sensitive but it's a notch below. So let's say this is only secret. So we have two labels here. L1, L1 sensitivity level it's TS comma the compartment is nuclear, comma chemical. Remember this compartment here looks like a set, okay? So their document may pertain to multiple topics or areas. Each one of them could be in this and that's the description of the document using the set of terms that we have here. So:

- L1 is (TS, {nuclear, chemical})
- L2 is (Secret [S[, {nuclear, conventional})

Using these labels L1 and L2, we're actually going to be able to decide axis control in the end comes to being able to read and write documents because that's what you do to files or documents.

So we actually, when disclosure or confidentiality is an issue, there is a model, Bell-LaPadula model, or BLP model, that makes use of labels exactly like the L1 and L2 that we have here. And actually the model is defined by a set of rules that tell you when, based on these labels, you can read or write. But implementation of the mandatory access control, is always going to require some kind of labels.

Health Data Quiz

Check the best answer:

A hospital is found to be lax in securing access to patient records after it suffers a major breach. It may have violated the following policy:

- ☐ HIPAA
- ☐ BLP

We're going to use this quiz question actually to reinforce the idea that mandatory access control applies in nonmilitary setting as well.

So hospital is found to be lax. Unfortunately, there are examples like these that you read about in the newspapers almost on a regular basis. So they were lax in securing access to patient records and they have a breach. And a breach means basically someone who's not authorized to access those records is able to do that or is able to gain access to them. That's not a good thing. So the hospital has violated some kind of a policy. And that policy, really, we talked about in healthcare regulation, is what requires that you handle data with care or in a certain way. So this quiz question basically is what policy may the company have violated? And there are two options. Think about each one of those and then we'll discuss the answers.

Since we talking about health information, remember the BLP model was talked about in the context of military or intelligence setting where we have. We're going to see that confidentially is the reason for it, but information that's classified and things like that.

So this is not for the health care environment that we're talking about. For health care, we actually have HIPAA, that's the regulation. And HIPAA is the Health Insurance Portability and Accountability Act, I think goes back to the mid-90s when this was put in place to protect patient privacy and so on. This HIPAA requirement is why hospitals must have these policies for controlling access to patient records and those policies are often mandatory access control nature that we're talking about. Just because someone did data entry about a patient record, of course they create that file but they don't have the discretion to decide now who all they can share it with. It's governed by the hospital's policy on how patient records should be accessed. So that's what makes it mandatory. So the answer here would be HIPAA because we're talking about patient records in the hospital setting.

Security Clearance Quiz

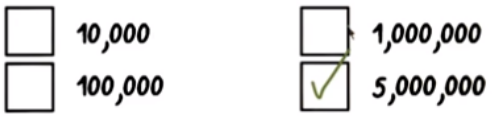
Check the best answer:

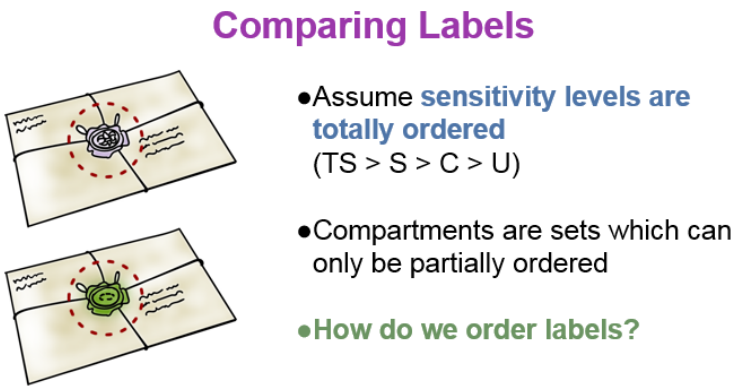
Highly sensitive defense or intelligence information should only be accessed by cleared personnel. Approximately, how many people in the United States have various types of clearances?

- ☐ 10,000
- ☐ 100,000
- ☐ 1,000,000
- ☐ 5,000,000

There's an interesting story in the newspaper a while ago, about how exclusive are these people who have access to classified information? It looked at various types of clearances people have, so one of them was how many people actually have secret clearance. The question is, in the US, there are various types of clearances, but how many people actually have such clearances? Is it pretty small, 10,000, or runs in the millions? I want you to just go with your gut and come up with a guess. We'll see whether you're right or not.

One thing that was interesting or surprising in this story that I just mentioned is just the large number of people who seemed to have these clearances. Actually, the answer here is the highest possible number we have in these different options. Close to 5 million people currently hold clearances. So it's not a very exclusive group of people. If you talk about 5 million people who have access to secret, or depending on their clearance level maybe top secret, information. The idea here is that people need to have clearance levels, but it looks like a lot of them do actually have that.





Now we talked about the need for labels when you want to talk about MAC, or multi-level security. And we even said, well, user has a certain label, file has another label. And we have to say, we have to check, if user with this label L1, should he or she be able to access the document that has label L2? So we have to sort of compare these labels. So here, we're going to explore how that comparison might work.

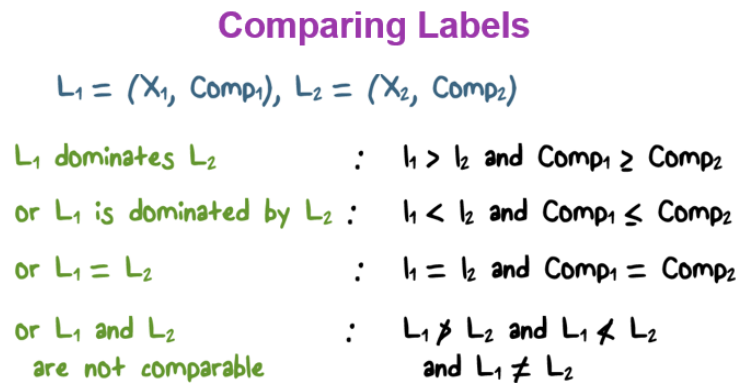
So let's just talk about labels of the type that we discussed in the context of the military intelligence community and how they access information. So, sensitivity, how sensitive data is. We said that, we had those circles, top-secret is the most sensitive, secret, next level, then classified, and unclassified. We had restricted in the middle there, but I'm just picking these four examples. So what does this greater than mean? So we are saying that, these levels are actually ordered totally. And by total order, what we mean here is that if I pick any two of these, I can tell you which one is more sensitive than the other one. So TS is the most sensitive. If you took TS and any other of these levels, then we know that this is higher, or is greater than, the other level. Similarly, secret is greater, or higher, than classified. So this greater really means its higher level of sensitivity. And total order basically means, when you pick any pair so the relations and various kind of orders that you can define for the elements that are in a given set, that's where it defines a relation. So the total here means you pick any two elements of this set, and there is an order between them, so one is always going to be more sensitive than the other one. Here we see that through this greater than sign, symbol that we have.

(TS > S > C > U)

We do know that labels have a level, sensitivity level. But they also have a compartment. And we said compartments are sort of sets of different topics or categories. And how do you compare sets? You take set S1 and S2. S1 may be contained in S2, S2 may be contained in S1, or neither set may be contained in the other one. So if set S1 has some elements that S2 doesn't, and vice versa, then neither set will be contained in the other. So when you have sets, the way you compare them is containment, or subset, or is one a subset of another one? And we know that that order is a partial order. So total order is when, pick any two, and one always is greater than or higher than the other. When you talk about sets, we know there are going to be pairs of sets where neither one contains the other one, and they're not going to be ordered, so that's called partial order. So we'll explore this a little bit more through examples. But

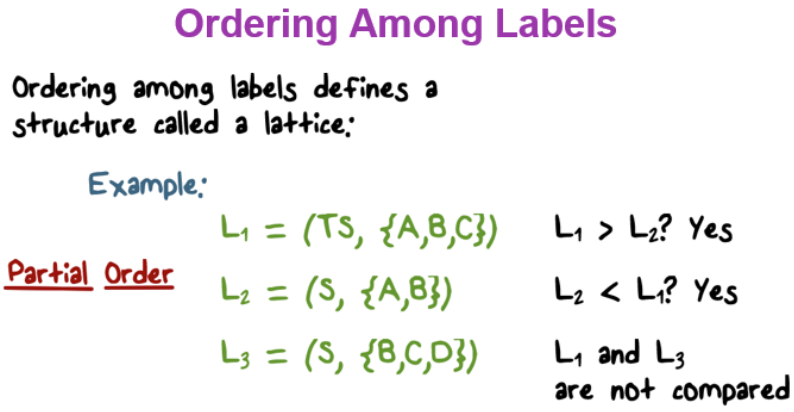
if a label is going to have two parts, one part is totally ordered, one part is ordered sometimes and not comparable at other times, so that's the partial order.

So when we talk about actually ordering labels, we had to look into the label. We have to look into the sensitivity level. What are the sensitivity levels? If the labels that we're comparing is L1 and L2, then we are to look at L1's sensitivity level and L2's sensitivity level and how they compare. And then we're going to look at the compartment of each one of those, and then compare those compartments. And by comparing both parts, then we going to be able to decide how to order the labels.



No transcript.

Lattice ✓
LUB, GLB



Comparison among labels, just look at a quick example.

L1 is TS and its compartment has these three categories, A, B and C. L2 is secret with A, B and L3 is secret also with B, C, D as their categories.

So does L1 dominate L2? TS is dominate S and A, B, C contains A, B. So, we said that each, for L1 to dominate L2, the sensitivity level has to be greater, which is the case, and its compartment has to contain the compartment of L2, which is the case. So this is, yes.

Does L2 is less than L1? Yes, the same reasoning we did. It's less because this is less than TS and this is containing that.

How about L1 and L3? As we know, TS is higher than S. So when you look at the first part, just the sensitivity level, that's the total orders, so this is greater than this. But B, C, D is not contained in A, B, C, okay? So there's this element D that's not contained here and because this is not contained here then L1 is not greater than L3. Similarly, L3 is not greater than L1, and the two are not equal because the compartments are different and the sensitive level is different. So, when neither one dominates the other one and they are not equal, then they can not be compared, which is the third case that we're talking about.

Order Quiz

Select the best answer:

The "<" relation among all real numbers defines a...

- ☐ Total order
- ☐ Partial order

And this one, we talked not about the labels that we were just discussing but we're talking real numbers. So these are number, integers, rational numbers, we have real numbers, it's all the numbers. So take any two numbers and there is this thing relation called less than it. So, number 1.5 is less than 2.5, for example. So, if you look at the less than relation and real numbers, does it define a partial order or does it define a total order?

- ☑ Total order
- ☐ Partial order

For numbers, the answer is actually going to be total order. If the two numbers are different, then one is definitely going to be greater than the other. Or, in this case, actually we're talking about definitely going to be less than the other. We don't have partial orders when you just look at real numbers and the relation is less than because you take any two Numbers. As I said, if they're different, than one is definitely less than the other. So you'll be able to order them. When you can order any pair of these numbers that means you get a total order.

Label Domination Quiz

Select the best answer:

If L1 = (secret, {Asia, Europe}) and L2 = {top-secret, {Europe, South-America}),

- ☐ L1 dominates L2
- ☐ L2 dominates L1
- ☐ Neither L1 nor L2 dominates the other one

In our next question, actually, we are talking about the kind of labels that we have seen, that we discussed.

So, two labels. Remember, I keep talking about labels. The uppercase L labels are the secret or top-secret sensitivity level, so L1 is secret comma, this is pretends to since this comes from the DOD, has a different command, so let's hit towers Asia and Europe. L2 is something about Europe and South America. So one is secret, the other is top secret. So there are three options and I want you to look into each and if it is true, then you check that box.

- ☐ L1 dominates L2
- ☐ L2 dominates L1
- ☑ Neither L1 nor L2 dominates the other one

So let's look at, does label L1 dominate L2? It doesn't because secret actually is not higher than top-secret. The domination fails right here. This doesn't contain this either, because there's no South-America in here. So L1 doesn't dominate L2. Does L2 dominate L1? Well, top-secret actually is higher than secret but the compartment here doesn't include Asia, so the containment doesn't hold. So L2 doesn't dominate L1 either and the two are different, so actually, in this case we're going to run into the third option, which is the correct one. Neither L1 nor L2 dominates the other one and they're different, so they can't be compared.

Sensitive Data Quiz

Select the best answer:

Assume that label L1 or a document D1 dominates label L2 of document D2 when these labels are defined by (sensitivity level, compartment).

- ☐ D1 contains more sensitive data than D2.
- ☐ D2 is more sensitive than D1.
- ☐ The data contained in D2 has a narrower scope as defined by its compartment

So this is Sensitive Data Quiz. Again, select the best answer. Assume that label L1 of a document D1 dominates label L2 of document D2. There are two documents, D1 and D2. The label are L1 and L2, and these labels are defined the way we've been discussing. So these are, let's say, DOD kind of documents. So sensitivity level could be top secret or secret and compartment is what kind of information is contained in this document. The question here is that L1 dominates L2. It dominates L2, so we know L1 is greater than L2. This is sort of what is the interpretation on that domination. I guess that three options sort of present three different interpretations and you have to say based on what we understand about label domination which one is the correct interpretation.

L1 > L2

So if Label L then dominates Label L2, this is because either the level is higher. So, for example, this could be top secret and this could be secret. If that's the case, then D1 would contain more sensitive data than D2. Okay. So this actually is correct.

☑ D1 contains more sensitive data than D2.

When D2 is dominated by D1, which is the case here, because L2 is less than L1, or L2 is dominated by L1. It cannot contain more sensitive data, so the second option is not correct.

☐ D2 is more sensitive than D1.

The third option says the data contained in D2 has a narrower scope as defined by its compartment. So what exactly does narrower scope mean? Remember, categories define what the more categories the data, at least the way we define a label comparison, so that one set contains another set means this information is about more topics or more areas or something like that. So, since L1 dominates L2, the compartment of L1 has more elements in the compartment of L2. Okay, remember L1 compartment, if I use this notation, is a superset of L2 compartment. So in this case, we know that the categories that we associate with the data that's in D2 are going to be fewer. So that's a narrower scope and we can say this is true.

L1 > L2

L1.comp ⊇ L2.comp

☑ The data contained in D2 has a narrower scope as defined by its compartment

Using Labels for MAC: Confidentiality

- **Bell and La Padua or BLP Model** (Developed by DoD)
  - Assumes classification of data (TS, S, C, U) and clearances for subjects
- **Read/write rules**
  - User with label L1 can read document with label L2 only when L1 dominates L2
    - **Read-down rule (simple security property)**
  - User with label L1 can write document with label L2 when L1 is dominated by L2
    - **Write-up rule (star property)**

We've been talking about labels and their use in mandatory access control. Now we're going to get to a concrete model. This is a model that deal with confidentiality and it was introduced earlier is the Bell and La Padua model. Project that developed this model was funded by the Department of Defense, so the kind of label we're going to talk about are the type we've been discussing, where you have classification and clearance and things like that.

Couple of things to keep in mind here is that confidentiality is the issue here. We're concerned about the disclosure of information. So more sensitive information we don't want that to be disclosed to someone who is not cleared at that level. So that's going to be one. And since we're talking with DoD the other thing to keep in mind is the labels what they're going to look like.

As I said, the DoD means it's going to assume that information or data that's contained in files is classified at levels like TS [top secret], secret, and classified and so on. The second part of a label is a compartment or category that as we said before. It depends, what is the document about. So is it chemical or nuclear or conventional or two kinds or something like that.

And the rule for read says user with label L1, top secret user, for example. Who's working on, let's say, nuclear and conventional. So that would be this user's compartments. So user with label L1 can read document with label L2 only when L1 dominates L2. So this domination that we defined before when the label of the subject or the user dominates the label of the document, then they're able to read it.

This is called the read-down rule. You can read documents that are classified at your or a level that's below yours. Or further down from your level. So that's the read-down rule. It's also called the simple security property.

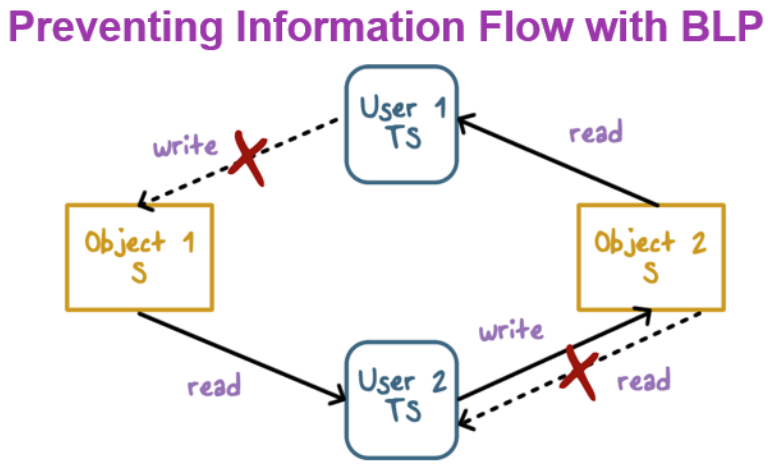
So how about writes? Because you both need, you need to address both read and write. So user with label L1 can write document with level L2 when L1 is dominated by L2. This is saying the user has to be at a level that is lower than the level at which the document is classified. Which means if the user is secret, he can write a top-secret document, he cannot write an unclassified document.

This is the write-up rule. So you can read down, but you have to write-up, and it's called the star property.

So let's just see what's the sort of rationale for writing up. Remember, our focus is on confidentiality or disclosure of information. This model says more sensitive information cannot flow. Well, the information flow control requirement that we had. So more sensitive information should not flow to users who are not cleared at that level. It's basically focusing on disclosure. It doesn't say anything about who can write information. That's sort of an integrity question and we have to revisit that perhaps. But if I can write at my level or at a higher level. If I could write at the lower level, then what could go wrong?

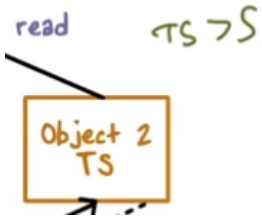
Well, I may have in my possession some information that is more sensitive. If I write it into a document that's at the lower level, then someone who's cleared at the lower level can in the future come and read that document. Which means we would have information flow from a more sensitive level to a level and to a user who should not have access to that.

So, you can't allow write downs. You can question why would an unclassified user have information that is top secret? But if they did write a top secret document, a top secret user would read it, as information flow from unclassified to top secret. But that doesn't, while with the confidentiality requirement that we have. So think about, a request comes. User has the users label. The target is a document that the user wants to access, had its own label. We can do this comparison if the read request is, user is making a read request. Then the user label better dominates the documents label, to satisfy the simple security property of the read-down rule.



Now, we want to revisit that question of information flow. And remember, recall that information flow is a problem that cannot be addressed by discretionary access control, okay. So we want to see how mandatory access control, or MAC, or multi-level security, in particular the BLP model that we are focusing on. And so how does the BLP model help us solve that information flow problem that we had?

So we say we have an object, this is a top secret object. So we have a top secret object 2 is TS. This is more sensitive information. We never want this information to go and end up in an object that has level is only secret. So this is more sensitive object. 2 is more sensitive than object 1 because we said TS is greater than S. So this is highly top secret information. We never want it to end up in an object that is only classified at the secret level.



Okay, so how can this information flow from here to there? There are two paths. Path one is that a top secret user reads it. We know that the read down rule means what? For this to read, for someone to be able to read this, the user has to be TS, otherwise they can't read it. A secret user can only read secret or below. So, a top secret user can read this object. So, this read will be allowed. And then they, can this information that they have pulled, remember the Alice, Bob, Charlie example in discretionary access control we had? Where Alice gave read access to Bob, who copied the data into a file that he created and then passed it on to Charlie? Well, can User 1 do something like that? Can it create Object 2, in which it may be able to write this information that it just read from Object 1. For someone who is not top-secret to be able to read this information, the object we create has to be secret or lower. TS then is just like here. They won't be able to read Object 2, they won't be able to read Object 1. This object is at the lower level, then the write rule says what? You can only write up. You can't write down. So all the top secret user can take this information, read it. But they're not able to write it in an object that can be

read by somebody who's not cleared at top secret. So this path doesn't allow informational flow to get to a lower level.

What about a secret user? We know the secret user can, once it's here, can read that. I guess we know that a secret user can't read. So, this path we said is not possible. This path actually ends right here because the secret user is not able to read a top secret document. This label would not dominate the label of the object. So this path is broken here, this path is broken right here, so this information actually never flows in here.

And information from Object 1 flowing to Object 2. Secret can read this object and can write up, yes. But that doesn't violate any confidentiality requirements. So secret information can now be read by a top secret person. This information, made its way from here, through user 2, into this object. The top secret could already read this, read down. So, in this direction, information of flow doesn't make more sensitive information available to a user, who is cleared at a lower level. The system would not allow information that is read from this object to be written into an object that is at the lower level. That's a guarantee that comes with BLP and the MAC model that we didn't have with DAC before. So this is how we're able to control information flow, and that's how we fix one of the shortcomings that we had with discretionary access control.

Unclassified Documents Quiz

Select the best answer:

Since an unclassified document contains no sensitive information, it can be read or written by anyone in a system that implements BLP

- ☐ True
- ☐ False

Unclassified document contains data that is not sensitive. So it could be read or written by anyone in a system that implements BLP. You can read down, so anyone can read it all the way from the top to the unclassified levels. It can be read. Can it be written by anyone? Well, so it can be read by anyone or written by anyone. The question is asking, is this statement true? Unclassified document contains nothing that's sensitive, so it can be read or written by anyone. It is true. Okay, so pick the right option in this quiz.

The answer actually is false. The statement is false. It's true that it can be read by anyone because of the redound rule, but it cannot be written by anyone, okay? So this is not correct. Okay. This part is wrong. So it can be read or written by anyone. That's, that's not true. It can only be read by anyone. So read is yes, true, anyone. So to write, it can only be unclassified. Okay. No one at the higher level is ever to write into a document because if they can write into this document then the information can flow from a more sensitive level to this level.

Classified Data Quiz

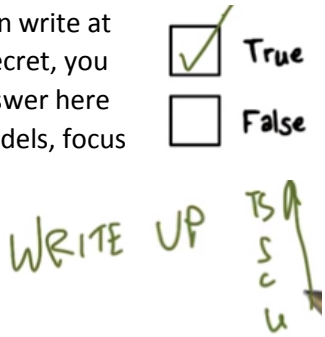
Select the best answer:

BLP allows an unclassified user to write a top secret document.

- ☐ True
- ☐ False

Our next quiz question talks about classified data, and in particular, let's say this is a top secret document that is what we're talking about. So it says BLP allows an unclassified user to write a top secret document. Is it true or false?

The answer actually is true, because remember we have write up rule. You can write at your or higher level. So unclassified user is down here, classified secret, top secret, you can write at any level. Write up rule is the star property that we have. The answer here is true. Again, if you think something is not right here, think about the BLP models, focus is on confidentiality or disclosure of information. An unclassified user has no sensitive information and gets written at whatever level. They may want to choose to call it top secret perhaps, but that is not going to disclose any sensitive information to somebody who should not have access to it. So, the write up rule is the reason why the answer to this quiz question is true.



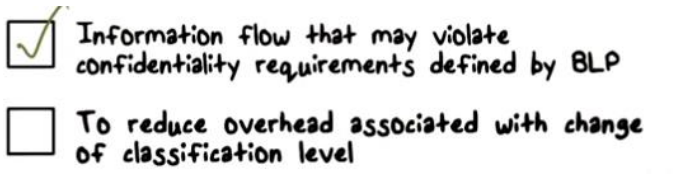
BLP Model Quiz

Select the best answer:

Tranquility principle in the BLP model states that classification of a subject or object does not change during a session. This principle is needed to avoid...

- ☐ Information flow that may violate confidentiality requirements defined by BLP
- ☐ To reduce overhead associated with change of classification level

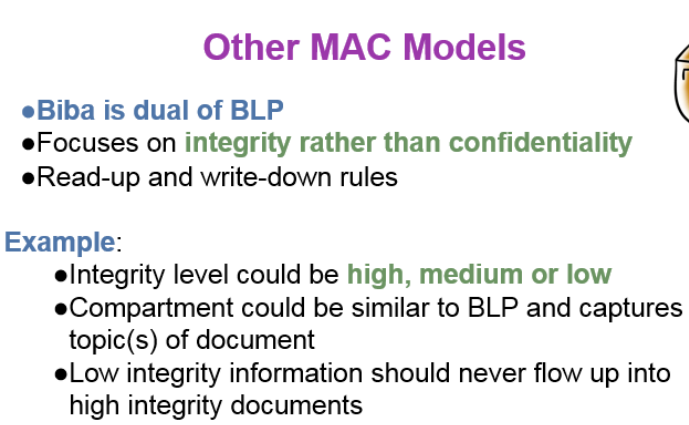
As you sort of talk about a more complete definition of this model, there is another requirement which is called the tranquility requirement in this model that says classification of a subject or object does not change during a session. You can't change the label associated with this. So this principle is needed, one of two reasons. Okay, so the two options we have here. So try to give the reason why we need to have this requirement, or need to have tranquility principle has to be satisfied. So you have to choose the right option, and sort of explain why.



The correct answer is the first one. Information flow is the reason why we have this requirement. Let's say your label can change, then let's say you were, some full example is that you are top secret, so you were reading top secret information, and then your label changes to secret light set. You are the user. Well, now you can write at the secret level, so top secret information that you could read previously can now be written to documents that only have secret classification level. Then the user, later on, who's cleared at secret level, can come and read this document into which information might flow from top secret to the secret level, as a result of the change that happened in the label during your session. So, if you do that and sort of do a couple of other examples, labels change during a session. Then, where you were reading, where you can write, all that changes. And because of that, we could have information flow from a TS object 02 that we have in the example to a secret object 01 and we don't want that. So, information flow is the problem.

To reduce overhead associated with change of classification level, you do not change the labels, and the reason for that is nothing to do with overhead associated with changing them. It really has to do with information flow, and the information flow that may violate the model that we have.

Think about what happens when a user's security level changes.



So we've been talking about sort of MAC models and BLP, Bell-LaPadula is one example of that. That's the Biba model. Anytime we talk about a model, you have to say what's the nature of the labels? How do you compare them? What kind of rules do you have? Similar to what we had in the BLP model.

The Biba model actually is dual of the BLP model. And this duality is because rather than focusing on confidentiality, which is what BLP focuses on, Biba focuses on integrity. Integrity is defined, somebody being able to write or corrupt data. The interpretation here is that, when you say it is top secret information, it is really top secret information or something like that, or the information indeed should be at that level. So think about the sort of, quality of information rather than disclosure of information that confidentiality focuses on.

So it's dual because the rules actually are opposite of what we had in BLP. When you're concerned about information quality or integrity you want to read-up. Because anything below you is lower, I mean information with lower integrity, you don't want to read that. So reads in BLP you have read-down, here you're going to have read-up in the Biba model. BLP you have write-up, in Biba you have write-down. So if you're at a certain level, well, the integrity of the information that you can create and then write is either at your level or or it's less integrity than the level at which you are. So you're able to write down. Again, we're not concerned about confidential disclosure here. We're concerned about information integrity.

Okay, so let's look at an example to clarify this a bit. Let's say integrity could be high integrity, medium integrity, or low integrity. So saying information quality, think of your favorite newspaper that you trust a lot, let's say for me, it may be the New York Times, the information that appears there is highly likely that I can trust it. Your supermarket tabloid, let's say at the other end. Apologies to anyone who may be a fan of those. When you're talking with confidentiality, anyone, it's not about controlling who can see it. It's controlling how good the information is. So here it's high, medium, or low.

Example:
- Integrity level could be **high, medium or low**
- Compartment could be similar to BLP and captures topic(s) of document
- Low integrity information should never flow up into high integrity documents

Compartment could be similar to BLP, depending on what the information is about. Really, it's the topics of the document that contains that information. So label here. So anytime I quote a model I said I want to talk about, what does the label look like.

So, high, medium, low is ordered again, totally ordered compartments, partially ordered, because they are sets. And, the model says low integrity information should never flow into high integrity documents.

Policies for Commercial Environments

- **User clearance is not common**
- Other requirements exist
  - **Data only be accessed by certain application** (e.g., payroll)
  - **Separation-of-duty and conflict-of-interest requirements**

But what about policies in commercial environments?

First of all, when you go work for a company, they might do a background check on you, but they don't clear you at secret or top-secret levels. So clearance is really not common when it comes to these types of environments.

However, there are other requirements that may arise in these kind of places. If we talked about, we go back to our payroll. If you are a payroll department employee, then you can access applications that have to do with payroll. But others should not be able to do that.

So one requirement could be that data is accessed by certain applications and only certain users be able to gain access to certain kind of applications. Requirement may be that non-payroll department people don't have access to the payroll application.

Other requirements that arise in these commercial environments is conflict of interest, is pretty common. So if you're a law firm, have two customers, and there's some competitive relations in between those, and a lawyer working on one case should not have access to data that belongs to the other company, when those two companies compete.

Separation-of-duty typically is good for reducing the likelihood of fraud. So if one person can do multiple things, it's easier for them to commit fraud. In commercial environments, clearance is not there, BLP's not very useful, but perhaps, the need for mandatory access control is there. So we want to look at some models that may make sense in this setting.

Policies for Commercial Environments

- Clark-Wilson Policy
  - Users → Programs (transactions) → Objects
  - Same user cannot execute two programs that require separation-of-duty

- Chinese Wall Policy
  - Deals with conflict of interest

So we can look at two different policies or models that make sense for commercial environments.

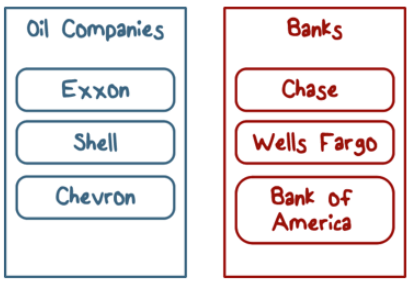
We're going to start with orders called the Clark-Wilson policy. So Clark-Wilson policy says that users should be able to access certain programs or applications. Objects are constrained and certain objects can only be accessed by certain programs. So the way you think about the policy is that somebody says this user should have access to this application, and somebody says this application should have access to these objects, so that's what the user can then gain access to those objects. And only when they're running those applications.

So if I want separation-of-duty, for example, where the same person should be able to do things that could potentially lead to fraud, then we can just say that user Alice cannot execute both of those applications. So, you can do one and somebody else will need to be able to run the other application and that's how we separate the two duties associated with the two applications that we have here. So, Clark Wilson essentially was developed for commercial settings.

The second one is more about the conflict of interest requirement, how we address that. And the name of the policy is called Chinese Wall Policy, and we'll see why this is an appropriate name for it. So the deals, as we said, with conflict of interest and what you can access or cannot access depends on what you have seen so far. So what you've seen so far, if it has no conflict with the next thing you're asking for then it's okay. If not, then we're going to deny access. So how do we decide when do you have a conflict or the absence of that conflict of interest?

Chinese Wall Policy (Conflict of Interest)

Objects are put into conflict classes:



The user **can access any object** as long as he/she has not accessed an object from another company **in the same conflict class**.

To explain this conflict of interest and the Chinese Wall Policy a little bit more, let's say we have a document on a system. Now some documents are financial documents that pretend to different banks. So Chase, Wells Fargo, Bank of America, whatever it is. So this grouping here, these are all the documents that are for Bank of America. These are all the documents for Wells Fargo and so on. Okay, similarly we have other documents that pertain to, let's say, oil companies. So some documents are Exxon, others Shell, Chevron, and so on.

The reason we have this outer box around all the bank documents or oil company documents is that this is a conflict class. Chase has a conflict with Wells Fargo because they're in the same kind of business so they compete with each other. So the conflict class is, you know, these documents form the conflict. Similarly, Exxon would have a conflict with Shell and Chevron. So these documents are in this other conflict class.

Okay. So Chinese Wall Policy basically says a user can access any object as long as he or she has not accessed an object from another company in the same conflict class.

The user can access any object as long as he/she has not accessed an object from another company in the same conflict class.

So let's say user Alice accesses Exxon, okay, a document pertaining to Exxon. She can because so far she has accessed nothing, so there's no conflict that we can see. As soon as she accesses a document from this class, from Exxon. She cannot access anything in these other classes that are in the same conflict class. So she can't access something that belongs to Shell or Chevron. Can she then go and access a document that pertains to Chase? Yes, that's in a different conflict class. There's no conflict between Chase and Exxon. Alice can access this and this. But as soon as she does Chase she can't access Wells Fargo, so you can basically access any oil company and any bank but not two oil companies and not two banks at the same time. So we're basically creating these walls around sets of documents, okay. That a conflict exists across those and that's why it's called a Chinese Wall Policy.

Clark-Wilson Quiz

Select the best answer:

Clark-Wilson is a mandatory access control policy because...

- ☐ Any user in a company can decide what files can be accessed by a program
- ☐ Only the company can decide (e.g., sysadmin) what files can be accessed by a program.

Let's do a quiz about Clark-Wilson. Why is it a mandatory access control? There are two possible answers I wanted to think about. So remember, mandatory is how things get accessed is not at the discretion of the owner or the user who creates a resource or a file. So we think Clark-Wilson is a mandatory access control. So why is that the case? What explanation do we have for that? Okay. So look at both options and see if either one or both of them help explain why Clark-Wilson is MAC.

- ☐ Any user in a company can decide what files can be accessed by a program
- ☑ Only the company can decide (e.g., sysadmin) what files can be accessed by a program.

If any user can decide what files to be accessed, then of course, there's no mandatory aspect to it. They should be able to decide about their files, not any file, but that doesn't make it mandatory access. The fact that only the company can dictate, somebody on behalf of the company, for example, sysadmin, what programs can access what files. That's what makes it mandatory, okay. So somebody, other than the user who creates a resource, is required to specify how it can be used. That's what makes it mandatory. And here, the company or company's representative, someone in the security organization, is deciding it which makes it mandatory.

CoI Quiz

Select the best answer:

A large law firm currently has two client companies that compete with each other. Thus, its lawyers working on cases related to one company must not be able to access documents related to the other company. To ensure proper access control, which policy should the law firm use?

- ☐ Clark-Wilson
- ☐ Chinese Wall

So we talked about a number of commercial policies. For example, a Clark-Wilson and Chinese Wall. So here we have a scenario. In this quiz, we have a scenario where there's a law firm that has two different client companies. And the interesting thing about these client companies is that they compete with each other. And lawyers work on cases related to many companies. So some of them work on company A, others work on company B, or whatever it is. And they need to access the documents. So to ensure proper access, in this case, which policy should you be using? That is the quiz question. The two options are Clark-Wilson and Chinese Wall.

- ☐ Clark-Wilson
- ☑ Chinese Wall

Clark-Wilson, remember, defines as transactions or procedures that you're able to execute, and you can define who should be able to do what and things like that, based on what they're doing in the organization. It doesn't address the conflict thing. The conflict thing is addressed. So, the correct answer is Chinese Wall. So remember, we talked about conflict classes? So similar, for example, airlines all because they compete with each other would be in a single conflict class. And then we said, as soon as you access one company's information, you're not allowed to access from any other company the same conflict class. That was the Chinese Wall policy. That's what would work here in this case, because if lawyers lurking on this competing companies, if the same lawyer does that, that is a conflict. We want to avoid that, and you avoid conflict of interest by using the Chinese Wall policy.

RBAC Quiz

Select the best answer:

Role-based access control (RBAC) is often used in a commercial setting. RBAC is an example of MAC because...

- ☐ File permissions are associated only with roles and not users
- ☐ Only the company can decide roles of its employees

So we discussed role-based access control and we also talked about discretionary and mandatory access control. So, this quiz question is asking you, if role-based access control, which is used often in commercial settings, is an example of mandatory access control and if it is, then how do you actually justify this statement that RBAC is in fact a mandatory access control policy?

- ☐ File permissions are associated only with roles and not users
- ☑ Only the company can decide roles of its employees

The correct answer says, well who has what roles gets defined by the company itself, okay. So whoever is responsible for secure access to its resources, trusted sys admin or whoever it is. So what role you can take on is mandated by the company and what you can access depends on the role. So, it doesn't depend on so, remember, discretion access control, the owner decides who can access their file or data that they created. Here that's going to be decided by roles have access and who can take on what role is going to be decided by

the company. And because of that, the fact that the company decides how accesses are going to be allowed, that's what makes this a mandatory access control policy.

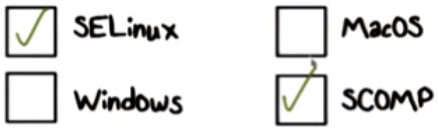
MAC Support Quiz

Select the best answer(s):

Which of the following operating systems supports a BLP-like model?

- ☐ SELinux
- ☐ MacOS
- ☐ Windows
- ☐ SCOMP

So we talked about mandatory access control models, multi-level security models. And talked with the BLP model for accessing classified information and that's done by individuals or users who are cleared. The one question you can ask, well, does anyone care about this model? And if they care enough do they implement? What operating systems actually implement this? So there are four options, consider each one of those and then we'll talk about it.



Well, MacOS and Windows are the commercial systems that most of you use. Neither one of them actually support BLP. These vendors choose not do that, for business reasons, or whatever. So security enhanced Linux or SELinux actually does do that. The NSA was of course keen on having operating systems that supported BLP-like models because they have information that is classified. And if the vendors are not going to do, they did support an open-source effort to build security enhanced Linux to include the SELinux kernel module that actually supports these kind of policies. So, SELinux is out there now. You can actually use it. SCOMP is a system actually that goes back to late 70s, early 80s. This was an effort between Honeywell and DOD and the goal was to do sort of hardware. So this is Secure Communications Processor, that's what this stands for. Was to build a system that actually can get an A1. So in a TCC criteria that we had, the version a, A1 it supports mandatory access control, verified design, and things like that. It's only the highest one, so that's what this system aimed for, and so it did obviously implement mandatory access control and in particular BLP-like models for it.

Trusted Computing Bases (TCB)

Revisiting Trusted Computing Base (TCB)

- How do we know TCB can be trusted?
- **Secure vs. trusted. vs high assurance**
  - Set of all hardware and software trusted to operate securely
  - Required for all other trust in the system security policy

First question we sort to have to ask ourselves is how do we know that a certain system that claims to be a TCB, or Trusted Computing Base, is actually, can actually be trusted?

So trusted is actually not the only set of qualified we use. We call systems secure systems, trusted, I've been using this word, but also high assurance systems.

So, security you can think of this as a binary property in some sense, either something is secure, or it's not secure and something can go wrong and while the security requirement. Trusted is typically sort of higher level of confidence that you have that it's going to do the right thing, what it is expected to do,

this level of trustworthiness, trusted our reliance on it and things like that. High assurance, basically, is a similar sort of idea. It does what it's supposed to do with a very high level of assurance.

- Set of all hardware and software trusted to operate securely
- Required for all other trust in the system security policy

I should say that the hardware that we have, you need some place to stand on. Hardware is obviously something that we're going to trust, and software that has direct access to the hardware, which makes up your operating system, the trusted computing base. We have to trust it and anything else that will be able to make claims about the level of trust that we have in the system is going to depend on the trust that we have in this underlying trusted computing base. So enforcement of a security policy because we implement access control in a certain way, or isolation of processes, isolation of untrusted code from trusted code. All that is going to rely on this level of trust that we have or the level of assurance that we have, that the trusted computing base would do what it supposed to do, and nothing more and nothing less.

Trusted Computing Bases (TCB)

Trusting Software:

- **Functional correctness**
  - Does what it was designed to do
- **Maintains data integrity**
  - Even for bad input
- **Protects disclosure of sensitive data**
  - Does not pass to untrusted software
- **Confidence**
  - Experts analyze program & assure trust
- **Statement giving security we expect system to enforce**
  - Do this formally when and where possible

We said the Trusted Computing Base has direct access to the hardware. Hardware, we said, we can trust, that's not something we're going to, although you may have reasons to, to, be suspicious, of hardware itself, but we're not going to go there, so we're basically talking about how can we trust software.

The software is supposed to perform, a set of functions. We've been discussing some of those, whether in a recently accessed control, or kind of policies that are supported and so on. So if it's going to implement certain functionality, it has to be functionally correct. Okay, when it says it implements a certain model, it implements that model correctly. The functional correctness basically means that it does what it was designed to do. So it implements a set of functions and that is done correctly.

Now, to implement those functions, it does have data structure. So data state that it makes use of. Also, it may be protecting other data that belongs to different users and programs, and things like that. If it's trusted software, we wanted to implement the function it's supposed to do correctly, as well as maintain and the integrity of that data that it is going to rely on. And we want it to maintain that integrity even in the presence of bad input. Okay, so we do explore some interface or API, hence the bad guys can try to use it in ways that we didn't expect it to be used, give it bad input for example. So the integrity that should be maintained by the trusted computing base actually is maintained in the presence of a correct model, but an adversity may even try to provide bad input.

Obviously, there is sensitive data that it uses and it protects for other users and applications. It has to address the confidentiality or disclosure of this sensitive data, it shouldn't be disclosed to somebody who doesn't have access to it. And it should do that in the presence of un-trusted software obviously. We're not going to assume that all software is going to be trusted. Okay, so we have our trusted computing base and then untrusted software that is run by applications so on. So this disclosure that we want to control for sensitive data can make assumption that all code is the system is trusted.

- **Confidence**
  - Experts analyze program & assure trust

So the last couple of requirements we identified in terms of integrity, confidentiality, implementing the necessary functions really talk about what it does. Our confidence basically says how well does it do that? Okay. Confidence, we can't demand a proof that's going to be really hard, a formal proof. But, as we said earlier there could be things that increase our confidence in the system's ability to do all these things that we're talking about. So where does this confidence come from? Well, a lot of things in real life, it could come from the fact there experts of these kind of stuff and evaluating this kind of a system. And maybe they analyze it, they spend time with it. And it's their word based on which we actually derive our confidence and the trust that is assured by the system.

- **Statement giving security we expect system to enforce**
  - Do this formally when and where possible

Finally, we can think about this as our requirements or needs, in terms of trustworthiness, can be maybe captured in a statement. And that statement says, well, this is the kind of security or trust we expect the system to provide, or this is what we expect the system to enforce. And then the system, someone who builds the system, design and builds the system. So let's say there's a vendor who provides the system. We said TCB functionality, normally operating system is where it goes in typical systems that we have, so an OS vendor, for example, may have to then tell us our requirements are met. We come with some sort of a requirements analysis, or a statement that sort of captures our expectation. And then somebody has to give us a checklist or a proof of some kind that allows us to believe that our expectations are met. And so the best possible scenario is that all this could be done formally. We actually have a proof that whoever is providing the trusted computing base actually meets our requirements, our expectations that we capture in this statement. When we have formal proof of it that would be ideal, but would it be possible always? This is one of the questions we're going to explore as we go through this lesson.

[*** no transcript ***]

TCB Design Principles

- **Least privilege for users & programs**
- **Economy**
  - Keep trusted code small as possible, easier to analyze & test
- **Open design**
  - Security by obscurity does not work
- **Complete mediation**
  - Every access checked, attempts to bypass must be prevented
- **Fail-safe defaults**
  - Default deny
- **Ease of use**
  - Users avoid security that gets in their way

Least Privilege Quiz

Select the best answer:

Least privilege is useful for damage containment when something goes wrong. Is this principle applicable to a TCB that must be trusted?

- ☐ No, because a TCB is guaranteed to function correctly
- ☐ Yes, because TCB only provides high assurance and not a guarantee

Is this principle applicable to a trusted computing base? Such a system has to be basically trusted. Everything that we want relies on trust that we place in the system. So, is this useful for a TCB is the question that we have here. We obviously talked about this principle being useful for applications and services and so on. So here, the question is that useful for TCB?

So the idea here, if something goes wrong, then this is useful. This one says, maybe nothing goes wrong when there is a TCB, when we're talking about the trusted computing base. The second option says, yes, it is relevant because TCB only provides high assurance and not a 100% guarantee. If you think about these two options, we said TCB is also a system that is fairly complex for a variety of reasons. We're going to do our best and use the design principals and do other thing for analysis and so on, but it's trustworthy. It's not 100% secure. So basically, it gives us this high assurance, but something could still go wrong. It helps to have this design principle. So if something can go wrong, even if it's a small likelihood, having this proves it's useful. So I think, the idea here is that you can't assume just because we have the word trusted computed base in TCB, that it is guaranteed secure.

- ☐ No, because a TCB is guaranteed to function correctly
- ☑ Yes, because TCB only provides high assurance and not a guarantee

TCB High Assurance Quiz

Select the best answer:

A TCB vendor claims its proprietary techniques help ensure high assurance, but cannot be disclosed. What principle does it violate?

- ☐ Complete mediation
- ☐ Open design

So, we have a vendor who claims some mystery techniques, proprietary techniques that we're talking about. So there's all of this debate between open source and closed software that is not open source. So proprietary is techniques to help ensure high assurance, but is not going to disclose those to you. So they cannot be disclosed. So what principle does it violate?

Remember, security wells security is what open design principles says you shouldn't rely on. So open design principle is the one that is not being followed in the system, so that option two here is the correct answer.

- ☐ Complete mediation
- ☑ Open design

Design Principle Quiz

Select the best answer:

A home wireless router comes with a setting that does not encrypt traffic unless security settings are explicitly enabled. This violates...

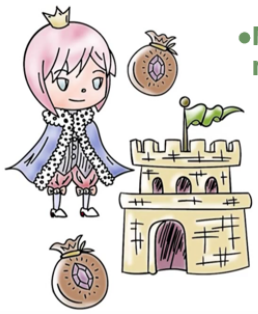
- ☐ Ease of use principle
- ☐ Fail-safe default principle

So if this is how the system comes out of the box, the question here is, is it violating a design principle for secure systems? And if it is, which one that is. So depending on your choice, one or both, you want to check those.

So actually the answer, here, is really talking about default settings. This is default, don't encrypt. And if you want encryption, which is the right thing when you have to mess with the settings? So the defaults are wrong here. That is why I would pick the second answer, which says fail-safe default principle is what is being violated here. Because the defaults, essentially, come such that this system is not secure.

- ☐ Ease of use principle
- ☑ Fail-safe default principle

How Do We Build a TCB: Support Key Security Features



- **Must implement certain security relevant functions**
  - Authentication
  - Access control to files & general objects
  - Mandatory access control (SELinux)
  - Discretionary access control (standard file permissions)

Okay now that we talked about what we want in a trusted computing base, we want it to be trustworthy, we want it to be high assurance and we said even in a follow designed principals. So let's talk a little bit about how do we build a trusted computing base.

Well first of all, if we claim it to be a trusted computing base, it has to implement certain key requirements or features that you associate with a secure system. In other words, it must implement certain security relevant functions. Certain functions that are important for securing access to resources.

So, for example, it has to address authentication. If it doesn't address authentication we will know who's making a request for a resource that is protected in the system. It has to do access control. Access control for whatever resources that we have, could be files, could be other kind of objects that we have. When it comes to access control, we talked about two different kinds of access control. We talked about mandatory access control. It may support that. It may support labels that can be used to implement the BLP model for example or some of the other mandatory access control models that we discussed. It has to support discretionary access control. So somebody who creates a resource that has some control over who can access it and who cannot access it.

The basic idea here is that when you say I'm going to build a trusted computing base, the first question you have to think about is, what are the core functions I have to provide? Some core mechanisms that have to go into the system and these are some examples of those.

How Do We Build a TCB: Support Key Security Features

- **Protection of data used by OS** (OS must protect itself)
  - Security features of trusted OSes
    - Object reuse protection
    - Disk blocks, memory frames reused
    - Process can allocate disk or memory, then look to see what's left behind
    - Trusted OS should zero out objects before reuse
    - **Secure file deletion**: overwrite with varying patterns of zeros & ones
    - **Secure disk destruction**: degaussing, physical destruction

So let's dig a little deeper into sort of how we're going to implement some of those functions, or those features.

First of all, we have to protect the data that is used by the operating system, or the transferred computing base. It must protect itself. Remember, one of the requirements we had is that it should be tamper proof. So, no one should be able to tamper with it. And that's only possible when it is able to protect itself, against that kind of tampering. So let's talk about some security features of the trusted operating systems that you have, in terms of protecting data and other kind of resources.

So one thing that we address in these kinds of systems, is what is called object reuse protection. Think of various kinds of objects or resources, a memory pages, for example. Disk blocks. And so what happens when you allocate a memory page to one process. That process completes and then goes away, and then that page is allocated to a different process. But what happens if that page had some sensitive data that belonged to the first process? This also applies to disk blocks for persistent data. When a file is deleted, we may free up certain blocks where the file data was stored, and these free blocks are free to be assigned to somebody else. And they could be, but there may be data already in these blocks. In fact, there is.

The one thing we can do, if we are concerned about having trust in the system, is that when a process is allocating either disk blocks or memory, then we should be aware of the fact that something could be left behind in these reusable resources.

Well, what can we do if something is left behind? If this the system is trusted, it can zero out these object before they get reused. So before the memory block is given to the new process, we make sure that it doesn't haven access to the data that was stored in that memory block or page by the last process. We can zero it out.

If we're talking about a file deletion, or data that's written out in the disk, we can override it with various patterns of zeros and ones. And we have secure deletion, for example. And that's basically writing it one or more times with some garbage, random zeroes and ones, to make sure that all data that was there is no longer there.

And you can sort of carry it even further for example, when you discard your disk, do you actually destroy it, do you degauss it, or you can do physical destruction. Because one of the things reported a number of years ago, a bunch of students at MIT actually went to some place, and bought some old PCs at almost no cost, and then pulled out the discs in those machines, and they contained all kind of sensitive data. Because even if you, either people don't delete files, even if you delete, if you're not doing some of these things that we discussed here, which a trusted system should do, the data would still be there. And if the disk gets in the hands of somebody else, then they will be able to sort of gain access to the data.

So if you're building a secure system, you have to worry about this idea of reuse of objects or resources. In particular, memory objects, disk blocks, and how information that's stored in them by a given process of the system has to be wiped clean, before those resources are re-allocated to somebody else.

How Do We Build a TCB: Support Key Security Features

- **Complete mediation of accesses**
- Trusted path from user to secure system
  - Prevents programs from spoofing interface of secure components
  - Prevents programs from tapping path (e.g. keyloggers)
- **Audit log showing object accesses** – only useful if you /look/ at the log
  - Detect unusual use of the system

So let's talk about some of the key features that we have to implement, or some of the other functionality that the Trusted Computing Base must have.

We have to have a complete mediation. We have to guarantee that there's no way to get to a resource without going through the trusted computing base, okay, or the reference monitor that it implements.

We have to have a trusted path. Remember, trusted path is needed because otherwise what could happen?

Well, some malicious program, a Trojan for example, can spoof the interface of your trusted system and ultimate trusted system is your trusted computing base. So it could look like the operating system and you may think that the log in is being, you're interacting with the OS. A trusted path allows you to guarantee that when you think you're talking to the trusted computing base or the operating system, you indeed are talking to that system.

Which, of course, prevents programs, malicious programs in particular, keyloggers. A quick example of they're trying to log your key strokes or keys that you press to figure out your password or whatever it is. So, it prevents these kind of programs from tapping the path from keyboard where you press your keys to the system where the key presses are captured. This path has to be trusted, and a result of that trust is that malicious programs are not going to be able to gain access to it and be able to tap and see the information that is being sent over it.

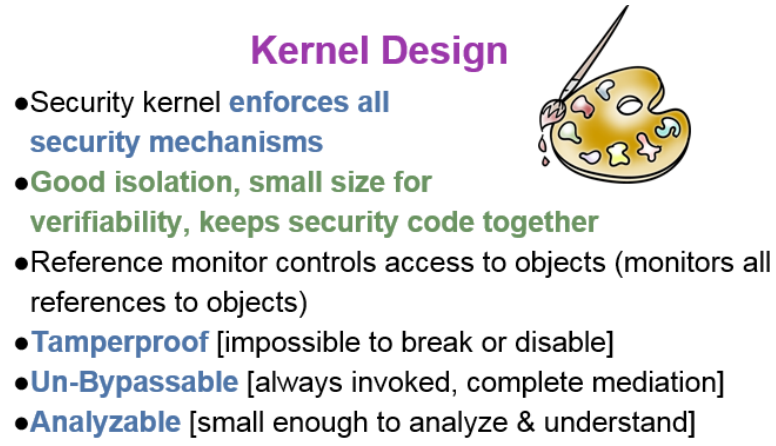
The next thing we have, in terms of functionality features, we didn't talk about that. A lot of our focus has been on the prevention side, the idea that, you know, authentication makes sure it's the right person. Access control says the right person gets access to a resource that is getting requested. The thing is that, no matter how hard we work at prevention. Things could go wrong. Somebody can steal your password, for example. And they can login as you. If we keep an audit log of what has happened, what objects have been accessed, when, by whom. Well this sort of an audit log can come in handy when, and some abuse or misuse might actually take place. So, prevention, if it's not 100%, then you have to go to detection, and having a log will help you with that detection.

Typically, detection is that for some definition of unusual, you see some unusual activity in the system that potentially means an attacker has gained access to it and that unusual use can be, an analysis that you do, that detection is possible when you have a recorder log of what the system did. Keep in mind that this will only be useful if we actually look at the log and you're able to correctly assess when there is an abuse. So although audit, we're not going to talk a great deal about that here, but Lamson, one of the

pioneers who worked in security, actually had this thing called the gold standard of security. This was based on the three AUs:

- Authentication,
- Authorization, and
- Audit.

Periodic table symbol for gold is AU. So the three AUs actually define the gold standard for security. And the third AU typically is audit function that system. So I said it's useful only if you look at it, but also it's only useful if malicious programs are not able to tamper with it. Okay, so it's integrity obviously is going to be important.



So we've been talking the trusted computing base, what it has to do. The fact that it has to be small, but it has to be sort of the kernel or be at the center of everything, isn't it. We talked about, it should do complete mediation. You have to go through it to get to physical hardware resources. Or, we shouldn't be able to bypass it. So that means this is something you always go through and this is at the hardware core of any system that we have.

So, the Kernel is that core, the inner-most. The most trusted, hopefully something that you can trust for things to be right. And this kernel is where the security mechanisms have to be enforced. So the security kernel is sort of at the center or the core. It has to enforce anything to do with security. The fact that all it's at the center and small.

It's just a kernel, and it's not everything, an entire blob, hopefully that helps with good isolation. Small size is good for verifiability so kernel design basically says look for the smallest possible system where you can put your security mechanisms.

The reference monitor that controls access to objects where all references are monitored. Enforcement all security mechanisms access to protected objects or resources, one of those. The reference monitor function of course has to be one that the security kernel has to address and the security kernel sends good isolation.

Actually, it becomes the core of the trusted computing base or the trusted computing base, so it has to be tamper proof.

You shouldn't be able to bypass it, always gets in the way, it's always invoked. That's our complete mediation function or reference monitor that has to check every reference.

And going back to the small size we had before, we want, the small size is going to help us be able to analyze it and understand it and make sure that it does enforce or implement the security mechanisms that we need, and those implementations are correct.

So the idea here is that trusted computing base, one approach is the security kernel approach where you bring the security relevant functions together and that goes at the core of the system that you're trying to build.

Kernel Design

What is included in the trusted computing base (TCB)?

- **All parts of OS needed** for correct enforcement of security policy
  - Handles primitive I/O, clocks, interrupt handling, hardware capabilities, label checking
- **Virtualization**
  - Virtual machine provides hardware isolation, logical OS separation

So you can say, well, it's hardware kernel design at the heart or the center or core of it. What really goes into it? What is included in the trusted computing base when it is done by using the security kernel kind of an approach.

So we said, well, last one we're saying everything that is necessary for the mechanisms that are necessary for correct enforcement of security policies we want to have in the system should go there.

We know resources are implemented using physical resources we have, so any I/O that's necessary to access your physical resources, primitive I/O has to be handled in there, which includes clocks, interrupt handling, at least capturing the interrupt, and being able to directly access the hardware resources with whatever capabilities that are necessary for it. In memory, for example, we said we have TLBs or speeding up the address translation process and things like that. So who can manipulate those control part of memory, somebody can use. So that kind of stuff, if you're doing manager access control, then there's label checking, that the model is going to rely on notion of labels and making sure secret information doesn't flow into an object that is unclassified and things like that. So that's part of your security policy and the correct enforcement, of course, has to be implemented by the security kernel. And direct access to hardware resources and their manipulation, and support for these various models for access control we have has to be in that.

The other thing we did is sort of virtualization. One way we get sort of a complete mediation, and you can bypass it, is that untrusted code is, and has handles who worked with resources. Unless we're talking about capability systems. So this virtualization, were putting this mechanism going from virtual to physical resources helps us isolate the hardware from untrusted code. It also allows us to do the separation.

Revisiting Assurance

Assurance: Ways of convincing ourselves that a model, design, & implementation are correct

Methods of assurance validation:

- Testing /Penetration testing
- Formal verification Validation
- Checking that developers have implemented all requirements
- Requirements checking, design & code reviews, system testing

Now we talked about what the trusted computing base has to do. Whether we take the security kernel approach or we don't but we largely sort of focused on what it has to do. When we say revisiting assurance, what we're really talking about is how well does it do that?

As the example we used before, somebody's trying to sell you an operating system, which is as we said is the defector trusted computing base. And they're making all kinds of claims about how trusted it is. When we say we want to get a sense of the level of assurance, somebody has to sort of convince us. So let's just sort of quickly talk about a couple of sort of ways in which assurance can be provided.

One thing they can do is saying you know we did implement it is a software system but we extensively, thoroughly tested it. So testing is normal testing. You do it for right functionality, things like that. But there's also pen testing or penetration testing. Penetration testing basically refers to sort of you come with the adversarial mindset, saying what vulnerabilities can I discover in this system?

If you can, better is to actually have a proof that cannot be refuted. So you have a formal verification. It's a theorem that we prove saying this is what my system is supposed to do and here you can check that it does exactly that and nothing else.

Other things we can do is, you know, some sort of a checking that developers have actually implemented all the requirements that we had. And we talked about a bunch of different functions that are trusted computing base or have requirements for a trusted computing base.

Revisiting Assurance

Testing:

- Demonstrate existence of problem
- Cannot demonstrate absence of problem
- **Regression testing**: ensure that alterations do not break existing functionality / performance (regression: “going backwards”)

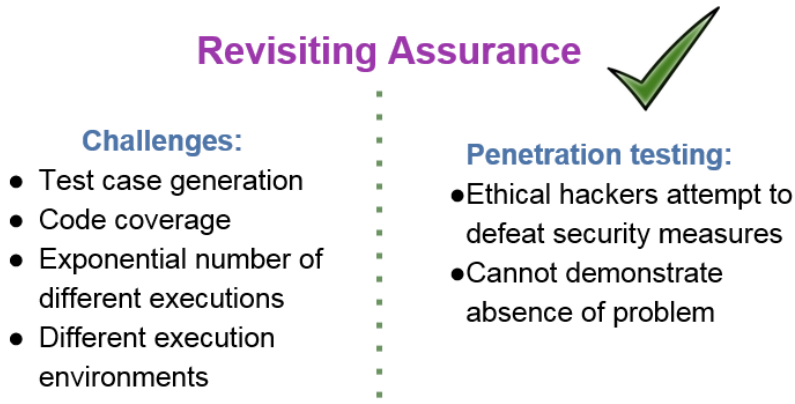
So, when it comes to assurance and it comes to testing, a word of caution here.

We should look into what exactly does testing give us in terms of assurance. So remember, testing is good for showing that there is a problem.

Testing obviously cannot demonstrate absence of a problem. You can't say, well I ran all these tests, so there are no problems. Tests can only demonstrate problems that you can find.

And the variety of testing, the testing is something that we do fairly frequently in the software development process, in the QA process, Quality Assessment process. And one kind of testing, for example, is regression testing that says when we enhance a system, add features to it, patch it, well it's a new system, it's a different system. So regression testing says, you run the test that we have to make

sure that these alterations that we did or these changes that you made don't break functionality or it could improve performance. However, the key take away from here is really the first two bullets which is testing is important. It's useful to discover problems. If you've done a lot of testing, it increases your level of assurance or confidence, but of course it doesn't guarantee that there are no more problems that may be left with the system.



So when talking about the level of assurance that we have in the system and we said, well, we can do testing. You can do formal verification. So let's talk about why that is harder, some of the challenges that we may face when we're trying to do that.

Well, when you're doing tests, you have to then rate the right set of tests to your test cases. Test cases are supposed to show the existence of a problem.

If you don't have a good set of these test cases, then a problem that you may have may go undiscovered. In other words you want to have good code coverage. You won't execute the code that we have in the software system and the kind of execution paths we may be able to take when the system is actually deployed.

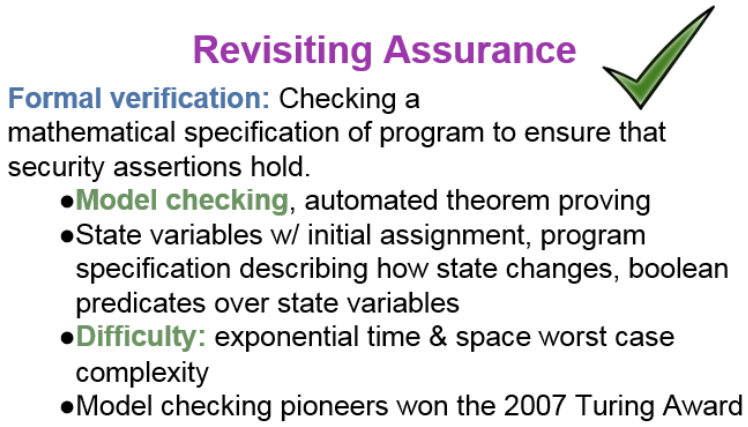
The problem with this is that there could be a very large impact on exponential number of different executions that are possible.

And these different execution paths we're talking about could change when we have different execution environments.

So we talked about the other kind of testing, which is the penetration testing, okay. So that is the adversary actually or somebody else trying to find one way. Here when we're doing testing as a good guide, we have to make sure that every possible execution path we know that is good. So you have the burden of showing that no matter how the system executes, things will not go wrong. If you're doing Pen testing, the problem is different. You're not concerned about every possible execution path. You're actually concerned about some execution path where a problem shows up, some vulnerability you discover that can be exploited.

One way we do that is if ethical, hopefully, hackers, they attempt to defeat the security measures that we have in place. If they're successful that means that we have a problem with the security measures that we use in our system, okay? If they can't find any, that doesn't mean that somebody else can't find them two weeks from today, but if they can defeat the security measures, then we know that we have a problem.

Keep in mind that, even with pen testing, as in testing, we only talking about showing the existence of a problem that is there. That a problem exists. We can never demonstrate the absence of a problem.



And that is formal verification. Formal verification is essentially the mathematical specification of what the program is supposed to do, and what security assertions have to hold, what security properties have to hold. And it's a proof, and the proof is about the correct behavior of the program or any executions that the program may go through.

So, how can we do this? Well, there are automated theorem proving techniques, things like model checking and all.

What they typically do is, at our program or code that we have, we'll have a bunch of state variables with some initial assignment or values. And program instructions or statements or the code that we have actually change these state variables. Depending on what the instruction or the statement is, it's maybe assigning new variables or altering, or computing and things like that, whatever it is doing. But the state changes and then we have some boolean predicates. The correctness is sort of captured in predicates that hold on the state variables that we have. And so essentially, model checking or theorem proving, basically is saying that certain security assertion we have is going to hold as the program executes or as the state variables change, due to the execution of the program. The assertions remain true. Okay, if you can show that for everything that the program does of course, all of its statements, starting in some initial state. That is a proof of correctness, or formal verification that whatever that we wanted actually holds.

Well the problem is that model checking, the same sort of difficulty we had with testing is that the state space can actually explode. And worst case complexity is again, exponential.

However, there are strides being made in this field, Verification. The problem that we have with it is would it scale to really large systems. And your operating system, of course, is one of those large systems. It doesn't quite scale to that size system, but as I said, strides are being made. We can check where these kind of properties, and this kind of work actually. So the model checking pioneers won the highest award that computer science has, the Turing Award, in particular in 2007. Because of the importance or impact this kind of work could have in, not just building or writing software that implements the functionality we desire, but also implements the security requirements that we want to have or security properties that we want to have in the system.

Reducing TCB Size Quiz

Check all applicable answers:

We discussed the need for reducing the size of the TCB. This helps with...

- ☐ Testing of the TCB
- ☐ Verification of the TCB
- ☐ Isolation of the TCB

We discussed the need for reducing the size of the trusted computing base. If we do that, we have couple of different options here, so you should check each one that you think smaller size TCB would help with. Okay? So, for example, tier testing or verification, if smaller size helps with each of these, you check those options.

- ☑ Testing of the TCB
- ☑ Verification of the TCB
- ☑ Isolation of the TCB

So small size or reducing the size of the trusted computing base, will it help the testing? Yes. Remember, testing, you need to have code coverage. Larger the code you have, harder it is going to be for you to cover all these possible execution paths. So testing would be easier. Verification we talked about state space and explosional state space and exponential. Well, size is small, that's going to be easier, too. Finally, isolation of the TCB. Depending on how isolation is being done, in general if it's smaller, fewer interactions with other components and things like that. So small size is actually good for everything. So I'm going to go ahead. The details as they say, the devil is in the details. So you could say well what exactly how, what isolation you talking about. But in general, we say reducing the size is a good thing.

Testing TCB Quiz

Check the correct answer:

Testing is challenging for a complex system like a TCB because of...

- ☐ It is hard to cover all executions
- ☐ It can show both existence and absence of problems

The next question is, what makes testing challenging for a system like a trusted computing base. So there are two options and, you look at each, and, see if it makes sense.

- ☑ It is hard to cover all executions
- ☐ It can show both existence and absence of problems

The first option actually is the only one that is correct. As the system goes in complexity, it's going to be difficult to cover all executions because in testing, you want to cover as many possible executions as you can. So, size, complexity grows, this is going to be more difficult.
Testing can actually never show the absence of a problem. So that it can not do. So, we're not going to pick the second option here. It can show the existence of problems, but not the absence of problems. So, this and that we have here makes the second option an incorrect option.

Model Checking Quiz

A key problem with model checking is...

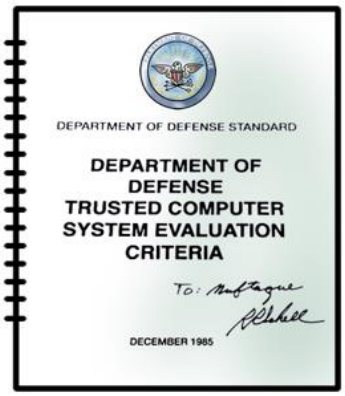
What is the key problem with model checking is what the question is asking. Two options, consider each, and check the one you think explains the problem with model checking.

- ☐ It cannot show absence of a problem
- ☐ It does not scale to practical large size systems

Model checking actually is correctness proof. If it is correct, that means there is no problem. Actually, it cannot show absence of a problem. It's true for testing but it is not true for formal verification. If you're able to verify, you are able to show that there is no problem. It can show the absence of a problem. If we can verify means there is no problem, and that means this is demonstrating or showing the absence of a problem. So the first option is not correct. It would have been correct if there was testing, but we're talking about verification. See the problem really is the second option, which is that because of the exponential nature of the work that you have to do, it doesn't scale to practical large size systems.

- ☐ It cannot show absence of a problem
- ☑ It does not scale to practical large size systems

Security Evaluations



Lot of people are faced with this problem of how do you evaluate the level of security that comes with the system. No one more concerned about this then, for example, the Defense Department. They have a lot of sensitive data. So they actually address this problem of how do you do security evaluation? How do you establish how trustworthy or level of assurance, level of trust that you can associate with a given system?

And they actually came up with a trusted computer system evaluation criteria. This is work that goes back to actually the 80s. And this is called the Orange Book or TCSEC Requirements. Trusted Computer System Evaluation Criteria. What you see here is, actually it doesn't the color. Actually it is orange. Copy the color page of that report. Now one of the people who were involved in designing that, Roger Shell, who signed it, my copy. So I just provided you a picture of that.

So the question is if you are sort of tasked with this, develop a sort of violation criteria that we can use to assess the trustworthiness or security of the system. What would be those criteria? I think some of the discussions we have had for example about functionality and how well it's done and so on, should actually factor into these requirements that we're going to have.

Government Security Evaluations

- U.S. Orange Book (late 1970's)
- D < C1 < C2 < B1 < B2 < B3 < A1
  - D: no protection
  - C: discretionary protection
  - B: mandatory protection
  - A: Verified protection
- C1, C2, B1: security features common to commercial OSes
- B2: Proof of security of underlying model, narrative spec of TCB
- B3, A1: Formal design & proof of TCB

So the Orange Book that I mentioned sort of captured the evaluation criteria, and came up with sort of way to place a system in a division, and within a division, a certain class.

So, the four divisions were D, C, B and A. The less than really refers to the least trustworthy and we'll see why D was the least trustworthy. And A was the highest level of trust. And within that, for example, the division C had C1 and C2, so C1 doesn't do certain things that C2 does for it to be more trustworthy, okay? So this ordering that we have is going from least trustworthy to the highest level of trust. B1, B2, B3 means three classes within the duration B.

Well, D really didn't have to do anything. If you didn't do, think about your old MS Dos, if you didn't isolate the trusted computing base from untrusted code, basically untrusted code could hook operating system functions in whatever way. There's no protection against tampering, bypassing, or things like that.

C, one of the things it has to do is do authentication and access control. Okay?. Access control. It only has to do discretionary access control but of course to do access control it also has to do authentication and if it does that then it addresses a bunch of other things. Just sort of giving you quick highlights here but of course in a TC isolation and stuff like that has to be here too. So that places you in division C.

If you implemented mandatory access control. For example, BLP like model where that will move you up to division B.

And you can verify, use formal verification techniques. Specification in verification. If you went that far, then that's what will be needed for you to get into division A.

Most systems that we use that the security features that they provide, commercial systems that are widely deployed, they're basically in the C division and C1, C2, or whether they move up to get B1, of course, they have to do mandatory axis control. So we said SELinux, for example, does implement MAC. So C1 and C2 actually depends on what functionality. So here it, shouldn't be confused by the fact that this we're talking about functionality with discretionary and mandatory access control. This is actually how well effort you put into making sure there are no problems.

So it's how well you do, you testing, testing, based on all these auditing support and audit logs and things like that. You will have different classes, but the main idea here is that most systems that we use widely these days, they basically division C some implement Mac features and they get into this division B1 that I have.

If you want to move up to B2 for example, what you may have to do is, you have to, whatever security model that your system is, underlies your system, you have to have a proof of correctness for that model. Or you also have to have a spec for the TCB. It doesn't have to formally done, but you have to have a narrative specification of the TCB and what it does.

Of course if you want to move up to B3 or A1 beside that's formal verification that you have to do an, that the TCB is implemented correctly and things like that.

Government Security Evaluations

Common Criteria (2005) international standard replaced orange book

- Originated out of European, Canadian, and US standards
- **Idea:** users specify system needs, vendors implement solution and make claims about security properties, evaluators determine whether vendors actually met claims
- **Evaluation assurance level** (EAL) rates systems
  - EAL1 most basic, EAL7 most rigorous

Well it started as a US DOD effort, at or the orange book, then with time multiple countries sort of recognized this importance of software procurement, they had to worry about security that they can evaluate for those software products. That evolved into what's called the Common Criteria which is an international standard. Common Criteria World came from efforts in Europe, in Canada, and the US one that we talked about.

The idea Is that you are consumer of the system or user of the system, you're going to specify what needs you have or what requirements you have. Vendor going to sort of implement a solution and going to make claims about that solution meeting your needs or your requirements. And in particular, we're talking about sort of security requirements or security properties. So whoever is going to evaluate is going to determine whether the vendor actually, the claims that they make are claims that you can believe. And these claims actually allow you to say, well, my needs that I had specified are met.

So there's three entities, you have some requirements, a vendor who provides the solution makes a certain set of claims how they meet those requirements, and somebody who evaluates has to then assess those claims, and the reason we should be able to trust whatever those claims are about. Based on that, we're going to give it an assurance level, which rates the trustworthiness of the systems.

Evaluation assurance level or EAL, we will be able to use that to rate a system, in particular, how well it meets the claims about security properties that are important to us. So similar to division D, we said, basically, it didn't do very much EAL1 is the most basic one and EAL7 is the most rigorous or the highest level of trustworthiness that you can get.

TCSEC Divisions Quiz

Many widely used operating systems **do not support MAC** and hence cannot be in a TCSEC division higher than...

- ☐ D
- ☐ C

Many widely used operating systems don't support mandatory access control, like BLP models. And hence, cannot be in a TCSEC division higher than what? Okay, so the two options here, D and C. So you're saying, if you don't implement MAC, I know you can't be higher than either D or C. Well, if you can't be higher than D, then you can't be higher than C either. See which option here makes sense in terms of, if you didn't do something, what division could you go into?

So in TCSEC we know that D was the basic one. Really had to do nothing. For C you had to implement some security functions, including discretionary access control, but you didn't have to do mandatory access control. For mandatory access control, if you did, you could go to B. If you don't implement mandatory access control, you can't be higher than C. The best you can do is C. So if you don't support MAC, you cannot be in a TCSEC division higher than C. This is the answer. Automatically also means you can't be higher than D, so this is somewhat poorly formulated question, but you could be higher than D actually. You could be in C, even if you don't implement MAC, which is higher than D. So the only option that you should be checking here is the second one, which is C.

☐ D

☑ C

Earning an EAL4 Certification Quiz

How did **VMware vCloud Networking and Security v5.5** system receive an EAL4+ certification?

☐ The system developers used formal techniques in its design and testing

☐ A systematic review and testing process was used by the system developers

The question here is, what did this vendor do, VMware in particular, that enabled it to get EAL4 but not higher? The two options sort of explore that, and you pick the one that you think is the right one.

So we know that formal techniques and design and validation and all that, that moves you up the scale. In fact, EAL7 is the formally verified most rigorous requirement, so since they only got EAL4, it is more the systematic review and testing process. It doesn't have to be formal techniques, okay? Systematic review and testing process was used by the developers, and that's why they ended up at EAL4, which is not bad. So the two takeaways are what you do and how well you do it.

☐ The system developers used formal techniques in its design and testing

☑ A systematic review and testing process was used by the system developers

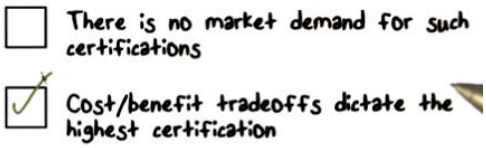
Cost-Benefit Certification Tradeoffs Quiz

Many OS vendors **do not aim for the highest certifications** because...

☐ There is no market demand for such certifications

☐ Cost/benefit tradeoffs dictate the highest certification

Many OS vendors don't aim for the highest certifications. AL7 or A1FDC sec is what you are using. So if they had the highest they would have to formally verify design correctness. They would have to implement mandatory access control, all that stuff. So they're not doing it because of what reason that two options capture those, and you check the ones you think makes sense.



The first one is, there is no market demand for such certification. Well, security has never been more important than it is. So if you can sell someone, you can convince somebody that you're selling more security or higher level of trustworthiness, there will be demand for it. So I don't think the first option is the right one. The second one is cost/benefit tradeoff is really the reason. As you move up the certification chain, of course, is you are to do more and more things and you have to do them in a better way. And there's a cost for doing more and doing extensive testing and verification and whatever that is required and that cost. And so that the benefit that comes from that cost, those tradeoffs are the reason that people don't choose to go for the highest possible certification.

Mandatory Access Control
Lesson Summary

- Provides enterprises **an ability to control how sharing** of sensitive information can be controlled
- Can address both **confidentiality and integrity** but require added functionality with labels
- **High level of assurance** for trusted systems is challenging

So we explored a number of mandatory access control models, ranging from ones that address confidentiality and integrity to those that are designed to meet the needs of commercial applications. We then revisited the idea of a trusted computing base, or a TCB, and how we can evaluate the level of assurance that is provided by the TCB. We saw that it not only depends on what the TCB does, or its functionality, but also how well it has been tested and verified to ensure that it is free of vulnerabilities.