Authorisation

Welcome back. In this video, we're going to introduce the concept of authorization. So far in our access control, we've looked at identification and authentication, so claiming an identity, verifying that you are that person. But we haven't looked at authorization.

Authorization is the element that basically says, OK, you have access to the system, but are you permitted to complete this particular task? This could be access to particular files, or it could be access to a particular function within a system. Access control can be considered as the way of managing the authorization element of this process.

Let's have a look now at a traditional access control model by Lampson. You'll see and the diagram here we have four entities. We have the subject, which is often an end user or even a process. The access request, which is the request itself to gain access to a particular resource, which we refer to as the object.

So again, this could be something like file or a function. And then we have what is referred to as the reference monitor. This is effectively the part of the system which is going to check whether you do have the correct authorization levels for that particular object.

Now, if we think about it at which point within this model we could include the authorization element, we could do it by attaching it to the subject. So for example, Alice is allowed to do this but not to do that. However, an alternative is that we could do it at the object level. People are allowed to do this with the object but not this.

There are, at a basic level, two access modes. Those are observe and alter. However, the set of access modes are often a lot richer. We can have a look at one particular access model if we look at the Linux-based systems. Let's have a look at one of those now. Within this system, we have the operations.

We have read, write, and execute. For any file within the Linux structure, we're going to have owner, group, and world. For each of these, we have the possible options of read, write, and execute. Owner is the user who has created that particular object. Group is a defined group

of users.

And world is anyone who tries to access it. So if an individual user doesn't fall within either the owner or the group, then they would default to the world permissions which are configured. If you wish, for example, the owner to have read, write, and execute access, which is fairly standard, then you would have R, W, and X. If you had the group access, as they can read, and they can write, but they cannot execute, then it would be R, W, and then a dash.

If you didn't want anyone else to have access to the file, then you would not allow read, write, or execute. So it would be dashes for each of those. There are commands where you can play about with this at command line within a Linux-based system. So if you want to do that, feel free to have a go. One of the important principles when it comes to authorization is the principle of least privilege.

This effectively says that you should permit access at the least amount of privilege possible. For example, to do their job, you should not give someone complete access to everything, but you should only give them access to the minimal amount of information that they need in order to execute their job properly.

The reason for this is that if someone were to be given more access than required, that, obviously, increases the chances of information disclosure, or an attacker trying to gain access to that particular account in order to increase their access to particular functions or objects. This is referred to as a privilege escalation attack.

There are two different types of privilege escalation attack. You can have a vertical privilege escalation, which is where the attacker tries to move onto an account which has more access than they have. Or you can have a horizontal escalation attack. In a horizontal escalation attack, they could have access to a similar functionality, but it would be different data. For example, if you have an account for a bank, and you're trying to get access to someone else's account, clearly, that would violate the information security.

But you would still be accessing the same level of functions. With the vertical privilege escalation, you could think of this as someone working with a university, for example, and trying to gain access to a user account for HR, which is going to have more access to the data relating to employee salaries. And these types of attacks are not just academic.

There are examples of these happening in real worlds. They can be delivered through, for example, password guessing attacks or SQL injection attacks as we'll see elsewhere. But that's all for this video where we've had a quick rundown of authorization. I hope you've enjoyed the video. And I'll see you next time.