Process Protection

Capabilities

Root access is a bit all-or-nothing: it allows the root user all access to all (user mode) resources

Process Protection

Capabilities

Root access is a bit all-or-nothing: it allows the root user all access to all (user mode) resources

And this is quite dangerous as it can too easily lead to accidental or malicious damage to the system through misuse

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

This breaks access rights down into small parts

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

This breaks access rights down into small parts

- Rights to access to the network driver

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

This breaks access rights down into small parts

- Rights to access to the network driver
- Rights to access to the sound card

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

This breaks access rights down into small parts

- Rights to access to the network driver
- Rights to access to the sound card
- Rights to access to the filesystem

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

This breaks access rights down into small parts

- Rights to access to the network driver
- Rights to access to the sound card
- Rights to access to the filesystem
- Rights to reboot the computer

Process Protection

Capabilities

A refinement of the root idea is *capabilities*

This breaks access rights down into small parts

- Rights to access to the network driver
- Rights to access to the sound card
- Rights to access to the filesystem
- Rights to reboot the computer
- And so on

This can be broken all the way down to rights to access to individual files, say

Process Protection

Capabilities

We now have

```
if uid_of_process == uid_of_resource or
   process_has_capability(uid_of_process, resource) or
   uid_of_process == uid_of_root
then
    allow access
else
    disallow access
```

Process Protection

Capabilities

These capabilities are like tokens or keys that can be passed around, inherited by processes and so on

Process Protection

Capabilities

These capabilities are like tokens or keys that can be passed around, inherited by processes and so on

Capabilities allow finer control of security at the cost of a more complicated checking system

Process Protection

Capabilities

These capabilities are like tokens or keys that can be passed around, inherited by processes and so on

Capabilities allow finer control of security at the cost of a more complicated checking system

A few OSs, notably Flex, were built around the notion of capabilities and required hardware support to make things work with an acceptable speed

Process Protection

Capabilities

These capabilities are like tokens or keys that can be passed around, inherited by processes and so on

Capabilities allow finer control of security at the cost of a more complicated checking system

A few OSs, notably Flex, were built around the notion of capabilities and required hardware support to make things work with an acceptable speed

These never took off, though

Process Protection

Capabilities

These capabilities are like tokens or keys that can be passed around, inherited by processes and so on

Capabilities allow finer control of security at the cost of a more complicated checking system

A few OSs, notably Flex, were built around the notion of capabilities and required hardware support to make things work with an acceptable speed

These never took off, though

But the idea has come back to modern OSs

Process Protection

Capabilities

In Android access to system resources are protected by capabilities: it calls them *permissions*

Process Protection

Capabilities

In Android access to system resources are protected by capabilities: it calls them *permissions*

At either install time or the first time the application tries to access a resource the OS (not the application) asks the user whether that application should be allowed to access that resource

Process Protection

Capabilities

In Android access to system resources are protected by capabilities: it calls them *permissions*

At either install time or the first time the application tries to access a resource the OS (not the application) asks the user whether that application should be allowed to access that resource

For example, WiFi network access, phone contact list access, SD card access, initiate phone calls, and so on

Process Protection

Capabilities

In Android access to system resources are protected by capabilities: it calls them *permissions*

At either install time or the first time the application tries to access a resource the OS (not the application) asks the user whether that application should be allowed to access that resource

For example, WiFi network access, phone contact list access, SD card access, initiate phone calls, and so on

If so allowed by the user, the application can access those resources *and no others*

Process Protection

Capabilities

In Android access to system resources are protected by capabilities: it calls them *permissions*

At either install time or the first time the application tries to access a resource the OS (not the application) asks the user whether that application should be allowed to access that resource

For example, WiFi network access, phone contact list access, SD card access, initiate phone calls, and so on

If so allowed by the user, the application can access those resources *and no others*

This mechanism would be great if only the user could be trusted to read and understand the list of requests. . .

Process Protection

Capabilities

Summary: user protection is useful and helpful

□

Process Protection

Capabilities

Summary: user protection is useful and helpful

So don’t run things as root/administrator unless absolutely necessary

□

Process Protection

Capabilities

Summary: user protection is useful and helpful

So don’t run things as root/administrator unless absolutely necessary

And don’t confuse it with kernel/user mode

□