Feb 11, 2005 -- Lecture 11

22C:169
Computer Security

Douglas W. Jones

Department of Computer Science

Interdomain Channels

Domains

The domain of a program component is

The objects on which it may operate

In general

Each component may have a domain

Granularity of a system

Size of domains

Fine -- one domain per instruction

Coarse -- only one domain at all

The need-to-know rule

Nothing should be included in a domain

Unless it is needed by the domain user

What things do we want to be in domains?

variables, system calls, files, memory segments, functions, methods, network ports, ...

In sum, all resources of any kind

Mechanism versus Policy

Policy

what ought to be in each domain

Mechanism

how do we enforce this policy

Scope rules: programming language

Access rights: file system

Scope Rule Example

```
int x;

void inc( int * p )
{ *p ++ }

void q() {
   int y;
   inc( &x );
   inc( &y );
   inc( (int *)0xBADBAD );
}
```

Policy questions

Domain size

- *Lots of little domains*
  - Raises development cost
- *One big domain*
  - No internal security

What to disclose, what secrets to keep

- *Assume the worst!*
  - Value of defense in depth

Role of operating systems

Archaic operating systems

provide ad-hoc mechanisms

different mechanism per resource class

Modern operating systems

provide uniform mechanism

independent of resource class

Subject of intense study since 1960s

Almost unknown in marketplace

Interdomain Channels

Overt channels

Those that are intended by design

messages

function calls

Covert channels

Those not intended in system design

covert communications

secret interfaces

Security of Overt Channels

Validation of content, parameter validity

pass a pointer to an object from a to b

object not in domain a

object is in domain b

pass code from a to b

not executable in a

executable in b

b must check safety!

Security from Covert Channels

- Hidden channels, a kind of Trojan
  - As with all Trojans
    - automatic detection - no guarantees
    - system inspection can be mislead
- Unintended channels, accidents
  - Resource usage channels
    - require ingenuity to find or use
    - eliminate shared resources
    - or inject noise

Uses of Covert Channels

Communication between

- *attackers*
- *attackers and malware*
- *components of malware*

Pathway for system attack

- *buffer overflow attack is an example*
  - failure of validity checking
  - created a covert channel!