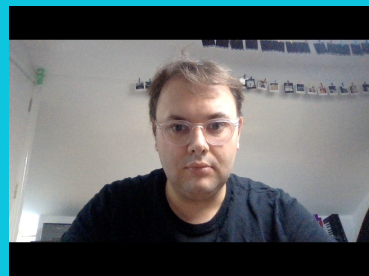
University of BRISTOL

Introduction to OS Security

Joseph Hallett



It all comes back to Saltzer and Schroeder…

related to hazard from lasers and other light sources," *Amer. J. Ophthalmol.*, vol. 66, p. 15, 1968.
[57] A. Vassiliadis, H. C. Zweng, N. A. Peppers, R. R. Peabody, and R. C. Honey, "Thresholds of laser eye hazards," *Arch. Environ. Health*, vol. 20, p. 161, 1970.
[58] P. W. Lappin, "Ocular damage thresholds for the helium-neon laser," *Arch. Environ. Health*, vol. 20, p. 177, 1970.
[59] W. T. Ham *et al.*, "Retinal burn thresholds for the He-Ne laser in the rhesus monkey," *Arch. Ophthalmol.*, to be published.
[60] T. P. Davis and W. J. Mautner, "Helium-neon laser effects on the eye," U.S. Army Med. Res. Develop. Com., Washington, D.C., Annu. Rep. Contr. DADA 17-69-C-9013, 1969.
[61] J. J. Vos, "Digital computations of temperature in retinal burn problems," Inst. Perception, Soesterberg, The Netherlands, RVO-TNO, Rep. IZF 1965016, 1963.
[62] M. A. Mainster, T. J. White, J. H. Tips, and P. W. Wilson, "Retinal-temperature increases produced by intense light sources," *J. Opt. Soc. Amer.*, vol. 60, p. 264, 1970.
[63] A. M. Clarke, W. T. Ham, W. J. Geeraets, R. C. Williams, and H. A. Mueller, "Laser effects on the eye," *Arch. Environ. Health*, vol. 18, p. 424, 1969.
[64] R. H. Stern and R. F. Sognnaes, "Laser beam on dental hard tissues," *J. Dent. Res.*, vol. 43, p. 873, 1964.
[65] R. H. Stern, "Dentistry and the laser," in *Laser Applications in Medicine and Biology*, vol. II, Dr. M. L. Wolbarsht, Ed. New York: Plenum, 1974, pp. 361–388.
[66] T. E. Gordon, Jr., and D. L. Smith, "Laser welding of prostheses—an initial report," *J. Prosth. Dent.*, vol. 24, p. 472, 1970.

The Protection of Information in Computer Systems

JEROME H. SALTZER, SENIOR MEMBER, IEEE, AND MICHAEL D. SCHROEDER, MEMBER, IEEE

Invited Paper

***Abstract*—This tutorial paper explores the mechanics of protecting computer-stored information from unauthorized use or modification. It concentrates on those architectural structures—whether hardware or software—that are necessary to support information protection. The paper develops in three main sections. Section I describes desired functions, design principles, and examples of elementary protection and authentication mechanisms. Any reader familiar with computers should find the first section to be reasonably accessible. Section II requires some familiarity with descriptor-based computer architecture. It examines in depth the principles of modern protection architectures and the relation between capability systems and access control list systems, and ends with a brief analysis of protected subsystems and protected objects. The reader who is dismayed by either the prerequisites or the level of detail in the second section may wish to skip to Section III, which reviews the state of the art and current research projects and provides suggestions for further reading.**

GLOSSARY

THE FOLLOWING glossary provides, for reference, brief definitions for several terms as used in this paper in the context of protecting information in computers.

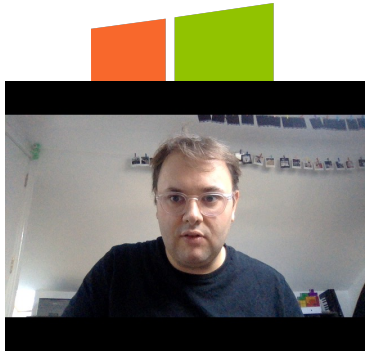
Access	The ability to make use of information stored in a computer system. Used frequently as a verb, to the horror of grammarians.
Access control list	A list of principals that are authorized to have access to some object.
Authenticate	To verify the identity of a person (or other agent external to the protection system) making a request.
Authorize	To grant a principal access to certain information.
Capability	In a computer system, an unforgeable ticket, which when presented can be taken as incontestable proof that the presenter is authorized to have access to the object named in the ticket.
Certify	To check the accuracy, correctness, and completeness of a security or protection mechanism.
Complete isolation	A protection system that separates principals into compartments between which no flow of information or control is possible.
Confinement	Allowing a borrowed program to have access to data, while ensuring that the program cannot release the information.
Descriptor	A protected value which is (or leads to) the physical address of some protected
Discret	
Domai	
Enciph	
Grant	
Hierar	

Manuscript received October 11, 1974; revised April 17, 1975.
The authors are with Project MAC and the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass. 02139.

What is an OS?

- Operating system provides an abstraction over the computer's hardware
- Bigger OSs have to run more than one program with more than one user…
- …We'd also like it to implement some security policies

- (for example access control)



Access Control Security Goals

Confidentiality: you can't see what you don't need to see
Integrity: you can't tamper with stuff that isn't yours
Availability: you can get at your stuff

All theses goals are interdependent:

If I can tamper with data, who cares if it is confidential…?
If nothing is confidential, then I can be root and do whatever

Principles, Subjects, Objects…

Jargon common to access control

Principal: a "person" describing the access control policy or human trying to follow the policy

Object: the resource (file, process…) that we are writing the policy about

Subject: the things (processes) interacting with the objects that we are trying to restrict

UNIX DAC

Traditional access control mechanism present (in some form or another) in almost all OSs
Objects have an owner (single user) and a group (multiple users)
At the Owner's *discretion* what they, the group, and everyone else can do with the object. (Read, Write, Execute)

```
-rw-r--r--   1 root  wheel    7630  1 Jan  2020 passwd
```


Problems with DAC

Suppose Alice wants to run a web browser…
We'd like that to be able to access her downloads folder...
…but maybe not her SSH keys

Suppose Alice wants to run an SSH server…
We'd like that to be able to access her SSH keys…
…but maybe not her downloads folder

Problems with DAC

DAC policy described at the *object* level…

We could work around it…
Alice's programs runs as an alice-unprivileged user, and use group permissions to set where they can access…

…and then we'd need to duplicate the policy for multiple users…

Gets **really complex** fast...
Becomes very hard to verify

Problems with DAC

As a sysadmin:
Do we trust Alice to get her policy right?

What if she wants to download and run programs she found online?

What if we know that some of her work must be kept confidential?

What if we trust Alice, but don't trust Bob quite as much?

Principle of *least privilege*!

We need a mechanism to be able to enforce a security policy from th
down... not just rely on discretionary controls!

So how are we going to fix it?

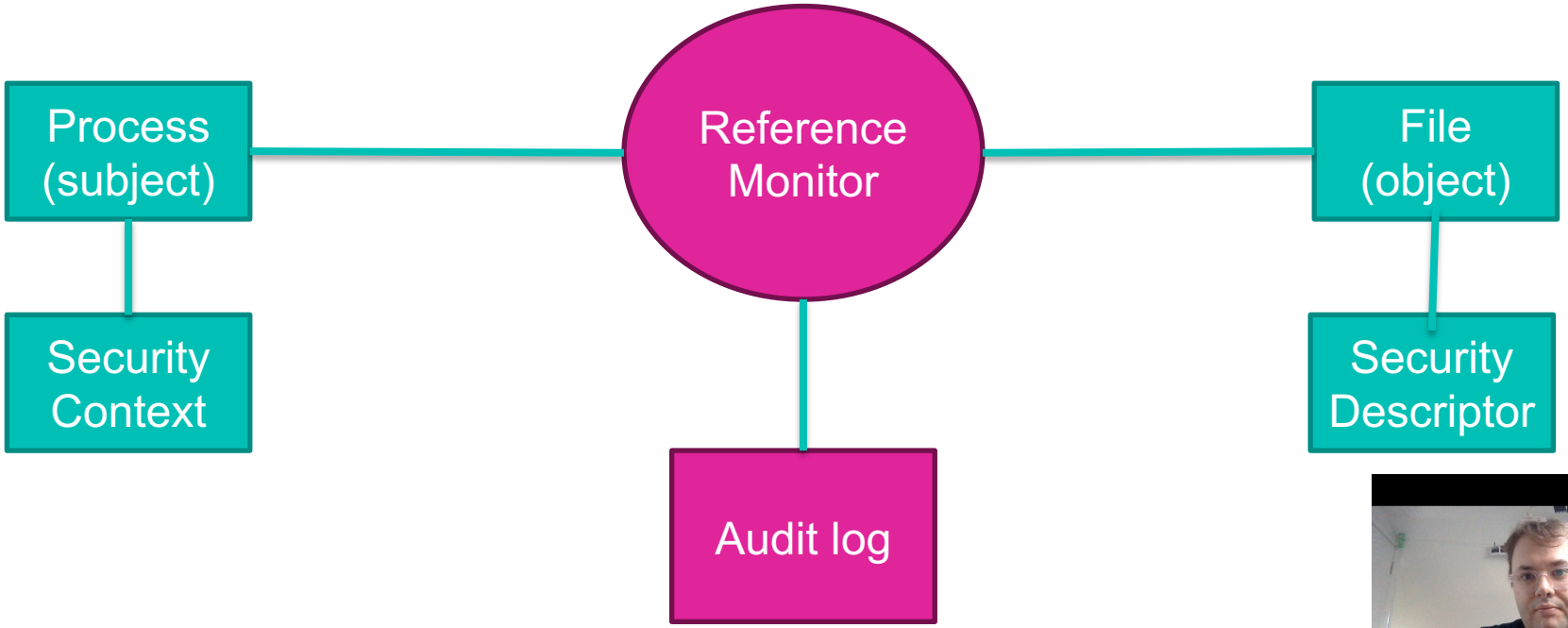
Reference monitors!

Subjects: processes are associated with a security context (user, group, and privileges)
Objects: files have security information (DAC and xattrs)

On login, processes get the capabilities of their principal and then these are progressively dropped
Processes inherit capabilities of the process that made them

Reference monitors

Everything gets mediated by Reference Monitor



Reference monitors

No way for subjects to access objects except through the reference monitor (Complete mediation)

When a subject makes a system call…
- Get information about subject
- Get information about object
- Apply system policy based on that information
- Log that a decision was made
- Return that decision

Be careful about race conditions…

MAC

Mandatory access controls
Sysadmin sets the access control policy (which may be use the DAC)

Simplest form in *Multi Level Security (MLS)*
- Emerged from the US military
- Subjects and Objects associated with a security level

UNCLASSIFIED < CONFIDENTIAL < SECRET < TOP SECRET

MLS models: Bell-LaPadula

Say Alice has a SECRET security clearance

Should Alice be able to *read* a TOP SECRET document?
• Probably not—she might learn a secret above her clearance!

Should Alice be able to *write* to an UNCLASSIFIED document?
• Probably not—she might leak a secret from a higher security clearance!

Bell-LaPadula: No read-up, no write-down… preserves *confidentiality*

MLS models: Biba

Say Alice has a SECRET security clearance

Should Alice be able to *read* an UNCLASSIFIED document?
• Probably not—why would she need to… keep her isolated to her level!

Should Alice be able to *write* to an TOP SECRET document?
• Probably not—she shouldn't be able to influence higher clearances!

Biba: No read-down, no write-up… preserves *integrity*

TSEC

Defines the security level of a system for the US DoD and government

Grade D: Nothing

Grade C: Discretionary controls

- C1: something like the UNIX DAC
- C2: C1 + logging

Grade B: Mandatory protection

- B1: C2 + MAC for named objects
- B2: B1 + formal security policies and full MAC
- B3: B2 + testing, recovery, auditing, incident response

bristol.ac.uk