CS604: Advanced System Security

Multics System

Deepti Vidyarthi
DIAT

(Acknowledgements to Trent Jaeger and William Enck for reference)

Secure System - Evaluation Criteria

- Mediation: Does interface mediate correctly?
- Mediation: On all resources?
- Mediation: Verifably?
- Tamperproof: Is reference monitor protected?
- Tamperproof: Is system TCB protected?
- Verifiable: Is TCB code base correct?
- Verifiable: Does the protection system enforce the system's security goals?

- *Does Multics satisfy these?*

MAC Systems

- Major Effort: *Multics*
- Multiprocessing system - developed many OS concepts (Including security)
- Began in 1965
  - Development continued into the mid-70s
- Used until 2000
  - Initial partners: MIT, Bell Labs, GE/Honeywell

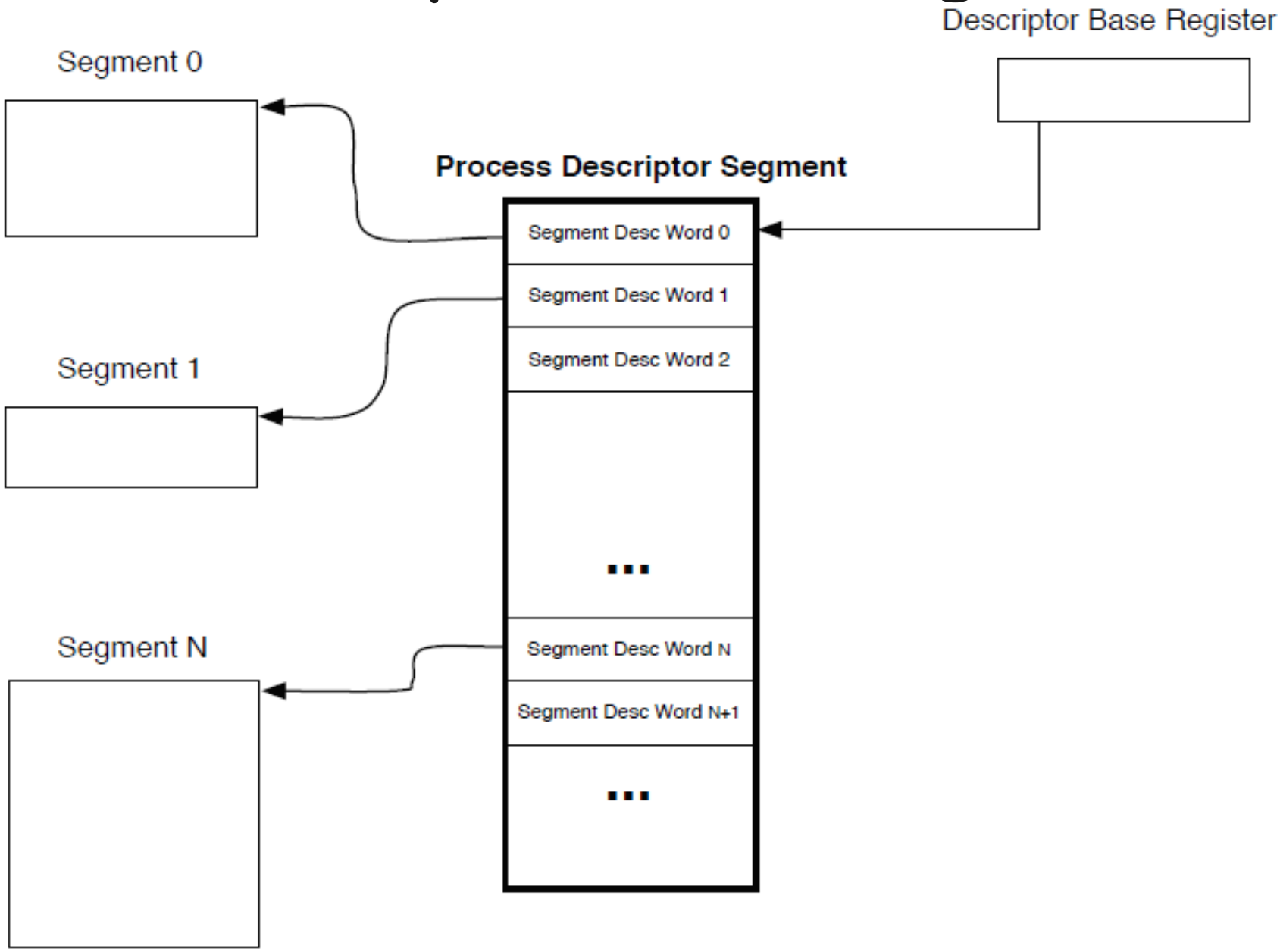
- Subsequent proprietary system, *SCOMP, became the* basis for secure operating systems design

Multics Fundamentals

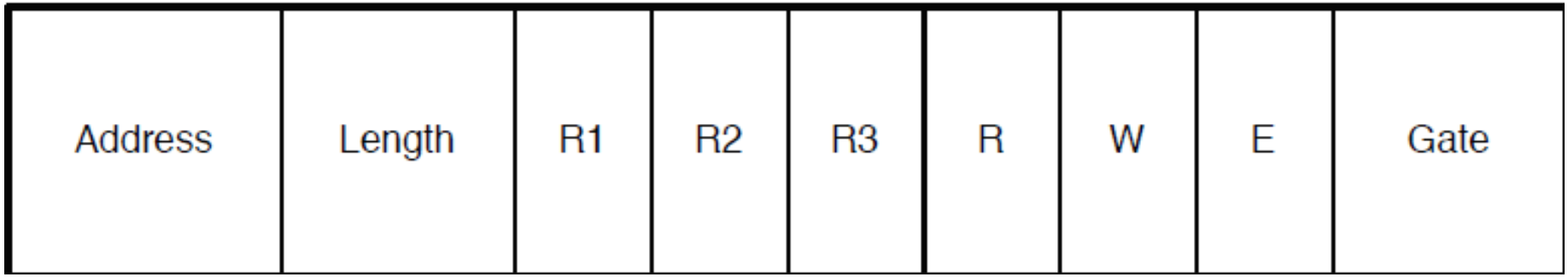
- Processes: (executable context to run code)
  - Protection domain of a process = segments it can access + operations it can perform on those segments.
  - Process Descriptor Segment

- Segments: files, I/O devices, etc..
  - Organized into a hierarchy of directories and segments
  - Segments in the process's context have a set of segment descriptor words (SDWs)
  - Segments outside the process's context must be addressed by full name

Multics process's segment



Multics process's segment



SDW: Segment Descriptor Word

Multics Security Fundamental Concepts

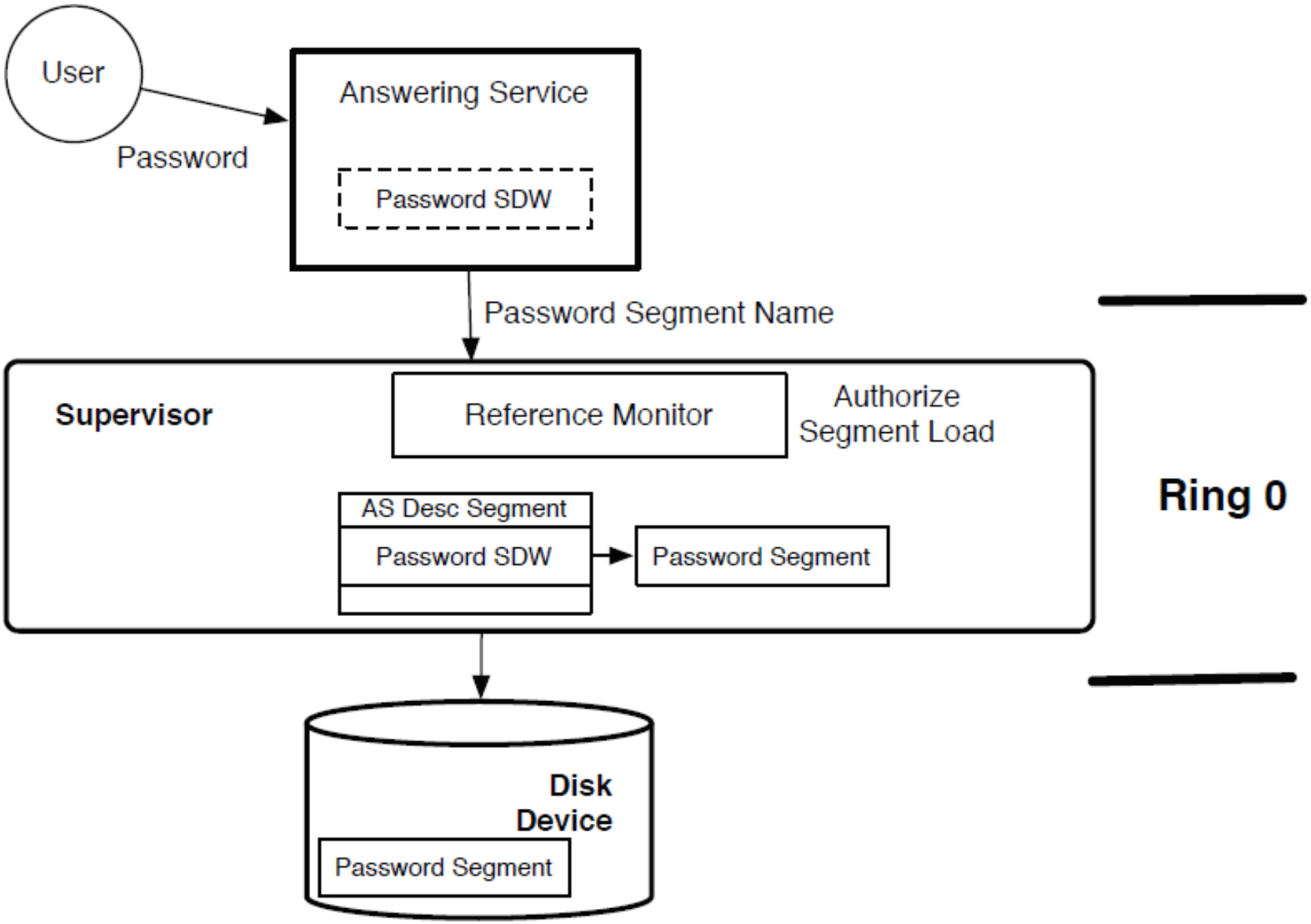
- Supervisor: core component - part of TCB.

- Protection rings: a hierarchical layering.
  - Ring 0: access control, I/O, scheduling, segmentation and memory management, etc.
  - Ring 1: accounting, stream management and file system search, etc.

The login process

- Handled by answering service.
- Answering service retrieves password segment to verify password (authorized by supervisor).
- If user and password match, a process is created with the appropriate code and data segments, and perhaps the appropriate SDW's.

The login process



Multics Protection System Models

- 3 different, interacting protection models :
  - Access control lists
  - Rings and Brackets
  - Multilevel Security

- Access is granted if all three models grant it.

Access Control Lists

- Each segment + each directory has associated an ACL - stored in the parent directory.
- Permissions:
  - r(ead), w(rite), e(xecute) for segments
  - (check)s(tatus of entry), m(odify entry), or a(ppend an entry).
- To change the ACL, m permission is need on the parent directory.

perms	user(s) (can be wildcarded)
rwe	**Backup.SysDaemon.***

Rings and Brackets

- Each process runs in a certain ring r.
- Each segment has three ring numbers associated with it: $r1 \leq r2 \leq r3$.
- Segment access - determined by r, r1 & r2.
- Segment execution - determined by r, r1, r2 & r3.

Rings and Brackets

- Ring Policy for segment access

r < r1	R, W - the process can read and write the segment.
r1 ≤ r ≤ r2	R - the process can only read the segment.
r2 < r	No access allowed

Rings and Brackets

- Ring policy for segment execution

r < r1	• Execution allowed • Ring transition r-> r' • r' , r1 ≤ r' ≤ r2, specified by the segment
r1 ≤ r ≤ r2	• Execution allowed • no ring transition
r2 < r ≤ r3	• Execution allowed if authorized by the gates in code segment's SDW; • Ring transition r-> r'
r3 < r	• Execution not allowed

Multilevel Security

W only	**S/D level ≥ Process level**
R only	**S/D level ≤ Process level**
R + W	**S/D level = Process level** **Process trusted**

Multilevel Security (MLS)

- Subjects & objects - placed in levels such that their levels can be compared.
- The higher the level the "more secret" it is.
- Reading and writing is constrained so that information can flow from less secrecy to more secrecy.
- Read below and Write above.

Multics Protection System Models

- 3 different, interacting protection models :
  - Access control lists
  - Rings and Brackets
  - Multilevel Security

- Access is granted if all three models grant it.

Multics Security

- Secrecy
  Prevent leakage – even if running untrusted code

- Integrity
  Prevent unauthorized modification – layers of trust

- Comprehensive control - enforce at lowest level

Multics Security

- ACLs and ring brackets for a segment may modified by process that has the modify privilege to the segment's parent directory.

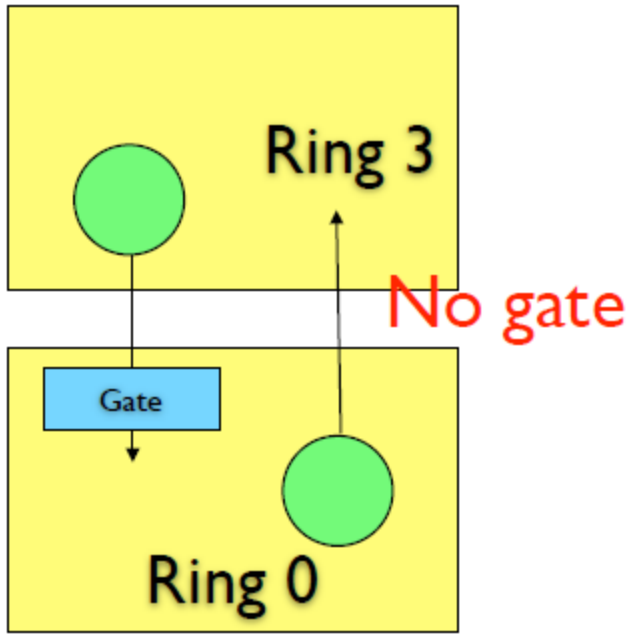
- MLS policy is loaded with the system at boot-time and is otherwise immutable.

Multics Protection System

- Protection State
  - ACL , Ring brackets, Levels in MLS
- Labelling State
  - Not specifically defined.
  - Presumably, creator assigns MLS labels and ring brackets to new segments
- Transition State
  - Ring brackets define - allowed protection domain transitions.
  - No object transitions specified.

Ring transition

- Gate is a special memory address where lower-privilege code can call higher
- Enables OS to control where applications call it (system calls)



Ring transition

Ring transition to a more-privileged ring

- Entry through a special gate segment
  - number of arguments expected;
  - data type on each argument;
  - access requirements for each argument (e.g. R / RW).

- Gate segment = gatekeeper
  - protect the invoked code from potentially malicious input from lower-privileged code

- Return to the calling code in the proper ring number r.

Ring transition

- Transition to a lower-privileged ring
- May leak information
  Each segment in which an argument is contained must be accessible to the called procedure/ some form of copying is necessary
- Higher privileged code protect itself on return
  - caller to provide a gate for return =return gate.
  - Similar in concept to a call gate - multiple calls may result in a stack of return gates (Not in SDW)
  - Supervisor must maintain stack of return gates for the process

Multics Reference Monitor

Mediation

- Security-sensitive operations on segments
- All objects are accessed via a named hierarchy of segments

Tamperproofing

- Reference monitor is part of the kernel ring
- Minimize dependency on software outside kernel

Verifiability

- Lots of code (well, 54K SLOC, but designers thought this was too much)
- MLS for secrecy and rings for integrity (not mandatory)

Vulnerability Analysis

- Evaluation of Multics system security 1972-1973
- Roger Schell and Paul Karger
- Implementation based flaws

Hardware Vulnerability Analysis

- Run the system for a long time
  - Didn't crash, but found one undocumented instruction and one vulnerability
- Indirect Addressing
  - Address provided includes the actual address to use
  - Mechanism only checked the first address
- Result
  - Bypass access checking (complete mediation)

Software Vulnerability Analysis

- Master mode vulnerability
  - Run privileged code with ring 0 permissions
  - Requires a trap to ring 0
  - Expensive as some privileged operations occur frequently (page faults)

- Where to keep signaller ??

Multics Security

- Largely good - but risks remain.
- Powerful mediation, expressive tamperproofing, and verifiable secrecy controls and system integrity controls.
- Challenges:
  - Managing the scope of the TCB,
  - Verifying correctness of system integrity policies
  - Ensuring integrity protection for all processes and segments correctly
  - Correct implementation

Take Away

- Multics originated the development of a "secure operating system"

  -Real attempts were made to achieve reference monitor guarantees and provide a mandatory protection system (e.g., MLS)
- However, it is not easy to satisfy reference monitor guarantees, even when you try

  -Especially, if your system maintainers are not trying
- If you are not trying to satisfy RM guarantees - You won't have anything close (UNIX and Windows)

References & Acknowledgments

- Book: Operating System Security by Trent Jaeger
- William Enck, OS Security lecture notes
- Lecture Notes: http://www.cse.psu.edu/~trj1/cse544-s18/schedule.html