SELinux RHEL5: A benchmark

Lars Strand

INF5071 - Mandatory student assignment

Autumn 2007

SELin...HÆH??

- Developed by NSA. Today: Open Source.
- In mainline Linux kernel since 2.6.
- Fedora since FC2. RHEL since v4.
- Today: RedHat aggressively pushes the development.
- SELinux consist of:
  1. Kernel patches. Uses LSM.
  2. Library 'libselinux' (ls,ps, ...)
  3. Administrative tools (sestatus, semanage, ...)
  4. Security policy.

Access Control

- Discretionary Access Control (DAC): The subjects are in control.

"If an individual user can set an access control mechanism to allow or deny access to an object, that mechanism is a discretionary access control (DAC), also called identity-based access control (IBAC)."

-- M. Bishop, computer security (2003).

- Mandatory Access Control (MAC): Access control enforced by the system – the subjects no longer in (full) control.

"When a system mechanism controls access to and and individual user cannot alter that access, that control is a mandatory access control (MAC), occasionally called a rule-based access control."

-- M. Bishop, computer security (2003).

Security context

- Four security attributes:

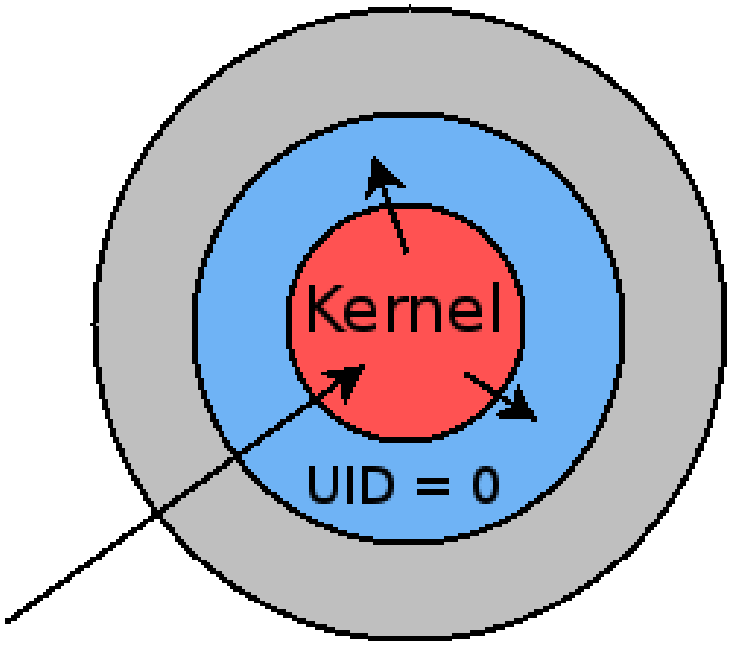
  <user>:<role>:<type>:<category/level>

- These build up a “security context”.
- Example:

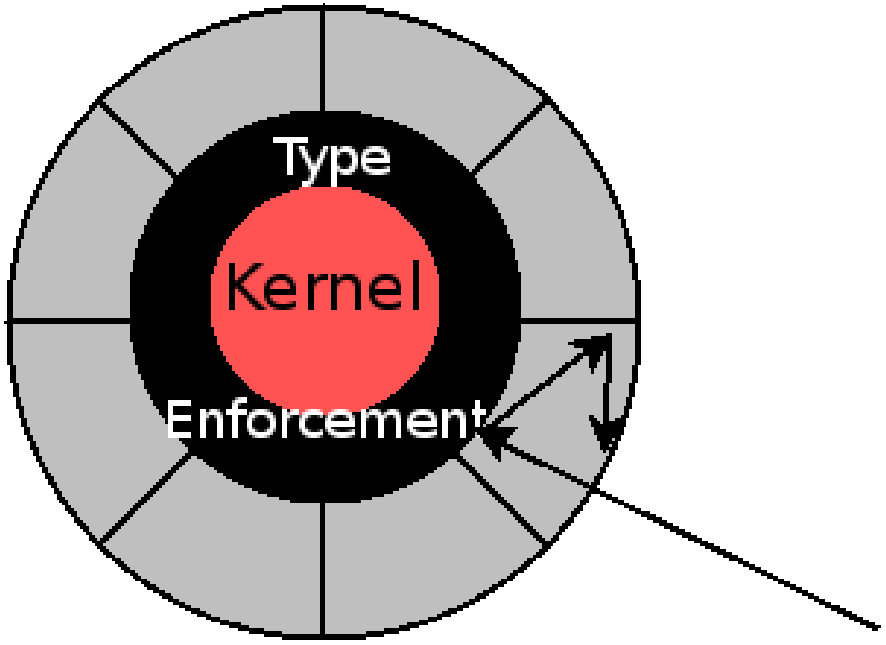
  system_u:system_r:unconfined_t:s0:c0

Security attribute	Name convention	Example name
User	_u	user_u
Role	_u	object_r
Type	_t	unconfined_r
Category/level	(none)	s0:c0

Type Enforcement (TE)

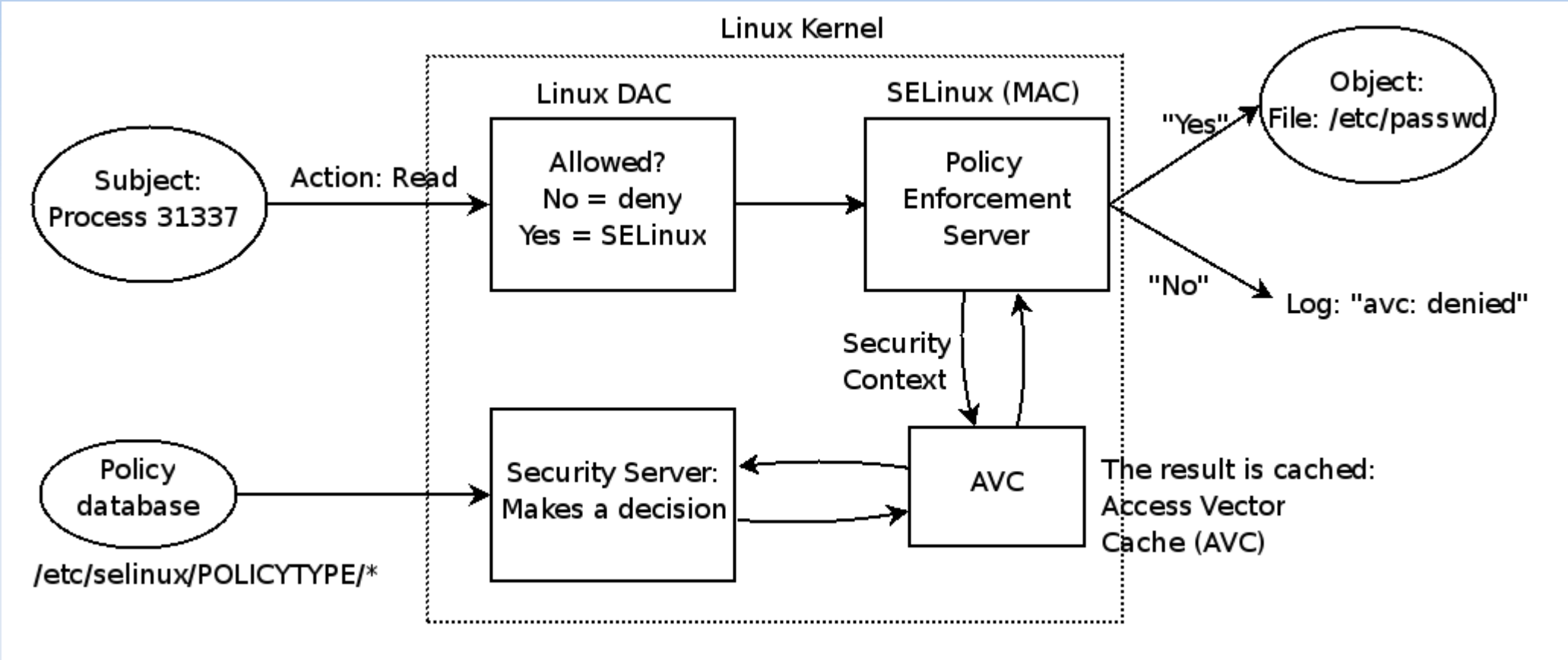


Traditional access control.
UID 0 have full access.

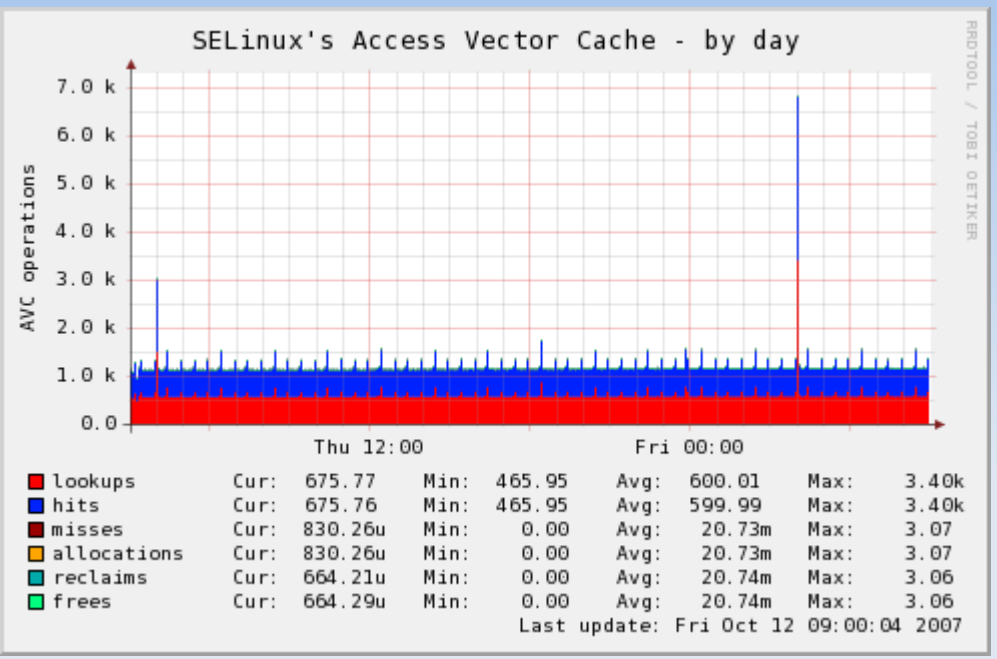


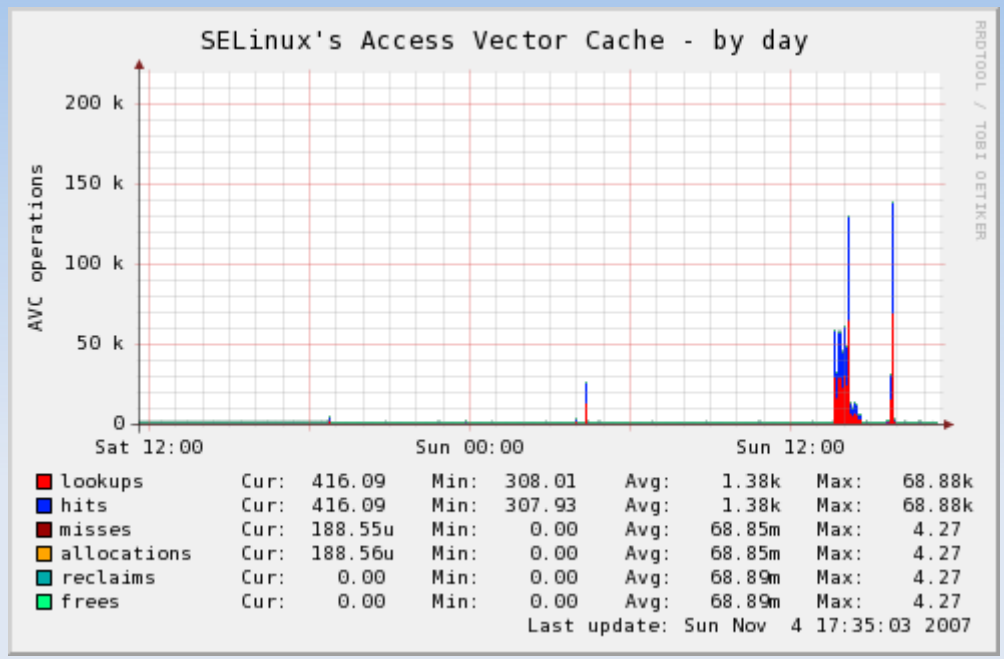
Domain/Type enforcement.
Programs confined in sandboxes.

SELinux MAC

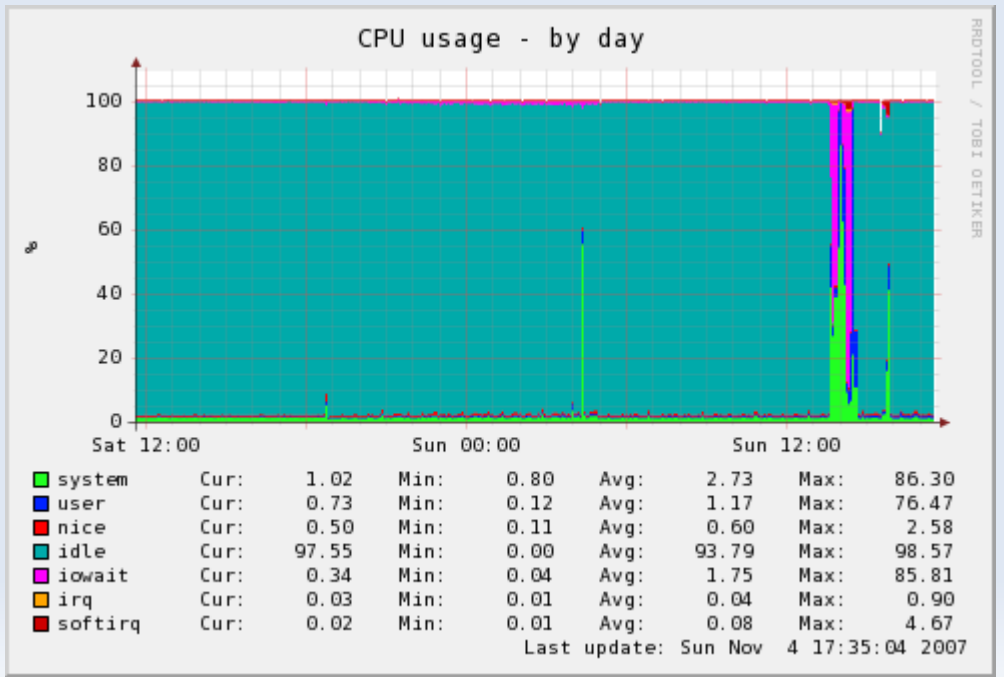

Linux Kernel
Linux DAC
SELinux (MAC)
Subject:
Process 31337
Action: Read
Allowed?
No = deny
Yes = SELinux
Policy
Enforcement
Server
"Yes"
Object:
File: /etc/passwd
"No"
Log: "avc: denied"
Security
Context
Policy
database
/etc/selinux/POLICYTYPE/*
Security Server:
Makes a decision
AVC
The result is cached:
Access Vector
Cache (AVC)

AVC load (Munin)



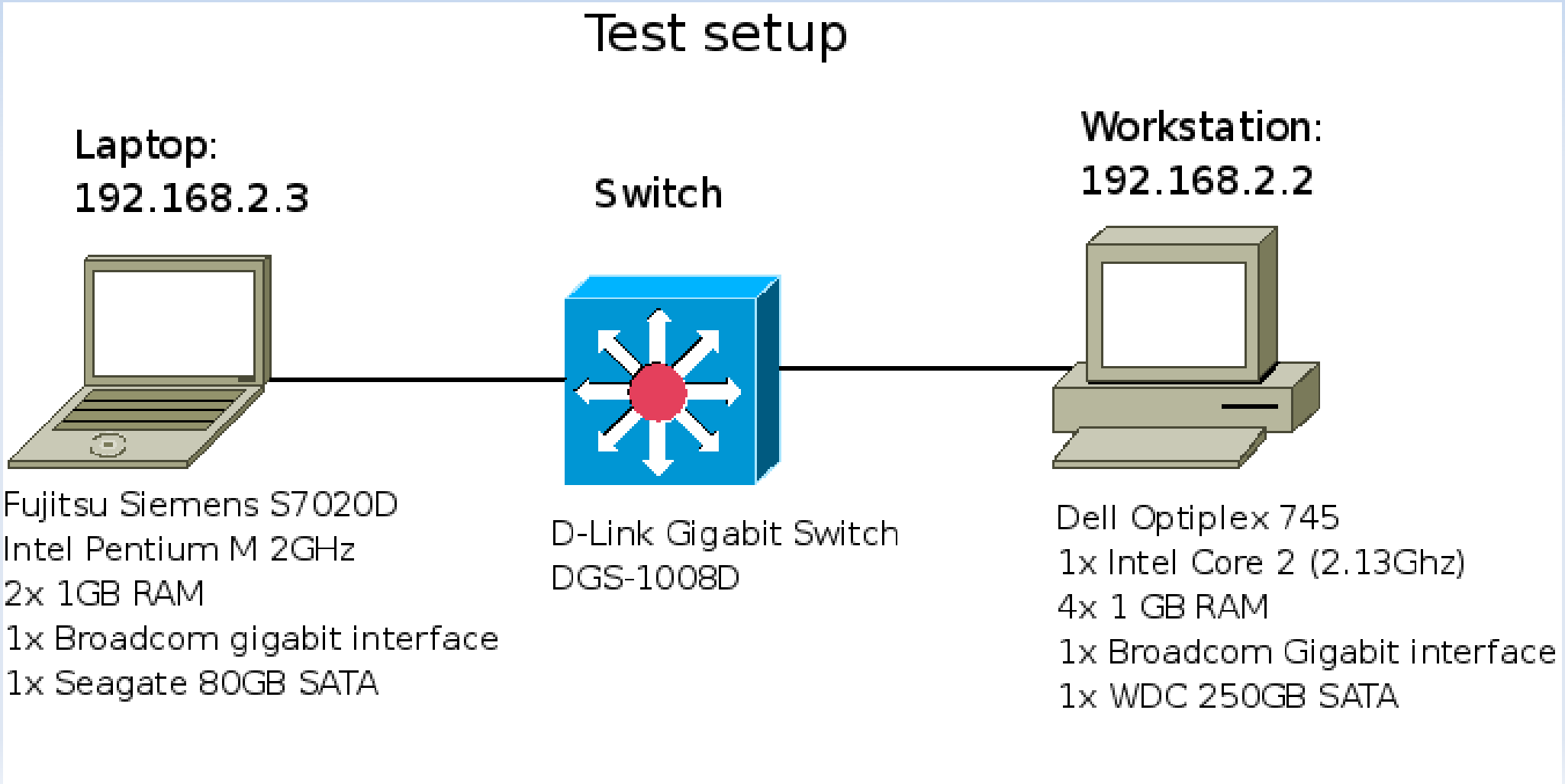


- MySQL benchmark (”run-all-tests”)
- Up to ~1 million queries / second.

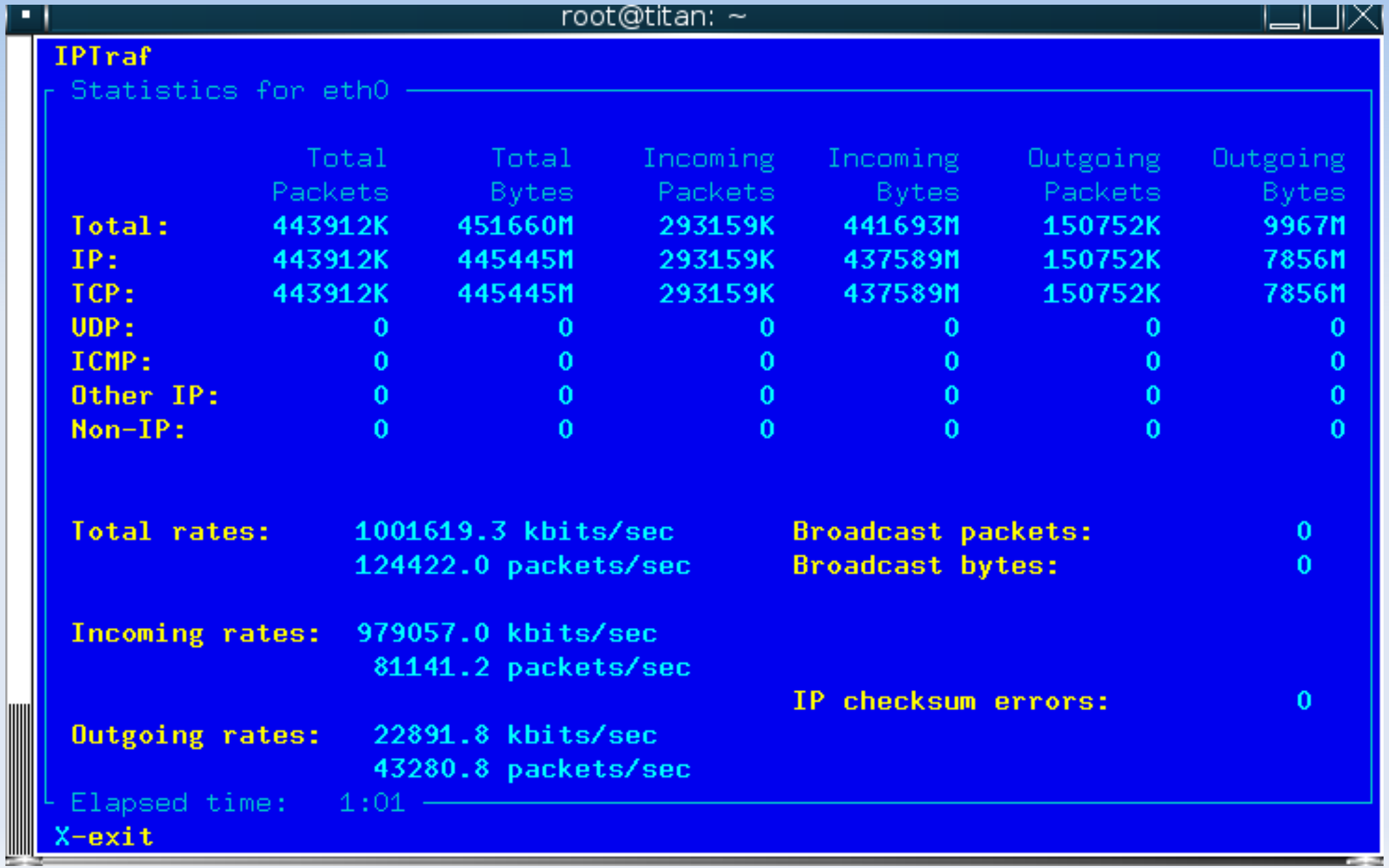


Test setup – two hosts

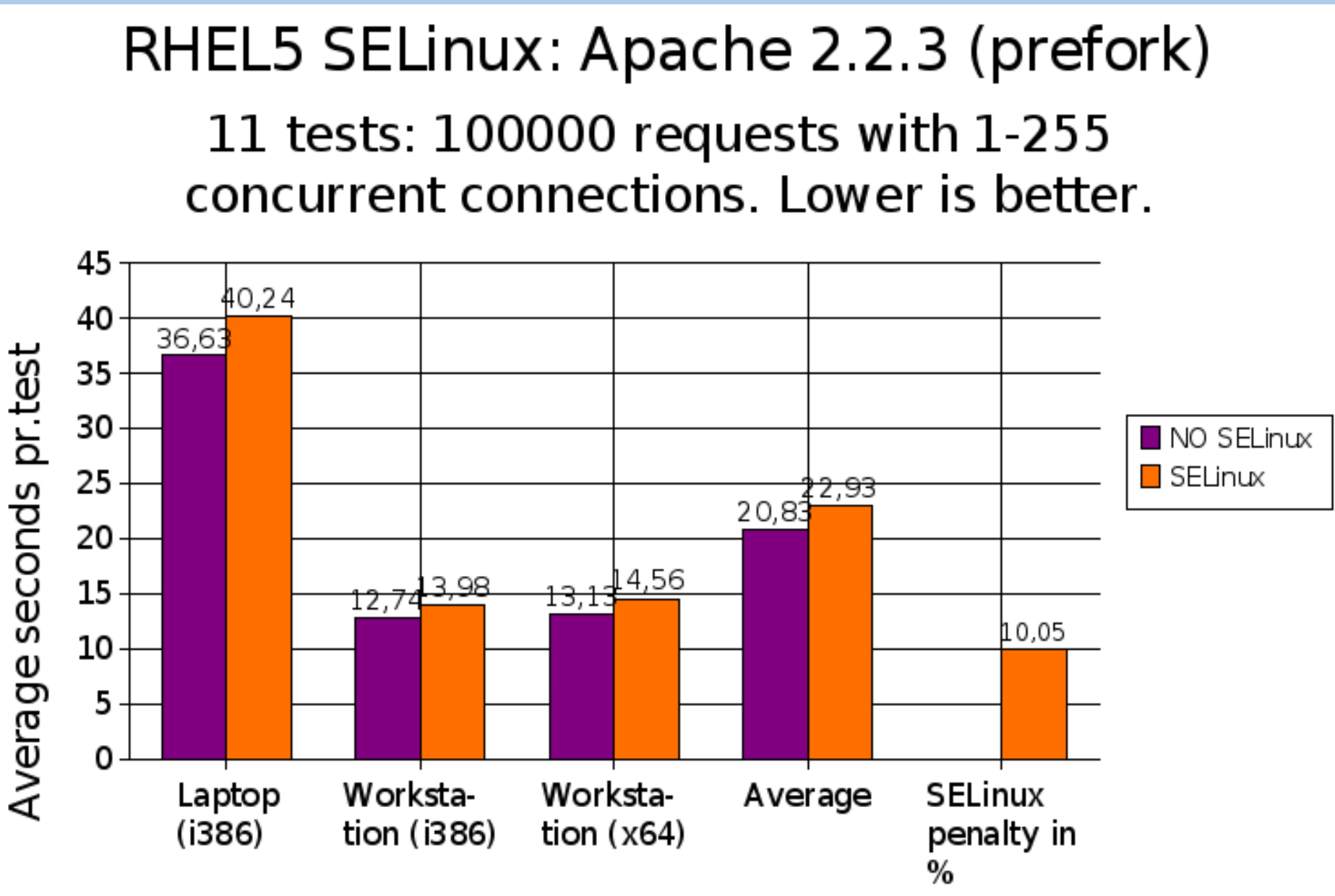
- OS: RHEL5 Server (i386 and x64)



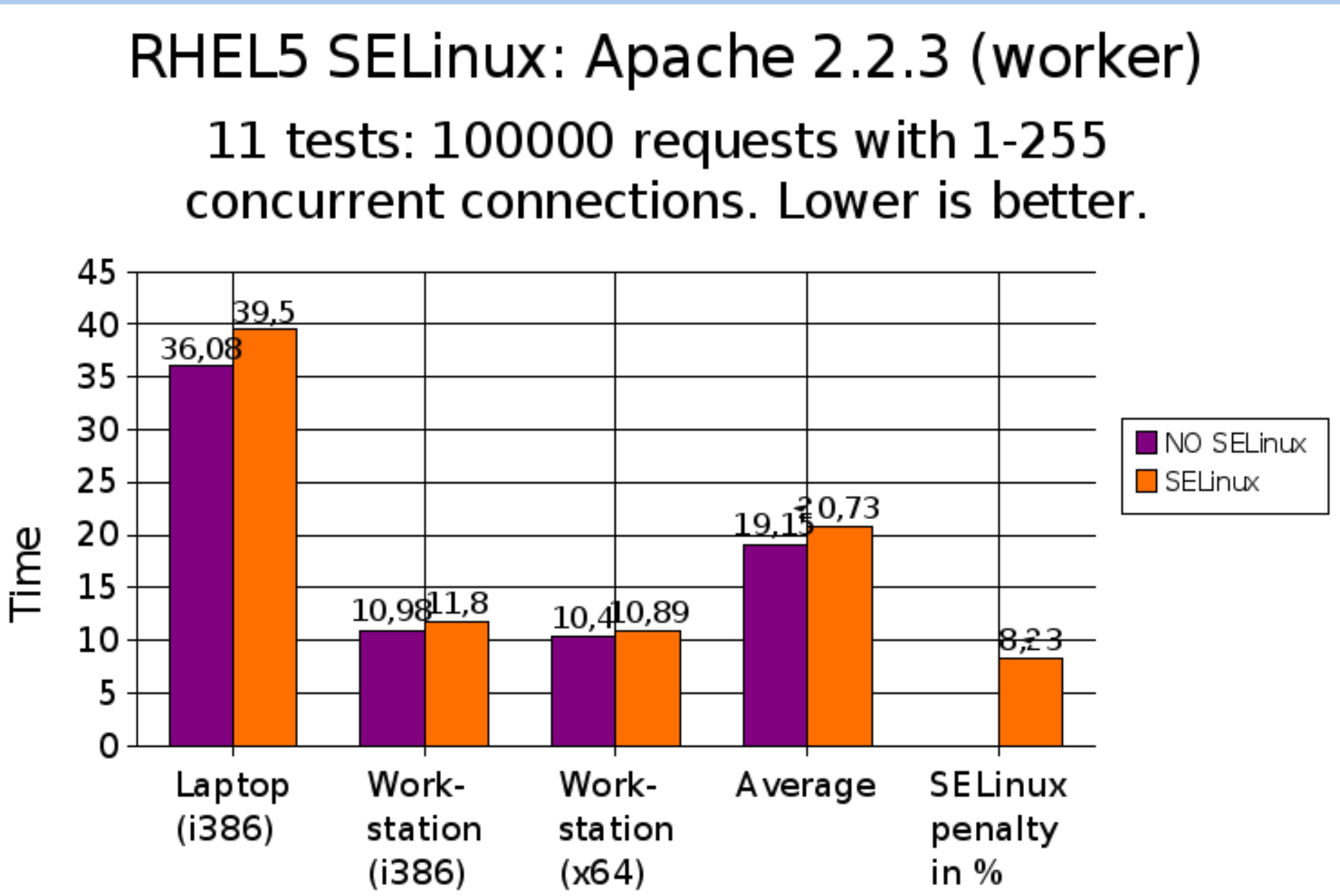
Trivia - sustained 1Gbps



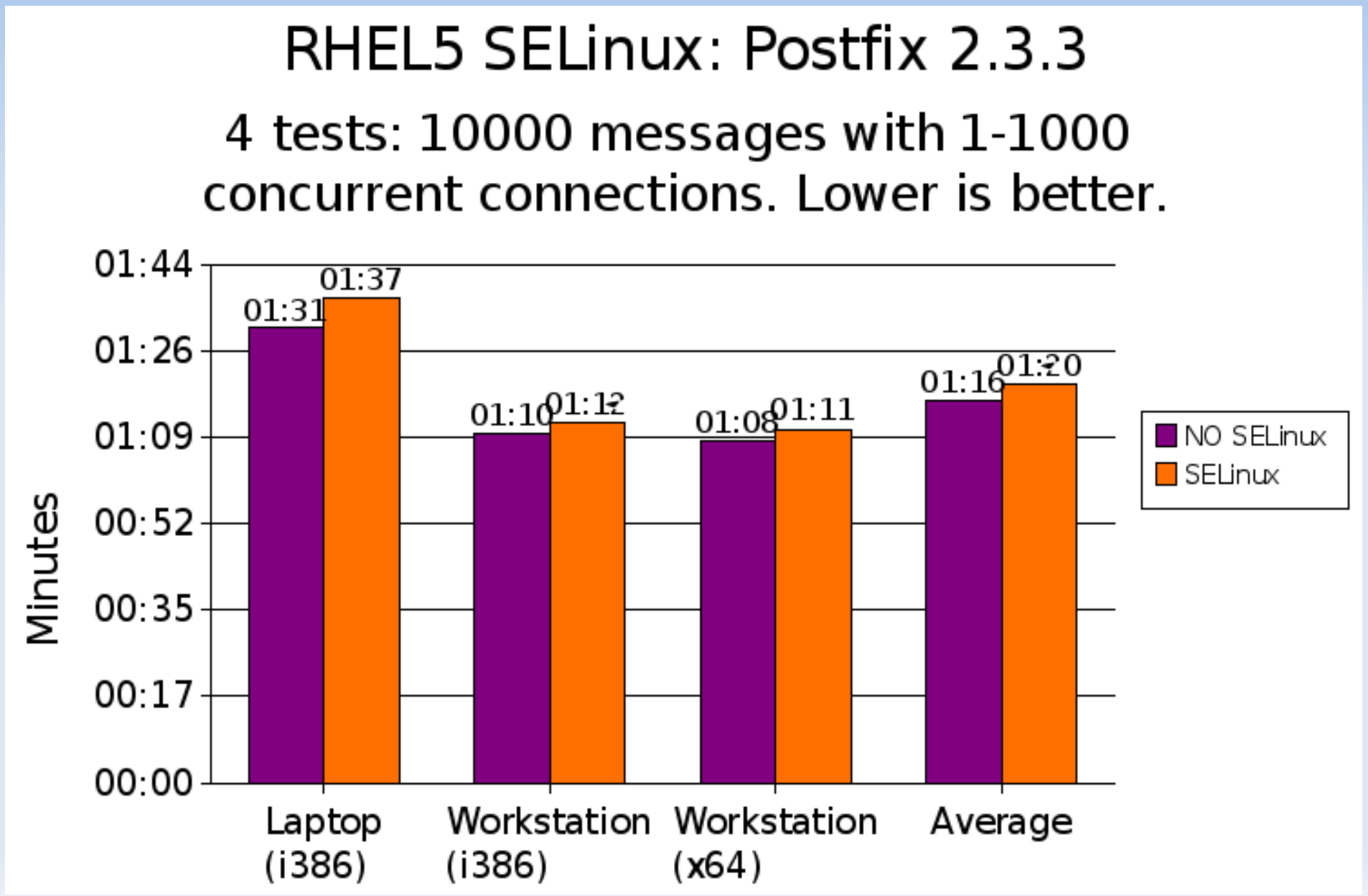
Test 1a: Apache



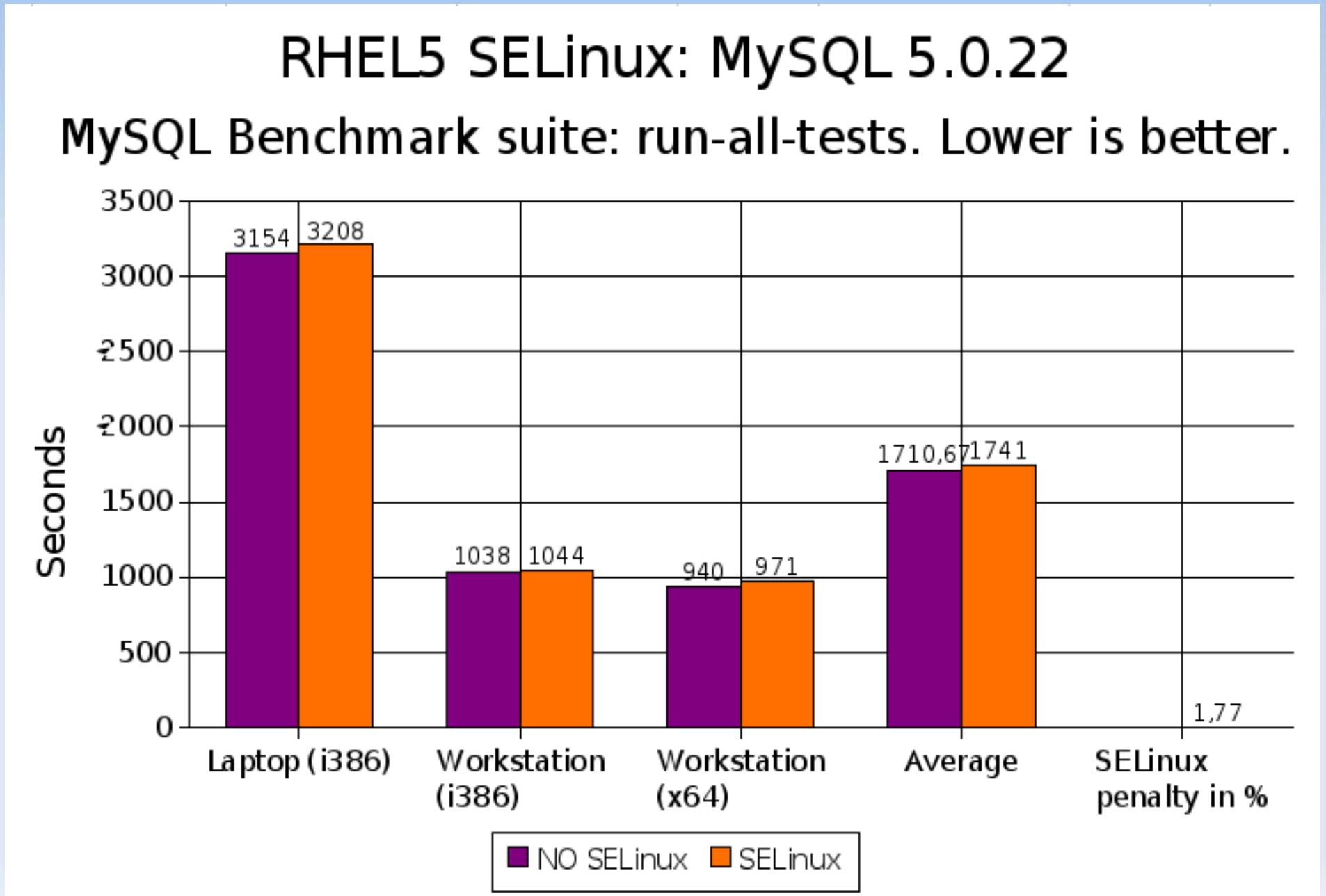
Test 1b: Apache



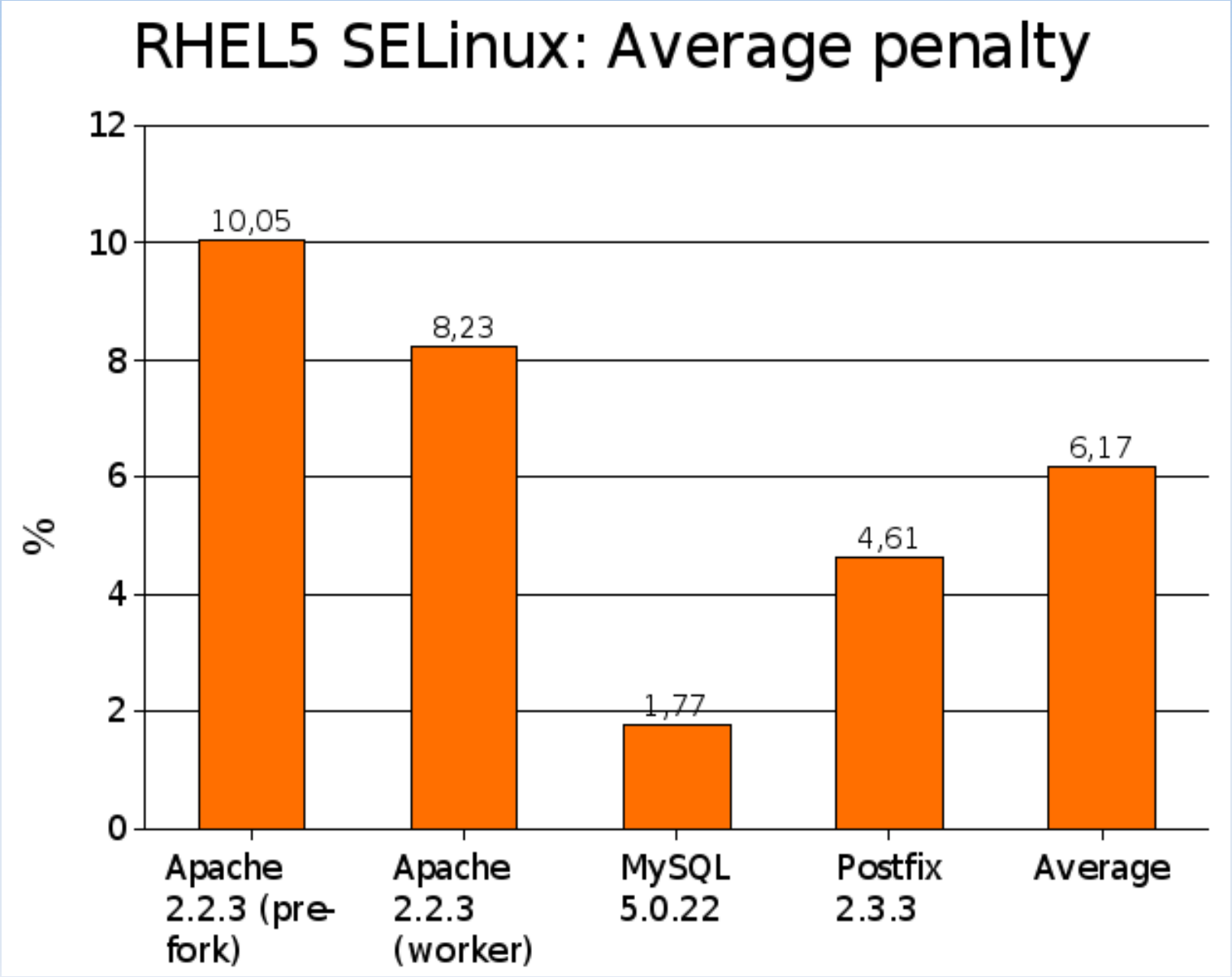
Test 2: Postfix



Test 3: MySQL


RHEL5 SELinux: MySQL 5.0.22
MySQL Benchmark suite: run-all-tests. Lower is better.
Seconds
3500
3000
2500
2000
1500
1000
500
0
3154
3208
1038
1044
940
971
1710,67
1741
1,77
Laptop (i386)
Workstation (i386)
Workstation (x64)
Average
SELinux penalty in %
NO SELinux
SELinux

Total average



Conclusions

- FC FAQ states: ~7% performance penalty.
- These tests show ~6%.
- More CPU bound = less penalty.
- Penalty depends on:
  - How program behaves.
  - Security policy written for the program.
  - Particular usage of the program.
- Dan Walsh: FC 8 has some improved SELinux kernel performance.

Questions?

Read more - full report:

http://blog.gnist.org/article.php?story=RHEL5-SELinux-Benchmark