Linux Kernel Security

Adapting 1960s Technology to Meet 21st Century Threats

James Morris

FOSS.IN/2010
Bangalore, India

Fig. 1

History

“The first fact to face is that UNIX was not developed with security, in any realistic sense, in mind; this fact alone guarantees a vast number of holes.”

Dennis Ritchie, “On the Security of UNIX”, 1979

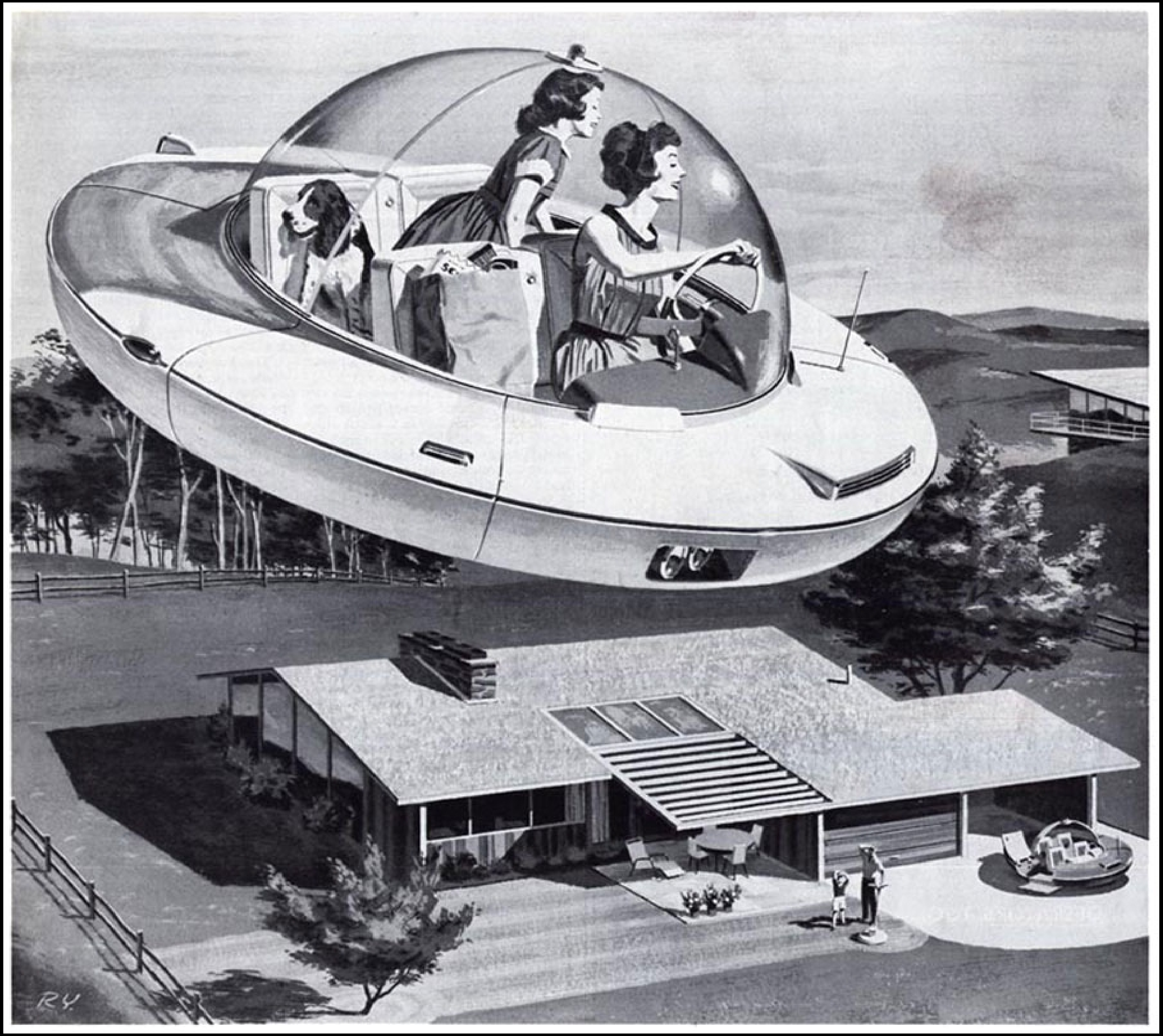


Fig. 2

Unix DAC

DAC is “simple” and somewhat effective, but inadequate for modern environment:

Does not protect against flawed or malicious code

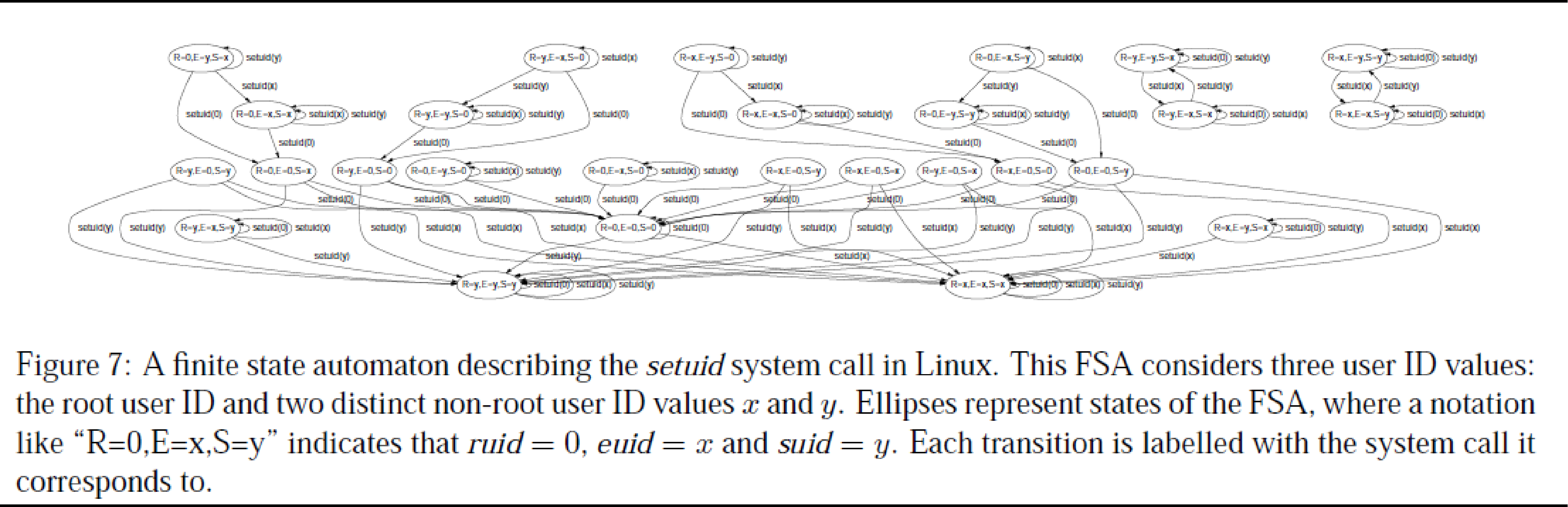


Figure 7: A finite state automaton describing the *setuid* system call in Linux. This FSA considers three user ID values: the root user ID and two distinct non-root user ID values x and y. Ellipses represent states of the FSA, where a notation like "R=0,E=x,S=y" indicates that *ruid* $= 0$, *euid* $= x$ and *suid* $= y$. Each transition is labelled with the system call it corresponds to.

Fig. 3

(Actually, DAC is not simple)

“It must be recognized that the mere notion of a super-user is a theoretical, and usually practical, blemish on any protection scheme.”

(also from Ritchie 1979)

Fig. 4

Enhanced DAC

POSIX Capabilities (privileges)

Access Control Lists (ACLs)

Beyond DAC

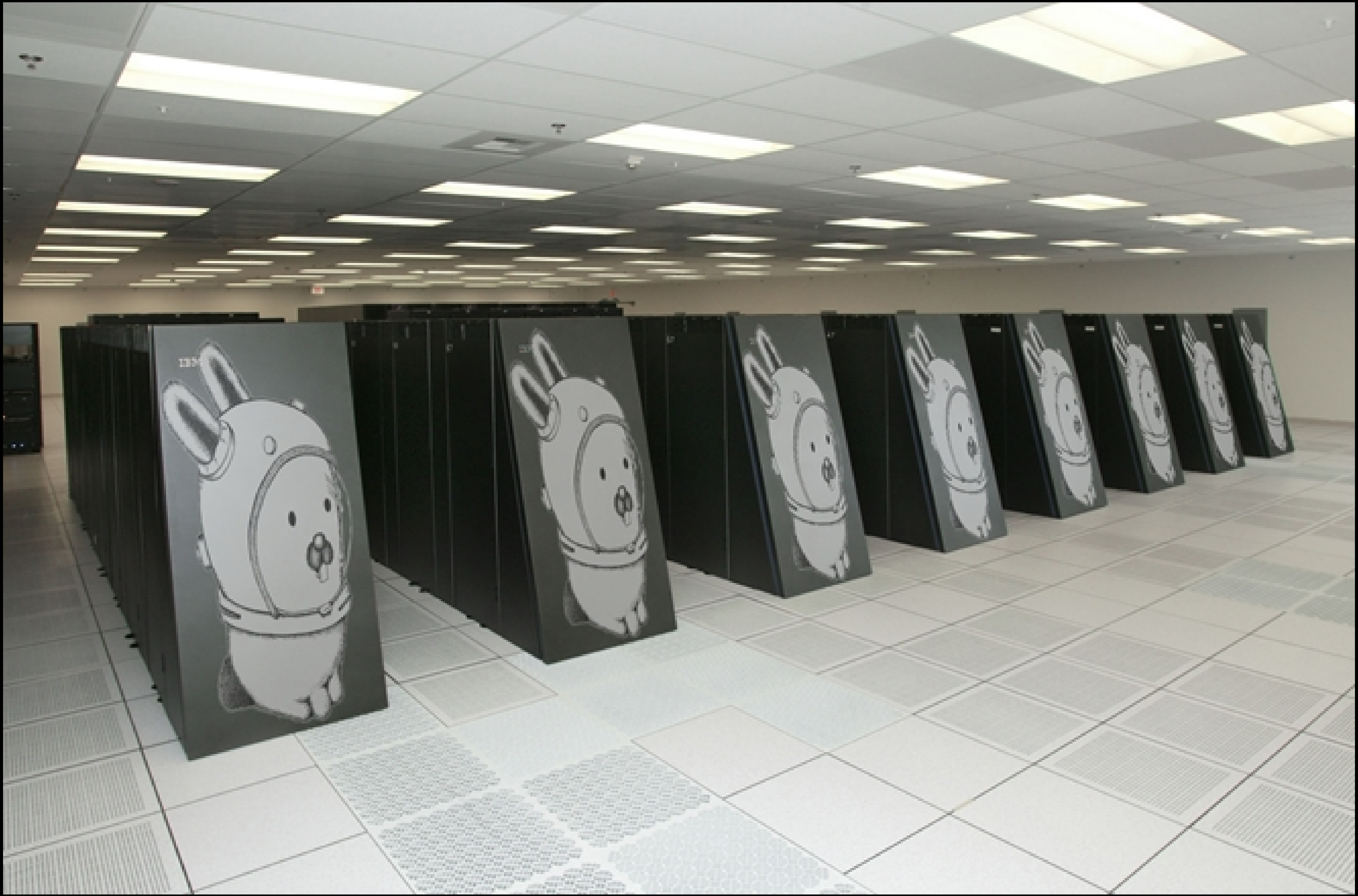


Fig. 5

Namespaces

Network Access Control

Fig. 6

Netfilter

iptables

ebtables

Fig. 7

Cryptography

Disk Encryption:

dm-crypt
ecryptfs

Network Encryption:

IPsec

System Hardening

ASLR

NX

GCC

/dev/mem

MAC policy

Kernel pointers

Fig. 8

Fig. 9

The Inevitability of Failure

The Flawed Assumption of Security in Modern Computing Environments

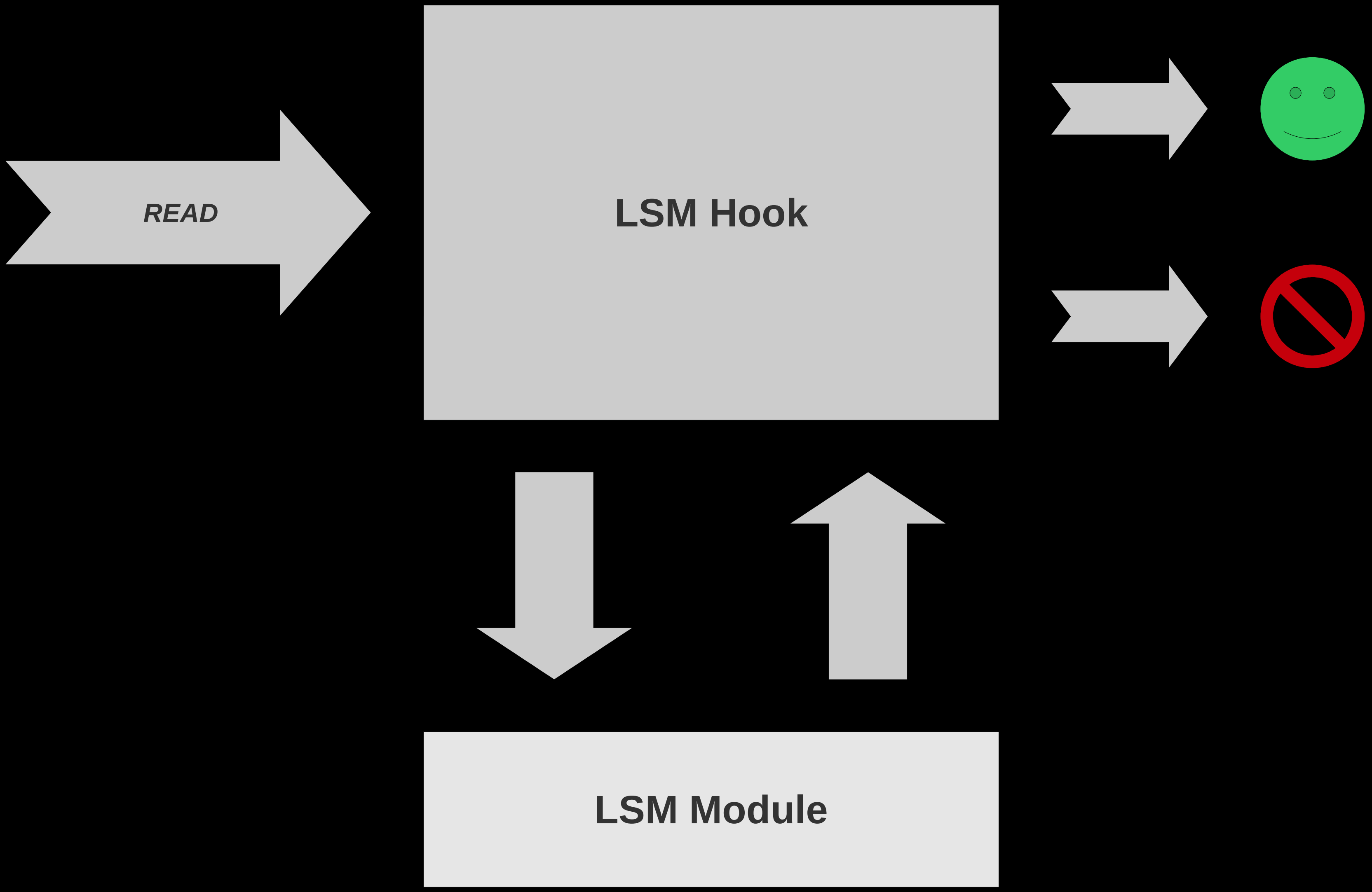
Mandatory security

Trusted / protected path

Assurance

Linux MAC

Linux Security Modules



SELinux

Generalized MAC

Very fine-grained

Policy-flexible

Simplified Mandatory Access Control Kernel (SMACK)

Simple label-based MAC

Policy is written as triples:

subject object [–rwxa]

TOMOYO

Path-based MAC scheme

Automatic real-time policy generation

Policy applied to trees of process invocation

AppArmor

Pathname access control scheme

Security usability via familiar abstractions

Extending MAC

Netlabel

Secmark

NFSv4

sVirt

Audit

Required for certification

Monitor syscall, LSM & misc. security events

Actually quite useful

Integrity & Platform Security

TPM

IMA / EVM

TXT

VT-d

Seccomp

Extremely lightweight sandboxing

Reduces attack surface

Current Status

Meets extremely wide range of security goals

Security features now mainstream

Better equipped to address modern threats

Ongoing Challenges

Continued refinement & hardening

Multiple security models hindering adoption

Threats will continue to evolve

How to Help

Fig. 10

Enable features

Report problems

Share knowledge

Resources

Linux Kernel Security Wiki

LSM Mailing List

LWN Security page

Questions ?

Useful URLs

Kernel Security Wiki
    http://security.wiki.kernel.org/

LSM Mailing List
    http://vger.kernel.org/vger-lists.html#linux-security-module

LWN Security Page
    http://lwn.net/Security/

“The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments”
    http://csrc.nist.gov/nissc/1998/proceedings/paperF1.pdf

LSM Usenix Paper
    http://www.usenix.org/event/sec02/wright.html

Kernel Memory Protection
    http://lwn.net/Articles/329787/

Linux Security Model Comparison
    http://tomoyo.sourceforge.jp/wiki-e/?WhatIs#comparison

SELinux
    http://selinuxproject.org/
"Have You Driven an SELinux Lately?" (OLS paper on current state)
    http://namei.org/ols-2008-selinux-paper.pdf
"Anatomy of Fedora Kiosk Mode"
    http://namei.org/presentations/fedora-kiosk-mode-foss-my-2008.pdf
"SELinux Memory Protection Tests"
    http://people.redhat.com/drepper/selinux-mem.html
"A seatbelt for server software: SELinux blocks real-world exploits"
    http://www.linuxworld.com/news/2008/022408-selinux.html

SMACK
    http://schaufler-ca.com/

AppArmor
    http://en.opensuse.org/Apparmor

TOMOYO
    http://tomoyo.sourceforge.jp/

"POSIX file capabilities: Parceling the power of root"
    http://www.ibm.com/developerworks/library/l-posixcap.html

"POSIX Access Control Lists on Linux"
    http://www.suse.de/~agruen/acl/linux-acls/online/

Useful URLs ...

"Implementing Native NFSv4 ACLs in Linux"
http://lca2009.linux.org.au/slides/79.tar.gz

“Applying mount namespaces”
http://www.ibm.com/developerworks/linux/library/l-mount-namespaces.html

“Disk encryption in Fedora: Past, present and future”
http://is.gd/16012

“Limiting buffer overflows with ExecShield” (2005)
http://www.redhat.com/magazine/009jul05/features/execshield/

“Linux Kernel Heap Tampering Detection”
http://phrack.org/issues.html?issue=66&id=15#article

“System integrity in Linux”
http://lwn.net/Articles/309441/
“Linux kernel integrity measurement using contextual inspection” (LKIM)
http://portal.acm.org/citation.cfm?id=1314354.1314362

Intel TXT Site
http://www.intel.com/technology/security/

IBM TCPA Resources
http://www.research.ibm.com/gsal/tcpa/tcpa_rebuttal.pdf

Invisible Things Labs
http://theinvisiblethings.blogspot.com/

Image Credits

1. Bell Labs
2. Duke University Ad*Access
3. Hao Chen, David Wagner, and Drew Dean.
4. “nofeel” (flickr)
5. Unknown
6. Ian Lloyd (flickr)
7. James Morris
8. Steve Jurvetson (flickr)
9. Michael Scott (flickr)
10. Alfred T Palmer (LoC)