TrustedBSD Project Update

11 May 2006

Robert N. M. Watson

Security Research
Computer Laboratory
University of Cambridge

UNIVERSITY OF CAMBRIDGE

Introduction

- TrustedBSD Project started in April, 2000
- Goals to provide
  - Infrastructure for advanced security services
  - Advanced security functionality
- Accomplished a lot in six years
- Updates on recent activities
  - MAC Framework discussions
  - Audit implementation
  - NFSv4 ACLs

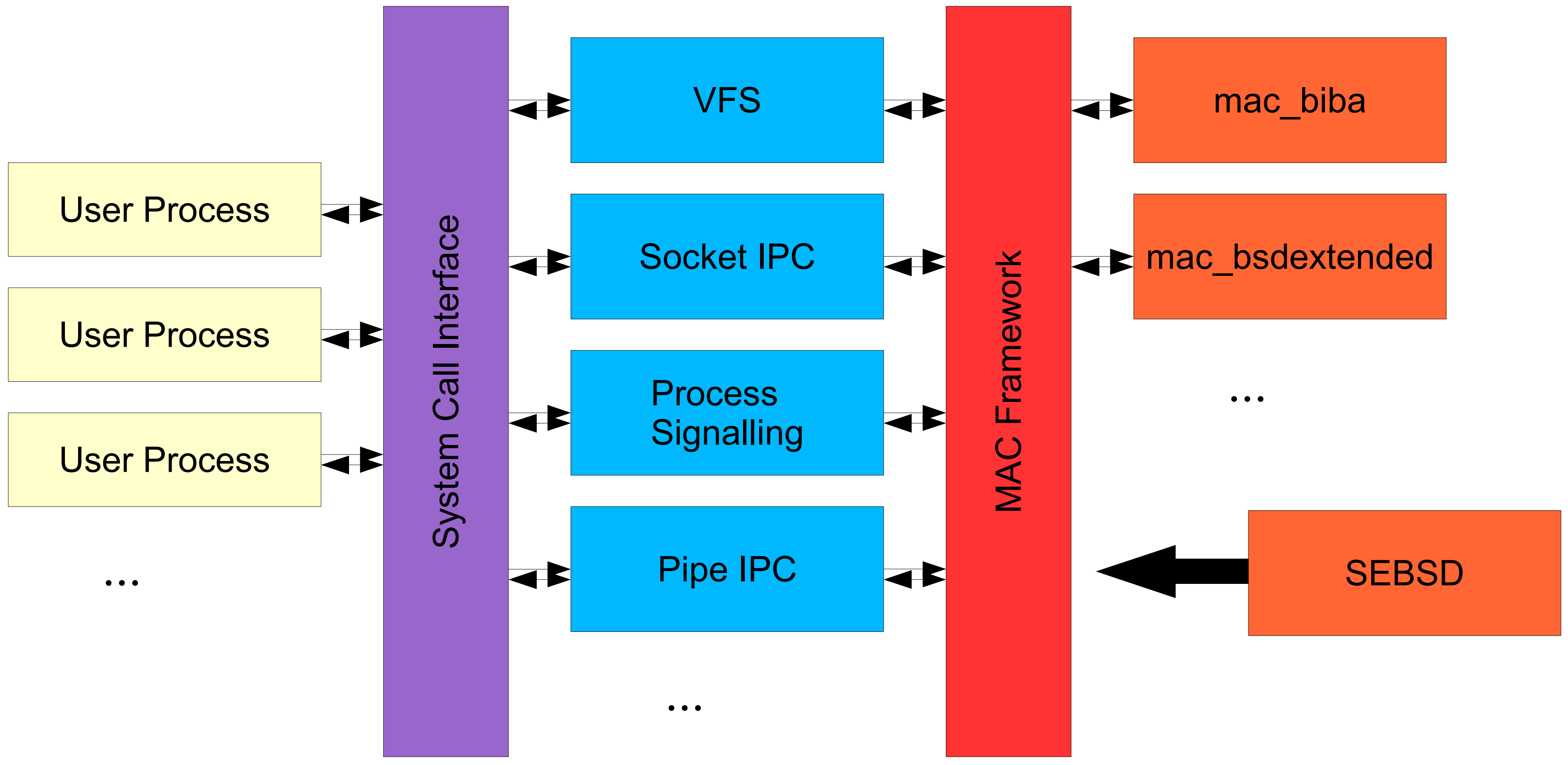
TrustedBSD Feature List Reminder

- Infrastructure
  - OpenPAM, NSS, UFS1 EAs, UFS2, GEOM, GBDE
  - Access control cleanup
- Stuff
  - TCP syncache, TCP syncookies, TCP TW
- Features
  - ACLs
  - MAC Framework, MAC policies
  - Audit

TrustedBSD MAC Framework Retrofit Discussion Summary

- Extensible kernel access control mechanism
- TrustedBSD MAC Framework merged in 2002
  - Followed two years of DARPA-funded R&D
- We now have significant real-world experience
  - At least half a dozen significant third party security policies written
- Time to review situation, and decide whether architecture meets needs going forward
  - If we haven't learned anything, we weren't trying

Overused Slide on MAC Framework Architecture



Proposals on Table

- Options MAC in GENERIC
  - Requires very careful look at performance
  - Locking, memory allocation model revision
- Broad range of syntactic cleanup
  - Entry point naming consistency, etc.
- IPv6, IPSEC support
  - Prototype labeling and access control explored
- Revised extensible label mechanism
- Integration with Audit

Additional MAC Framework Issues

- Entry points for system call entry/exit to allow system call wrappers
- Provide infrastructure for MAC policy modules desirable
  - Increasing number of third party moduls
  - Not desirable/possible to put all in src

Larger Directional Changes

- Allow plugging of current DAC/privilege models
  - UNIX DAC (permission/ACLs)
  - UNIX superuser
  - UNIX IPC protections
  - UNIX inter-process access control
- Revised system privilege model
  - Suser to... ?

Retrofit Schedule

- Goal to ship moderate revisions to MAC Framework kernel interfaces in 7.0
  - That means 12-18 months to shake out
  - Sounds about right
- Will require third party vendors to update their MAC modules
  - Mostly syntactic changes, but should help with module structure
- Helping hands welcome!

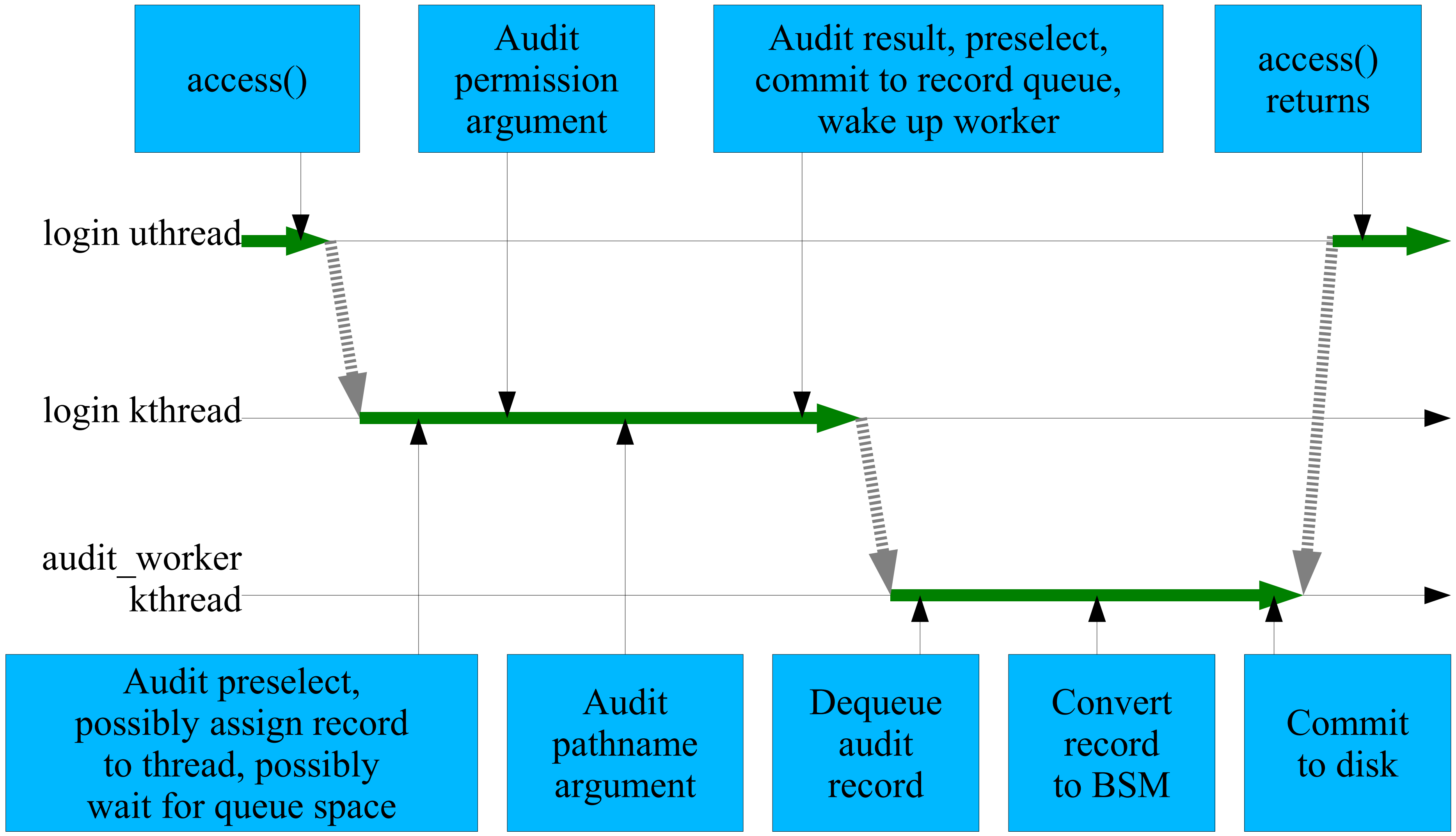
TrustedBSD Audit

- Last year, I told you about exciting new feature
  - Well, it took a bit, but it's there now :-)
- Security event audit
  - Derived from open source Apple audit code
    - Implemented by McAfee Research
  - Fine-grained, configurable, reliable security logging
  - Produce post-mortem trails, as well as live event streams for intrusion detection and analysis
  - Meets requirements for CAPP evaluation

Audit High Level Design Traditional Features

- Token-stream BSM log format
  - De facto industry standard API/file format from Sun
- Records describing security-relevant events
  - Many system calls
  - Authentication, system management, etc
- Reliable trail
  - Bounded loss in the presence of failure, fail-stop support, etc.

Sample Audit Control Flow

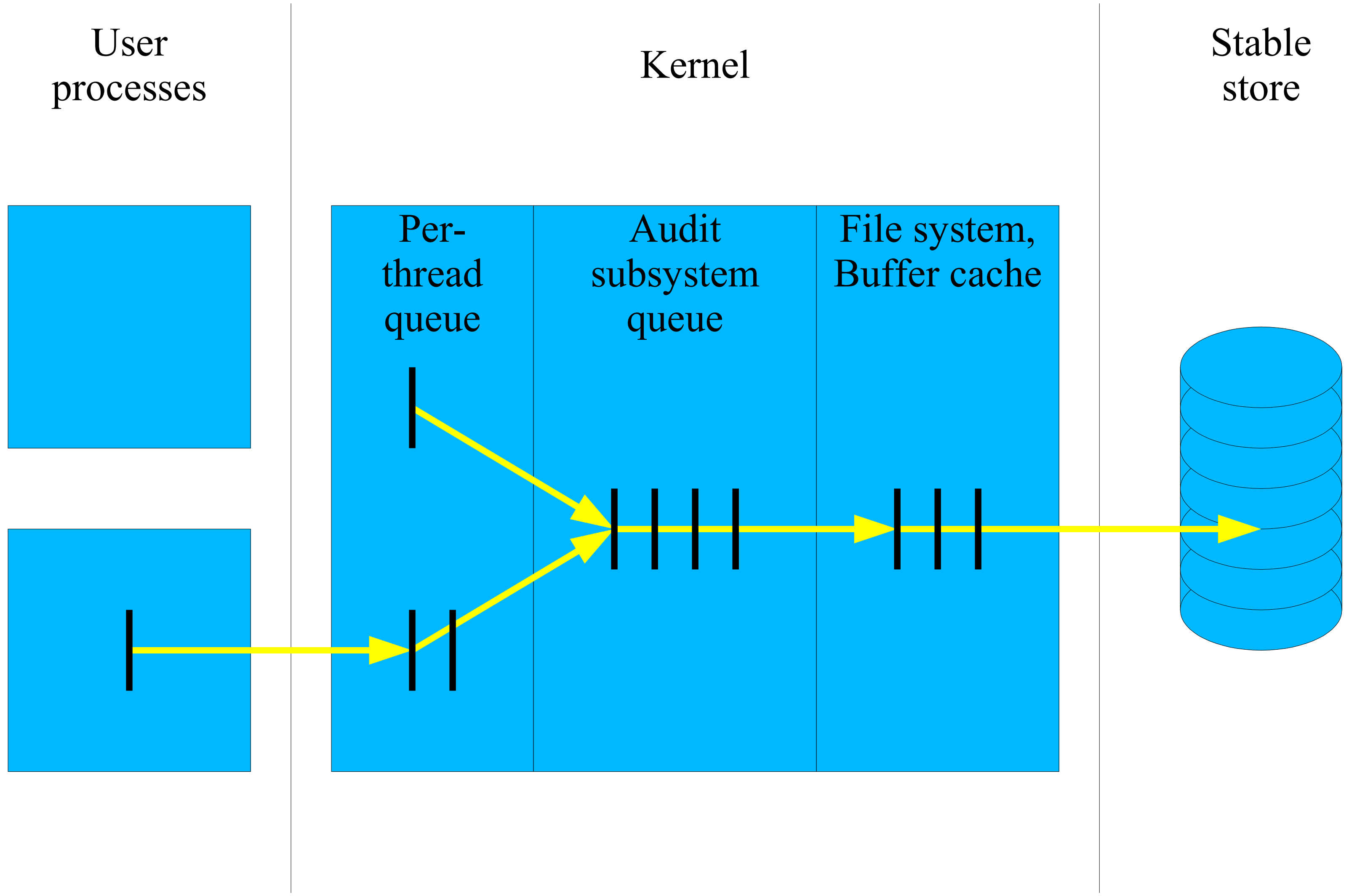

access()
Audit permission argument
Audit result, preselect, commit to record queue, wake up worker
access() returns
login uthread
login kthread
audit_worker kthread
Audit preselect, possibly assign record to thread, possibly wait for queue space
Audit pathname argument
Dequeue audit record
Convert record to BSM
Commit to disk

UNIVERSITY OF CAMBRIDGE

TrustedBSD Audit Implementation Less Traditional

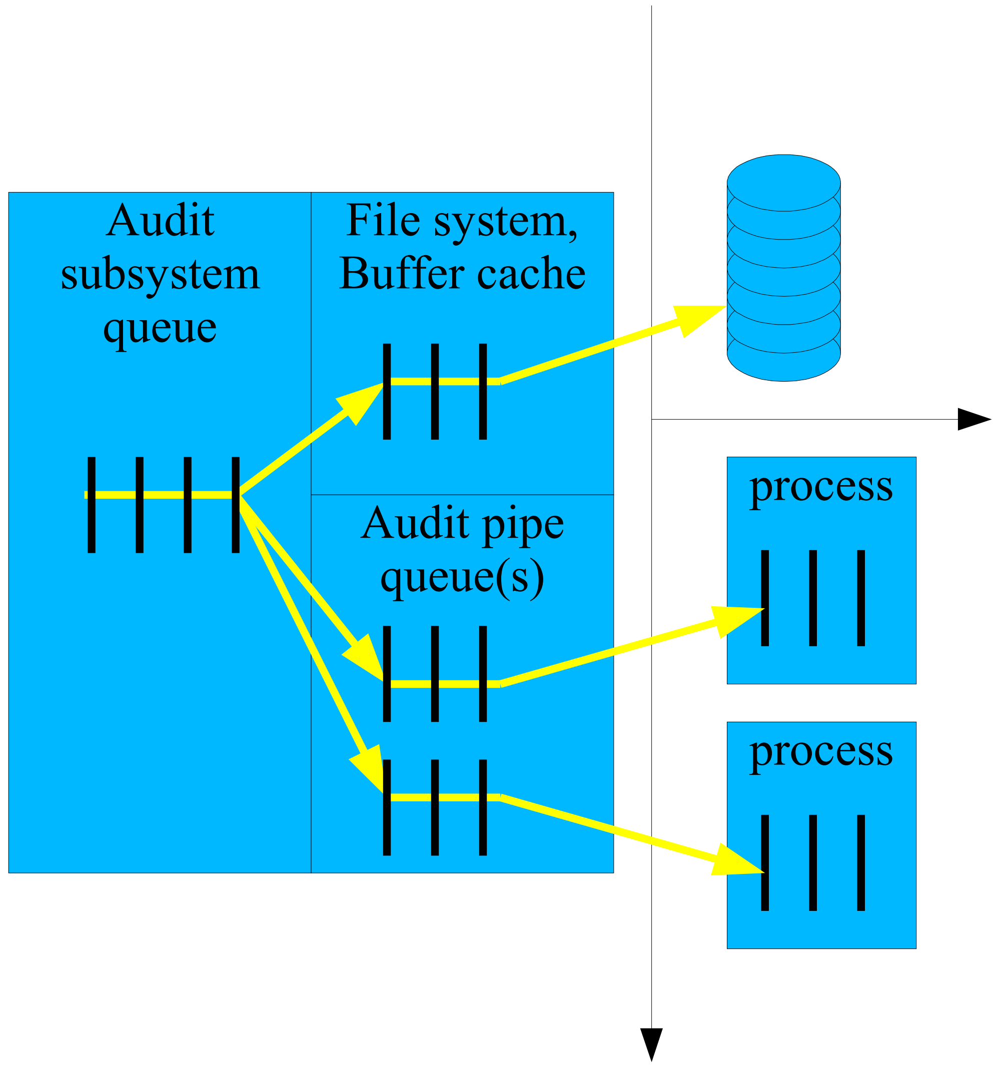
- Classic motivation for including Audit is trusted system evaluation
  - All decent protection profiles require security audit
- More immediate reason is intrusion detection
  - Changes focus of implementation
  - Still want reliable, configurable, fine-grained
  - Also want concurrent stream delivery to processes
  - Want per-stream configuration
- Audit pipes

Audit Queuing



Audit Pipes

- Audit pipes provide live record feed
  - Lossy queue
  - Discrete audit records
  - Independent streams
  - Independent preselection



Audit Event Daemon

- Want to support pluggable analysis and processing services
- Auditeventd
  - Shared library modules
  - Amortizes parsing costs for token stream
  - Common configuration format
- No modules currently, but easy to write
  - Module presented with a series of parsed token arrays containing event circumstances, arguments

Audit Summary

- Audit now largely merged to CVS HEAD (7.x)
  - Some areas of further work required
    - Additional system call auditing (ACLs, EAs, MAC, ...)
    - Additional application auditing (management tools)
- Plan to merge to RELENG_6 for 6.2
  - Not quite yet, but soon
- Feature work still going on
  - Audit pipes especially
  - Interested in multi-trail support

NFSv4 ACLs

- Current TrustedBSD ACLs based on POSIX.1e
  - Obvious choice at implementation time
  - Less obvious choice now
- NFSv4 ACLs are essentially Windows ACLs
  - Notionally similar, semantically quite different
- Mapping from POSIX.1e to NFSv4 is terrible
  - Internet draft reads “It can be done”
  - Between the lines, “But don't”

Tentative Strategy

- Surprisingly, Apple has made NT ACLs fit behind POSIX.1e API
  - But not POSIX.2c command line tools
- Sun also exploring NFSv4 ACLs in ZFS
  - Also investing in improving POSIX.1e mapping
- Create parallel ACL implementation
  - kern_acl.c -> subr_acl.c, subr_acl_posix1e.c
  - Add subr_acl_nfsv4.c
- UFS flag will specify desired ACL model

Lots of Open Questions

- What to do about command line tools?
  - Will need to look in detail at Apple, Sun choices
- What to do about APIs?
  - New ACL_TYPE_?
  - Take this opportunity to roll struct acl format to support longer ACL lengths?
  - Will require compatibility system calls
- Application adaptation needs to be done also
- NFSv4 server/client integration also desirable!

NFSv4 ACL Status

- Have read the NFSv4 RFC
  - Rather non-specific, “See NT”
  - Asked on mailing list, two days later Sun posted draft with proposed semantics
- Have started breaking out ACL code into parts
- Started on system call compatibility
- Help wanted