New Security Features in OpenSolaris and Beyond

Jan Pechanec, Mark Phalan
Sun Microsystems, Inc.

OSDevCon 2008, Prague
06/2008

Disclaimer

- this presentation doesn't bring to you an exhaustive list of new security features in OpenSolaris, or all new features under development
- we focus on a selected lists of contributions
- with some examples and demos
- as of June 2008, for features under development the details are subject to change as work progresses

Contents

- filesystem encryption
  - lofi encryption
    - and short demo
  - ZFS crypto
- Key Management Framework (KMF)
  - SunSSH as a KMF consumer
- OpenSSL PKCS#11 engine
  - using CF through the PKCS#11 engine
  - SunSSH as an PKCS#11 consumer
  - and demo

Contents (cont'd)

- libmd – message digests library
- Kerberos enhancements
- Validated Execution
- Solaris Trusted Extensions
  - > and demo
- some features we won't talk about today

lofi(7D) Encryption

- lofi is an **existing loopback file driver**
  - exports a file as a block device
  - R/W on it is translated to the underlying file
  - initially created to mount ISO images
- `lofiadm(1M)`extended to support requesting encryption and setting the key
- no data integrity protection (yet)
  - use of non-expanding mode with built-in integrity protection is planned (e.g., CCM)
- short demo using current prototype

ZFS on-disk Encryption

- Phase 1
  - **per dataset policy for enabling encryption**, including algorithm and key length
  - per dataset keys wrapped by single per pool key
  - pool key from pass phrase using PKCS#5 PBE
  - pool key can be stored in PKCS#11 token
  - **development complete**, under code review
- Phase 2
  - remote key manager using KMF
  - zone / TX Label key management delegation
  - per user key management

ZFS on-disk Encryption (cont'd)

- Phase 3
  - secure deletion
  - key escrow (via KMF functionality)
- Phase 4
  - encrypted root file system (KCF early in boot issues)
  - HA/Cluster ZFS and encrypted data sets

Cryptographic Framework (CF)

- provides cryptographic services
  - to users and applications through commands
  - a user-level API
  - a kernel-level API
  - a plug-in framework using PKCS#11
- see `cryptoadm list` on your system
- *tokens* are used through *slots*
- see Wolfgang Ley's presentation for more information
  - presented at this conference

OpenSSL PKCS#11 Engine

- an OpenSSL *engine* is a container for an implementation of crypto algorithms
- the ENGINE API manipulates the engines
- **the PKCS#11 engine accesses the CF through the PKCS#11 API**
  - the engine is internally developed at Sun
  - not shipped with the OpenSSL source code
  - a patch for the latest OpenSSL version exists
- originally developed to accelerate Apache on Niagara-1 by use of `ncp(7D)`

OpenSSL PKCS#11 Engine (cont'd)

- the situation changes with **`n2cp(7D)`** and machines based on the T2 chip (Niagara-2)
  - offers hardware acceleration of symetric crypto and message digests
  - **lots of potential consumers now**
- `openssl engine -vvv -t -c` shows the list of supported mechanisms
- support for offloading more mechs under development
  - e.g., AES counter mode (AES CTR), SHA-2, ...

SunSSH & PKCS#11 Engine Support

- problem: SunSSH is “slow” on Niagara2
  - because all SSH implementations are slow
  - problem: **SSH apps are single-threaded**
- **what we can do ?**
  - use hw acceleration using `n2cp`, `mca` (SCA-6000)
  - parallel processing of data (with CTR mode)
- hw acceleration: 2 ways to do it
  - through the OpenSSL PKCS#11 engine
  - get rid of OpenSSL & use PKCS#11 API directly

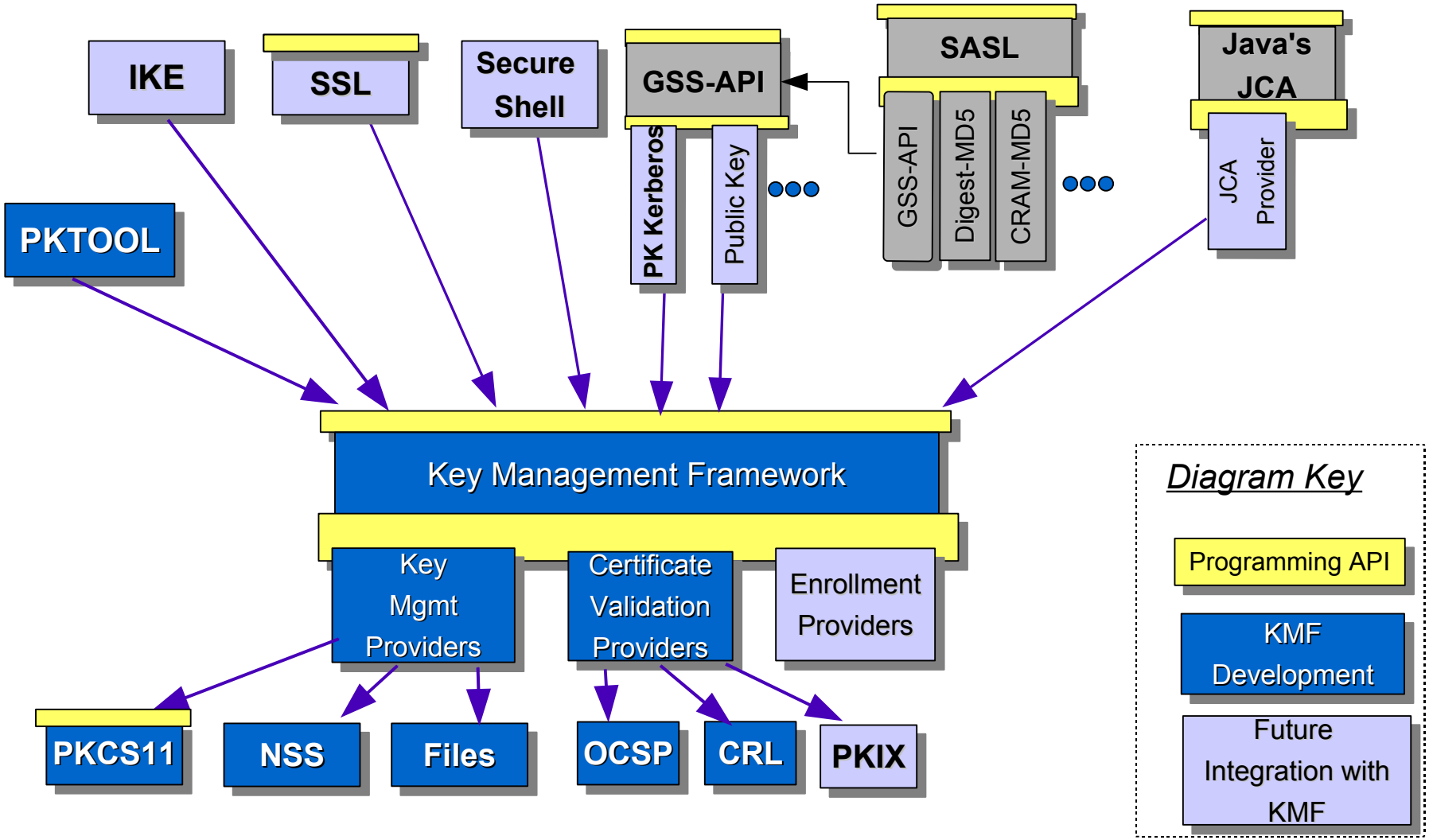
SunSSH with the Engine (cont'd)

- **switching to the PKCS#11 engine** through OpenSSL seemed faster to do
- expectations – **how fast can we go?**
  - don't forget the inherent overhead of CF
  - offloading small data packets to hardware is slower than using the OpenSSL code
  - each SSH packet is processed twice (HMAC, encryption)
- demo with the SunSSH prototype using the PKCS#11 engine
  - 2.5x faster using default AES-128-CTR on T2

Key Management Framework (KMF)

- before KMF, several existing “keystore systems” for designing a PKI application
  - NSS, OpenSSL, PKCS#11 API
  - **problem:** not compatible with each other
  - **problem:** no PKI policy enforcement
- KMF provides a consistent mgmt interface
  - through the `pktool(1)` utility
- the KMF API hides the details of the underlying keystores
- KMF provides a system wide security policy

KMF Big Picture



KMF (cont'd)

- more on KMF API
  - **Solaris 10:** project private, architecture not pluggable and not for 3rd party application use
  - **new API in Nevada:** public API, pluggable architecture
  - and, new things to come
    - PKIX (so far, path length=1 only is supported)
    - certificate-to-name mapping
- `pktool(1)` for managing certs and keys
- `kmfcfg(1)` for managing policy files

SunSSH as a KMF Consumer

- SSH2 protocol: ***publickey* auth with public keys vers. X.509 certificates**
- *"are you sure you want to continue connecting (yes/no) ?"* is annoying
  - > and might not be possible to say "yes" safely
- what we can get using X.509 certs
- no RFC for X.509 pubkey auth, only drafts
  - > and, IETF SSH work group closed down
- will need cert-to-name mapping in KMF

KMF API Example Usage

- our objective – **to validate a certificate**
- only the certificate is needed
  - everything else is in the policy file
  - location of the Trusted Anchor (TA) will be in the policy, too
    - now it must be provided to the KMF function for certificate validation
- **source code example**
  - using the SunSSH X.509 prototype code
  - calls `kmf_cert_validate()` using a chosen policy through a KMF handle

libmd(3LIB) – Message Digest Library

- small library for message digests
  - when PKCS#11 might seem too heavy
- `libmd5` was in Solaris before OpenSSL, `libmd` was a logical step forward
  - some other systems have the same API
- doesn't make use of CF
  - suitable for apps that need **digests only, quickly**, and don't process too much data
- ABI compatibility
- short source code example

Kerberos Enhancements

- Client Zero-Conf
  - DNS lookup KDC
  - heuristic to determine REALM
  - krb5.conf defaults
  - limited referral support
  - Nevada 71
- LDAP DB Support
  - pluggable db framework
  - DB2 plugin
  - multi-master KDC ?
  - Nevada 73

Kerberos Enhancements (cont'd)

- kclient Version 2
  - MS AD join
  - non-Solaris KDC support
  - dynamic clients
  - Nevada 91
- re-syncs with MIT Kerberos
  - 1.4.3 fully synced
  - major features from 1.5, 1.6 already introduced
  - 1.6.3 full-resync coming

Kerberos Enhancements (cont'd)

- PKINIT
  - initial authentication with x509 certificates
  - pre-authentication plugin
  - OpenSSL / KMF
- Kerberos Master-Key Encryption Type migration
  - single DES default
  - cannot be changed without destroying existing db
  - MIT Consortium (www.kerberos.org)

Validated Execution

- automatic integrity verification of all code **before execution**
  - kernel
  - libraries
  - utilities, including shell scripts, Java, Perl, etc.
- secure boot
- initial Solaris support for Trusted Computing Group (TCG) technologies
- disclaimer: early in project lifecycle, so details described here may change

Validated Execution - Key Elements

- signed manifests describing executables
  - > shipped with Solaris releases and patches
  - > can be generated by customers and ISV's
  - > manifest format defined by TCG specifications
- verifies executables upon `exec(2)`, `dlopen(2)`, `mmap(2)`, `modload(1)`
- uses Trusted Platfom Module (TPM) storage to validate initial modules during boot
- incorporates TPM as a keystore in the Solaris Cryptographic Framework

Validated Execution - Administration

- administrator control over
  - which signatures to accept
  - maximum privileges available to each executable set
- use `bart(1M)` to
  - generate custom manifests
  - verify filesystem contents against manifests
- key administration principle
  - simple configuration
  - predictable effects

Solaris Trusted Extensions (TX)

- a redesign of the Trusted Solaris product using a layered architecture
- an extension of the Solaris security foundation **providing access control policies based on the label of objects**
- a set of additional software packages added to a standard Solaris system
- a set of label-aware services which implement multilevel security

What is Labeling?

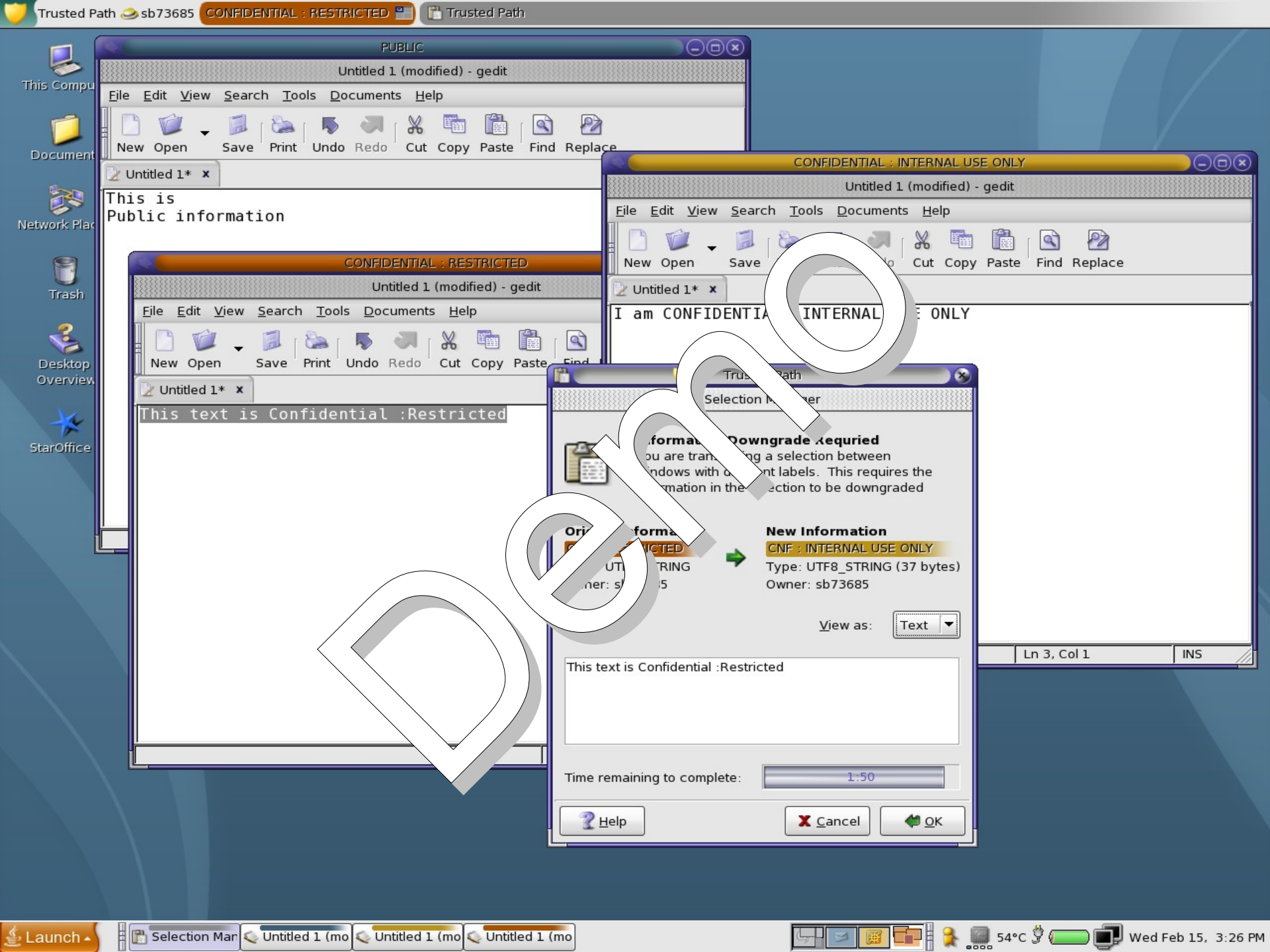
- **every object has a label associated with it**
  - files, windows, printers, devices, network packets, network interfaces, processes, etc...
- hierarchical or disjoint relationships
- accessing/sharing data is controlled by the objects' label relationship to each other
  - reading requires label dominance
    - reader's label ≥ objects label
  - writing requires label equality for the subject and object

Solaris Trusted Extensions (cont.)

- hooks included in Solaris 10 Update 3
  - > Layered product included in EA directory
  - > Free, just like Solaris
- **CC EAL 4+ for S10 11/06 Trusted Extensions issued in June 11, 2008**
- Sun Ray SRSS 4.0 (Bacchus) fully supports TX
- Nevada and S10u5 enable TX w/out separate install

Labeled Zones in Trusted Extensions

- each zone provides a security boundary
  - unique sensitivity label per zone
  - labels are implied by process zone IDs
  - processes and data are isolated by label
- no object is writable by more than one zone
  - mount policy prevents writing down or reading up
  - network policy requires endpoint label equality (default)
- information sharing between zones is based on label relationships


Trusted Path
sb73685
CONFIDENTIAL : RESTRICTED
Trusted Path
PUBLIC
Untitled 1 (modified) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste Find Replace
Untitled 1*
This is
Public information
CONFIDENTIAL : INTERNAL USE ONLY
Untitled 1 (modified) - gedit
File Edit View Search Tools Documents Help
New Open Save Cut Copy Paste Find Replace
Untitled 1*
I am CONFIDENTIA INTERNAL E ONLY
Ln 3, Col 1
INS
CONFIDENTIAL : RESTRICTED
Untitled 1 (modified) - gedit
File Edit View Search Tools Documents Help
New Open Save Print Undo Redo Cut Copy Paste
Untitled 1*
This text is Confidential :Restricted
Selection
Downgrade Requried
ng a selection between
nt labels. This requires the
ection to be downgraded
New Information
CNF : INTERNAL USE ONLY
Type: UTF8_STRING (37 bytes)
Owner: sb73685
View as:
Text
This text is Confidential :Restricted
Time remaining to complete:
1:50
Help
Cancel
OK
Demo
This Compu
Document
Network Plac
Trash
Desktop Overview
StarOffice
Launch
Selection Mar
Untitled 1 (mo
Untitled 1 (mo
Untitled 1 (mo
54°C
Wed Feb 15, 3:26 PM

What we didn't talk about...

- Secure By Default
- ipfilter enhancements
- IPsec enhancements
- IKE enhancements
- Virtualization Security
- ...and more

Design discussions

- **we're open!**
- security-discuss@opensolaris.org
- crypto-discuss@opensolaris.org
- kerberos-discuss@opensolaris.org
- kmf-discuss@opensolaris.org
- and lots of others...
  - http://mail.opensolaris.org/mailman/listinfo

References

- General Security
  - http://opensolaris.org/os/community/security/
- lofi encryption
  - http://www.opensolaris.org/os/project/loficc/
  - lofi(7D) manual page
- ZFS Crypto
  - http://opensolaris.org/os/project/zfs-crypto/
- KMF
  - http://opensolaris.org/os/project/kmf/
- Kerberos
  - http://opensolaris.org/os/project/kerberos/
- SunSSH
  - http://www.opensolaris.org/os/community/security/projects/SSH/

References (cont'd)

- Validated Execution
  - http://www.opensolaris.org/os/project/valex/
- PSARC case information
  - http://opensolaris.org/os/community/arc/
- OpenSSL engine(3) manual page
- libmd(3LIB) manual page
- Jan's blog
  - mostly on PKCS#11 engine and SunSSH
  - http://blogs.sun.com/janp/
- Mark's blog
  - Kerberos
  - http://blogs.sun.com/mbp/

Questions?

Jan Pechanec
Jan.Pechanec@Sun.COM

Mark Phalan
Mark.Phalan@Sun.COM