Solaris 10 Security

Darren J Moffat

Sr. Staff Engineer, Solaris Security

June 2005

Sun microsystems

Agenda

- Solaris's Overall Security Goals
- Strategic Investment Areas
  - Solaris 10 OS Features
  - Solaris 10 Networking
- How do things fit together
- Applies to AMD64, SPARC and X86

Solaris Security Goals

- Defending
  - Provide strong assurance of system integrity
  - Defend system from unauthorized access
- Enabling
  - Secure authentication of all active subjects
  - Protect communications between endpoints
- Deploying
  - Emphasize integratable stack architecture
  - Interoperable with other security architectures
  - Ease management and use of security features
  - Receive independent assessment of security

Solaris Hardening

GOAL: Defend system from unauthorized access, Provide high assurance of system integrity

Secure Deployment

Secure Network Install
Minimal Initial Install
Profile-based Install
Validated Execution
File Integrity Protection

Breach Containment

Minimal Process Privileges
Service Containment

Access Control

Role Based
User Based
File Based
Packet Based

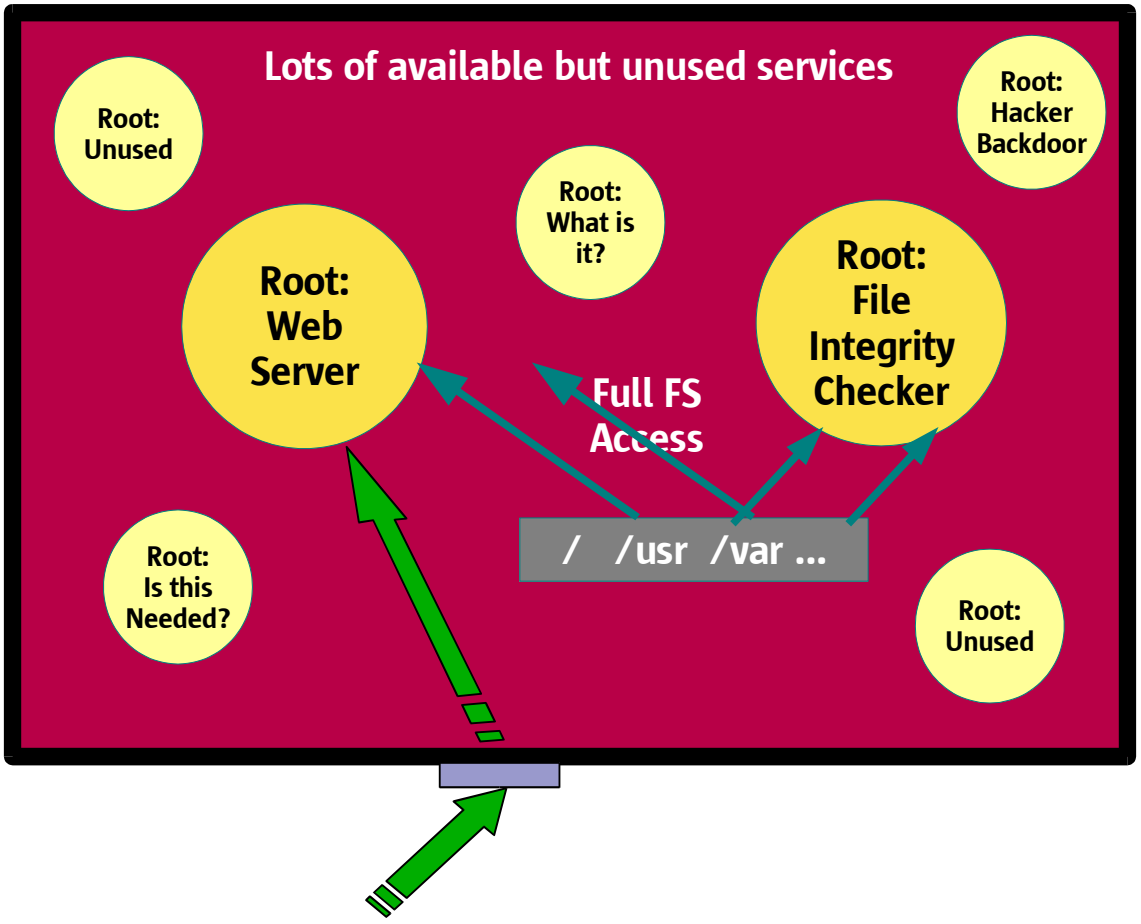
Auditing

Detailed Audit Trail
Centralized Logging
Enabling Intrusion Detection

Solaris 10 OS Security

- **Conservative Security Posture @Install**
  - Minimal install option (Reduced Networking)
  - More services *off or local only* after install
  - Service Manager for hardening
- **Privileges (Process Rights Mgmt)**
  - Decomposition of root privileges
  - Default for compatibility (root has all privs)
  - Privileges can be inherited or relinquished
- **Zones (Containers)**
  - Virtualization into application environments
  - Quarantine potentially risky software
  - Global Zone can see into other zones (IDS use)

Basic Web Server Deployment



Solaris Hardening Progress

Solaris 8

- Role-Based Access Controls
- Tightened file permissions

Solaris 9

- More Granular Packages
- Non-executable stack option
- Flexible Password Encryption
- SunScreen 3.2
- OS/Net - Non-exec stack

Solaris 10

- Secure Remote Install (WANboot)
- Service Management Framework
- Granular Process Privileges
- File Integrity Checker (BART)
- Zones
- Minimal Install Option
- Audit Enhancements
- Audit Logging w/syslog
- Stateful Packet Filtering

Least Privilege in Solaris 10

- Traditional UNIX is root or user
  - Kernel checks explicitly for uid = 0 or object owner
- CMW and later (expired) POSIX specifications on least privilege.
- Solaris 10 privileges evolution of 10+ years of common criteria evaluated implementation experience from Trusted Solaris.

Privilege Sets

- 48+ fine grained privileges instead of uid == 0
  - ppriv -lv : Shows privilege and what it protects.
- Each process has 4 privilege sets in its' kernel cred:
- Inheritable set (I)
  - The set of privileges child processes get on exec.
- Permitted set (P)
  - The maximum set of privileges for the process
- Effective set (E)
  - Subset of P that are currently asserted as needed by the process
- Limit set (L)
  - Upper bound a process and its children can obtain (takes effect on exec)

Current Privilege Names

“contract_event”	Process/Request critical/reliable events
“contract_observer”	Obsever events other than euid
"cpc_cpu”	Access to per-CPU perf counters
"dtrace_kernel"	DTrace kernel tracing
"dtrace_proc"	DTrace process-level tracing
"dtrace_user"	DTrace user-level tracing
"file_chown"	Change file's owner/group IDs
"file_chown_self"	Give away (chown) files
"file_dac_execute"	Override file's execute perms
"file_dac_read"	Override file's read perms
"file_dac_search"	Override dir's search perms
"file_dac_write"	Override (non-root) file's write perms
"file_link_any"	Create hard links to diff uid files
"file_owner"	Non-owner can do misc owner ops
"file_setid"	Set uid/gid (non-root) to diff id
"ipc_dac_read"	Override read on IPC, Shared Mem perms
"ipc_dac_write"	Override write on IPC, Shared Mem perms
"ipc_owner"	Override set perms/owner on IPC
"net_icmpaccess"	Send/Receive ICMP packets
"net_privaddr"	Bind to privilege port (<1023+extras)
"net_rawaccess”	Raw access to IP
"proc_audit”	Generate audit records
"proc_chroot”	Change root (chroot)
"proc_clock_highres"	Allow use of hi-res timers
"proc_exec"	Allow use of execve()
"proc_fork"	Allow use of fork*() calls
"proc_info"	Examine /proc of other processes
"proc_lock_memory"	Lock pages in physical memory
"proc_owner"	See/modify other process states
"proc_priocntl"	Increase priority/sched class
"proc_session"	Signal/trace other session process
"proc_setid"	Set process UID
"proc_taskid"	Assign new task ID
“proc_zone”	Signal/trace processes in other zones
“sys_acct”	Manage accounting system (acct)
“sys_admin	System admin tasks (node/domain name)
"sys_audit"	Control audit system
"sys_config"	Manage swap
"sys_devices"	Override device restricts (exclusive)
"sys_ipc_config"	Increase IPC queue
"sys_linkdir"	Link/unlink directories
"sys_mount"	Filesystem admin (mount,quota)
"sys_net_config"	Config net interfaces,routes,stack
"sys_nfs"	Bind NFS ports and use syscalls
"sys_res_config"	Admin processor sets, res pools
"sys_resource"	Modify res limits (rlimit)
"sys_suser_compat"	3rd party modules use of suser
"sys_time"	Change system time

Basic — Non-root privileges

Interesting — Some interesting privileges

Basic Privileges

- New for Solaris 10 are basic privileges.
  - Not in previous Trusted Solaris releases.
- These are things all normal users can normally do.
  - `proc_fork, proc_exec, proc_session, proc_info, file_link_any`
- Basic set will expand in future releases
- Dropping `proc_fork` and `proc_exec` from system daemons that should never fork or exec gives extra protection against buffer overflow exploits.
- Dropping `proc_info` means you can't even see other processes exist.

Viewing process privileges ppriv(1)

```
islay# ps -o pid,user,ruser,group,rgroup,comm -p `pgrep nfsd`
  PID     USER    RUSER    GROUP   RGROUP COMMAND
 1145   daemon   daemon   daemon   daemon /usr/lib/nfs/nfsd
islay# ppriv `pgrep nfsd`
1145:   /usr/lib/nfs/nfsd
flags = PRIV_AWARE
  E: basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_session,sys_nfs
  I: basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_session
  P: basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_session,sys_nfs
  L: basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_session
islay#
islay# ps -o pid,user,ruser,group,rgroup,comm -p `pgrep statd`
  PID     USER    RUSER    GROUP   RGROUP COMMAND
  245   daemon   daemon   daemon   daemon /usr/lib/nfs/statd
islay# ppriv `pgrep statd`
245:    /usr/lib/nfs/statd
flags = PRIV_AWARE
        E: basic,!file_link_any,!proc_exec,!proc_info,!proc_session
        I: basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_session
        P: basic,!file_link_any,!proc_exec,!proc_info,!proc_session
        L: basic,!file_link_any,!proc_exec,!proc_fork,!proc_info,!proc_session
islay#
```

What privileges do I need ?

Privilege "Debug" mode allows you to determine this:

```
$ ppriv -D $$
$ cat /etc/shadow
cat[3003]: missing privilege "file_dac_read" (euid =
35661, syscall = 225) needed at ufs_iaccess+0xd2
cat: cannot open /etc/shadow

$ cp /usr/sbin/ping /tmp
$ /tmp/ping jurassic
ping[3016]: missing privilege "net_icmpaccess" (euid = 35661,
syscall = 230) for "devpolicy" needed at so_socket+0xa7
/tmp/ping: socket Permission denied
```

RBAC & privileges

- RBAC profiles list the privileges the process will inherit when run.
- Examples:

- `Process Management:solaris:cmd:::/usr/bin/nice:privs=proc_owner,proc_priocntl`
- `Process Management:solaris:cmd:::/usr/bin/kill:privs=proc_owner`
- `File System Management:solaris:cmd:::/usr/sbin/umount:privs=sys_mount`
- `Network Management:solaris:cmd:::/usr/sbin/ifconfig:privs=sys_net_config`

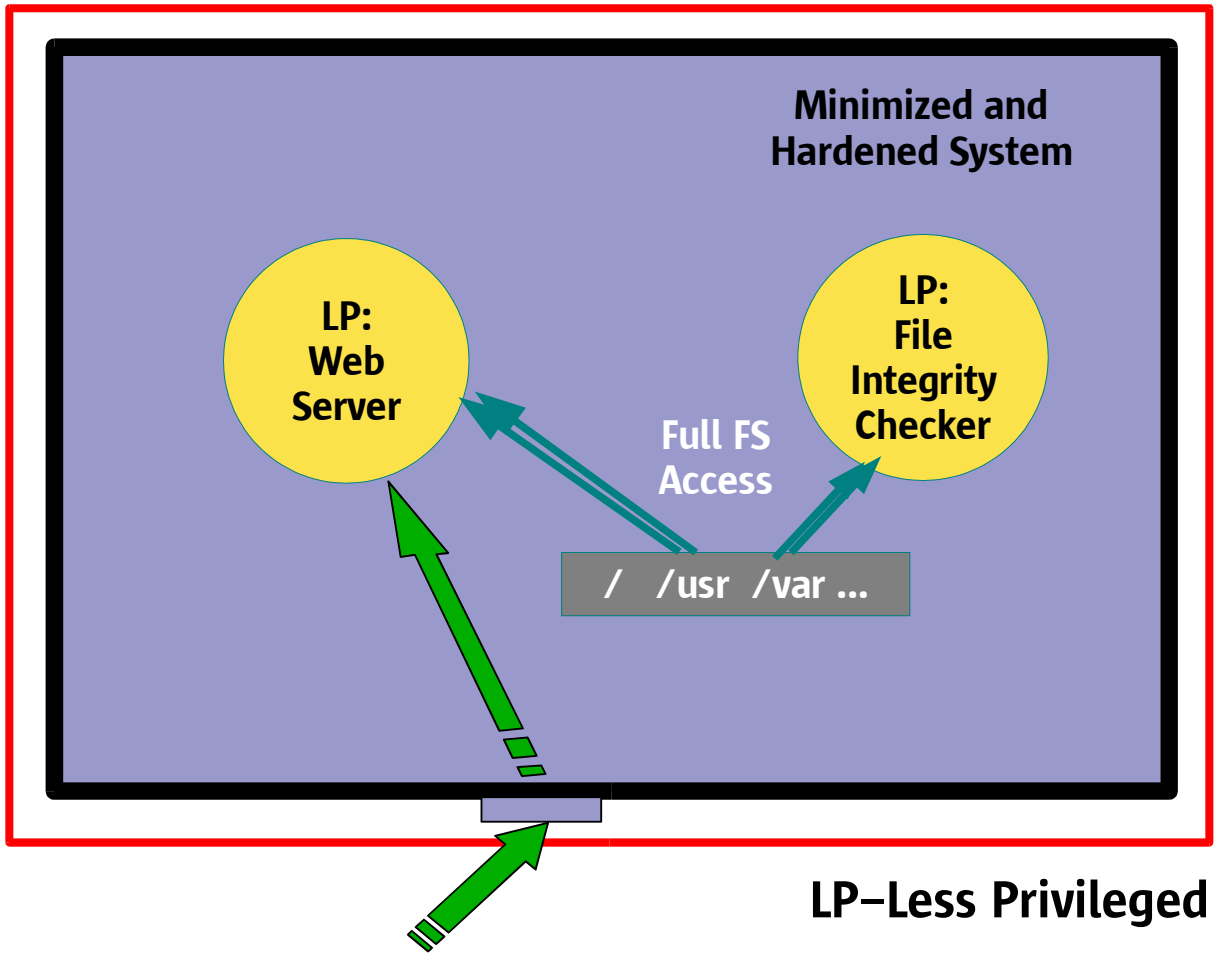
SMF – Service Management Framework

- SMF: Dependancy based service startup/recovery
- SMF service definitions (manifests) contain security attributes:
  - Assign uid/gid/default and limit privileges for services
  - Provide a Solaris RBAC authorization that is required to administer the service.
    - Restart the lp service if the user had the authorization:
    - $ svcadm restart svc://network/lp
  - Provide a Solaris RBAC authorization for reconfiguration of the service
- Provides distinction between configured/enabled
  - Service can be fully configured but disabled
  - enabled/disabled temporarily or permenantly

SMF Profiles

- Solaris 10 GA has two SMF profiles
  - generic.xml:
    - Most services enabled similar to Solaris 9.
  - generic_limited_net.xml
    - Fewer remotely available network services enabled
    - Only ssh for remote login, most other things off.
  - More coming in future releases
- Support for site profile(s) to enable/disable/configure services
  - Applied after system default (one of the above)

Solaris 10 Protected System
OS Hardening



Can be as simple as using SMF to start server

Zones (Solaris Containers)

- Multiple virtualized application environments from a single Solaris kernel
- Works on all Solaris platforms from 1 CPU upwards.
- Process containment
  - Resource usage & security isolation
- No direct access to hardware
- Zones appear as separate hosts from “outside”
- Allows for separate uid /gid namespace per zone
  - Each zone has their own root user.
  - Can be in different nameservice domains
- Separate file system space

Zone Security Properties

- Services can be isolated from each other
  - Quaranteening potentially risky software
  - Isolating multiple dis-trusting parties
  - Containing potential damage by a breach
- Global Zone can:
  - observe all activities inside each zone
  - not be seen by software in non global zone
  - change the contents or processes in each zone
  - house IDS that is undetectable and tamper-protected from zones
    - Including BSM Audit, BART/Tripwire
- Non-global Zones run with less privileges

Zones & Privileges

- Each Zone in Solaris has a subset of the available privileges.
  - Currently hardcoded: maybe configurable in future
- Zones don't have any of the system management privileges or the privileges for DTrace.
- Can only see processes in same Zone (except global zone)
- Processes in Zones can't send signals to other zones even if they do have proc_session or proc_owner
- Can't use shared memory between zones
- IPC possible, but needs “assist” from global zone or it is network based.

Zones are Less Privileged

“contract_event”	Process/Request critical/reliable events
“contract_observer”	Obsever events other than euid
"cpc_cpu”	Access to per-CPU perf counters
"dtrace_kernel"	DTrace kernel tracing
"dtrace_proc"	DTrace process-level tracing
"dtrace_user"	DTrace user-level tracing
"file_chown"	Change file's owner/group IDs
"file_chown_self"	Give away (chown) files
"file_dac_execute"	Override file's execute perms
"file_dac_read"	Override file's read perms
"file_dac_search"	Override dir's search perms
"file_dac_write"	Override (non-root) file's write perms
"file_link_any"	Create hard links to diff uid files
"file_owner"	Non-owner can do misc owner ops
"file_setid"	Set uid/gid (non-root) to diff id
"ipc_dac_read"	Override read on IPC, Shared Mem perms
"ipc_dac_write"	Override write on IPC, Shared Mem perms
"ipc_owner"	Override set perms/owner on IPC
"net_icmpaccess"	Send/Receive ICMP packets
"net_privaddr"	Bind to privilege port (<1023+extras)
"net_rawaccess”	Raw access to IP
"proc_audit”	Generate audit records
"proc_chroot”	Change root (chroot)
"proc_clock_highres"	Allow use of hi-res timers
"proc_exec"	Allow use of execve()
"proc_fork"	Allow use of fork*() calls
"proc_info"	Examine /proc of other processes
"proc_lock_memory"	Lock pages in physical memory
"proc_owner"	See/modify other process states
"proc_priocntl"	Increase priority/sched class
"proc_session"	Signal/trace other session process
"proc_setid"	Set process UID
"proc_taskid"	Assign new task ID
“proc_zone”	Signal/trace processes in other zones
“sys_acct”	Manage accounting system (acct)
“sys_admin	System admin tasks (node/domain name)
"sys_audit"	Control audit system
"sys_config"	Manage swap
"sys_devices"	Override device restricts (exclusive)
"sys_ipc_config"	Increase IPC queue
"sys_linkdir"	Link/unlink directories
"sys_mount"	Filesystem admin (mount,quota)
"sys_net_config"	Config net interfaces,routes,stack
"sys_nfs"	Bind NFS ports and use syscalls
"sys_res_config"	Admin processor sets, res pools
"sys_resource"	Modify res limits (rlimit)
"sys_suser_compat"	3rd party modules use of suser
"sys_time"	Change system time

Interesting	Some interesting privileges
Basic	Non-root privileges
Removed	Not available in Zones

Secure Network Communications

GOAL: Secure authentication of all subjects,
Protect communication between endpoints

- Strong User/Host Authentication
  - Single Network Sign-On
  - Mobile User Credentials
  - Network Identity
  - Public Key Technology
- Data Path Integrity
  - Digital Signatures and Hashes
- Private Communications
  - Encryption Technology
  - Secure Key Management
  - Encrypted Data communication
  - Virtual Private Networking

Secure Communication Progress

Solaris 8	
IPsec Support (AH, ESP) Smartcard Framework	Kerberos Protocol/Crypto GSS-API exposed
Solaris 9	
TCP Wrappers w/inetd support Kerberos Infrastructure (KDC) Kerberos Enhancement Bundled 128 bit cryptography JDK 1.4 (JGSS, J-Kerberos) Smartcard Middleware API Smartcard Terminal API	Restructured PAM modules LDAP protected by SSL Internet Key Exchange (IKE) /dev/random Secure Shell IKE Hardware Crypto (PKCS#11) IPsec Hardware Crypto.

Secure Communication Progress

Solaris 10

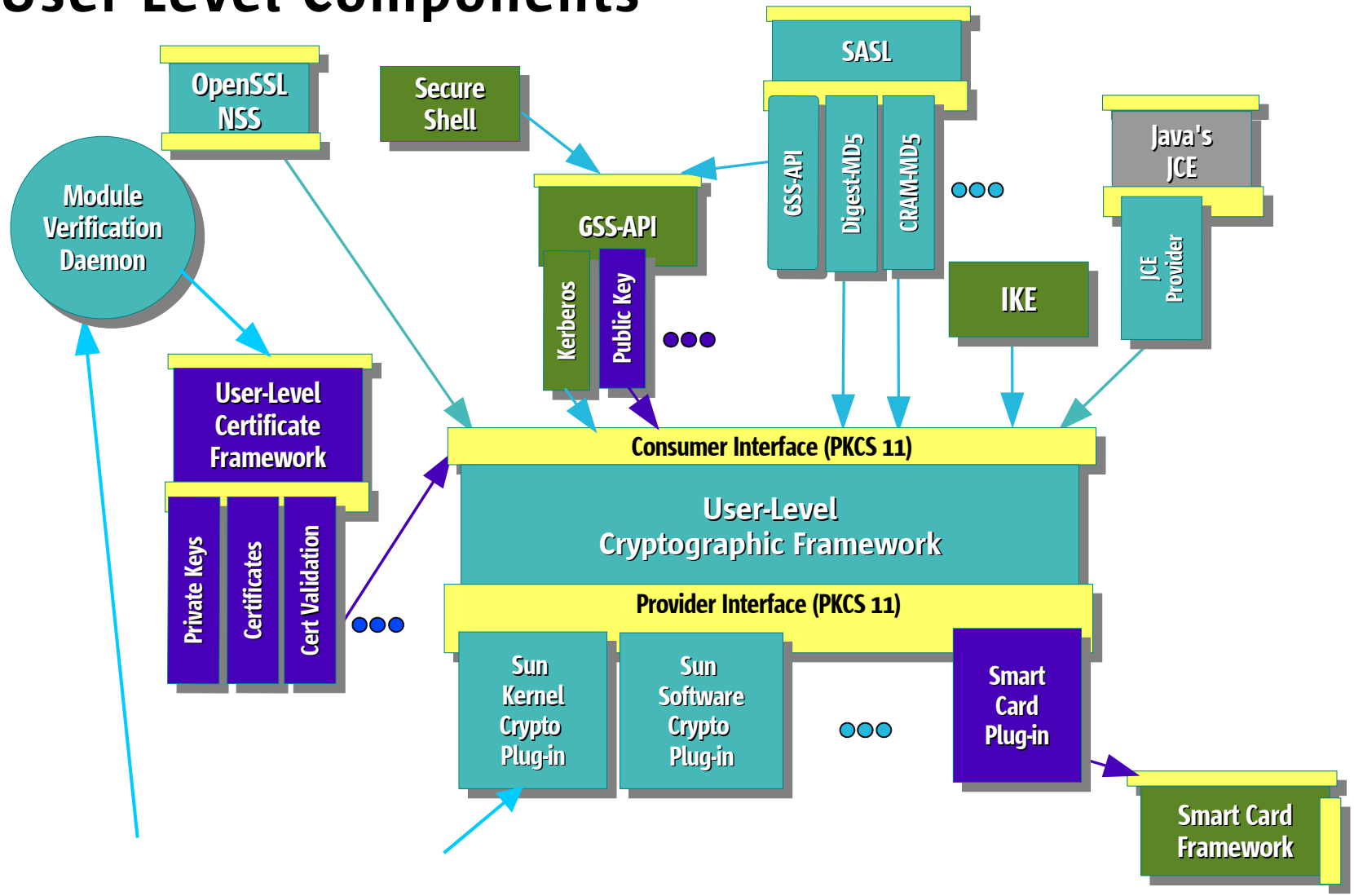
- User Crypto. Framework (uCF)
- Kernel Crypto. Framework (kCF)
- SASL Framework & Mechanisms
- Kerberos use of uCF and kCF
- Kerberos support for 3DES/AES
- PAM enhancements
- KDC Incremental Propagation
- Apache SSL
- Mozilla GSS/Kerberos
- Kerberized Applications
- SPNego (GSS Negotiation)
- IKE use of uCF
- IPsec use of kCF
- Java JCE use of uCF
- LDAP protected by Kerberos
- Secure Shell use of GSS-API
- Apache GSS

Password enhancements

- Failed login attempts can now lock account
  - Accounts can be marked as no lock
- Can now unlock an account preserving old password, passwd -u.
- Password history
- Improved control over password sanity checks
  - Including cracklib support
- Support for pluggable crypt(3c) interface
  - Supports Linux/BSD MD5 & Blowfish
  - Custom modules (eg UK government)

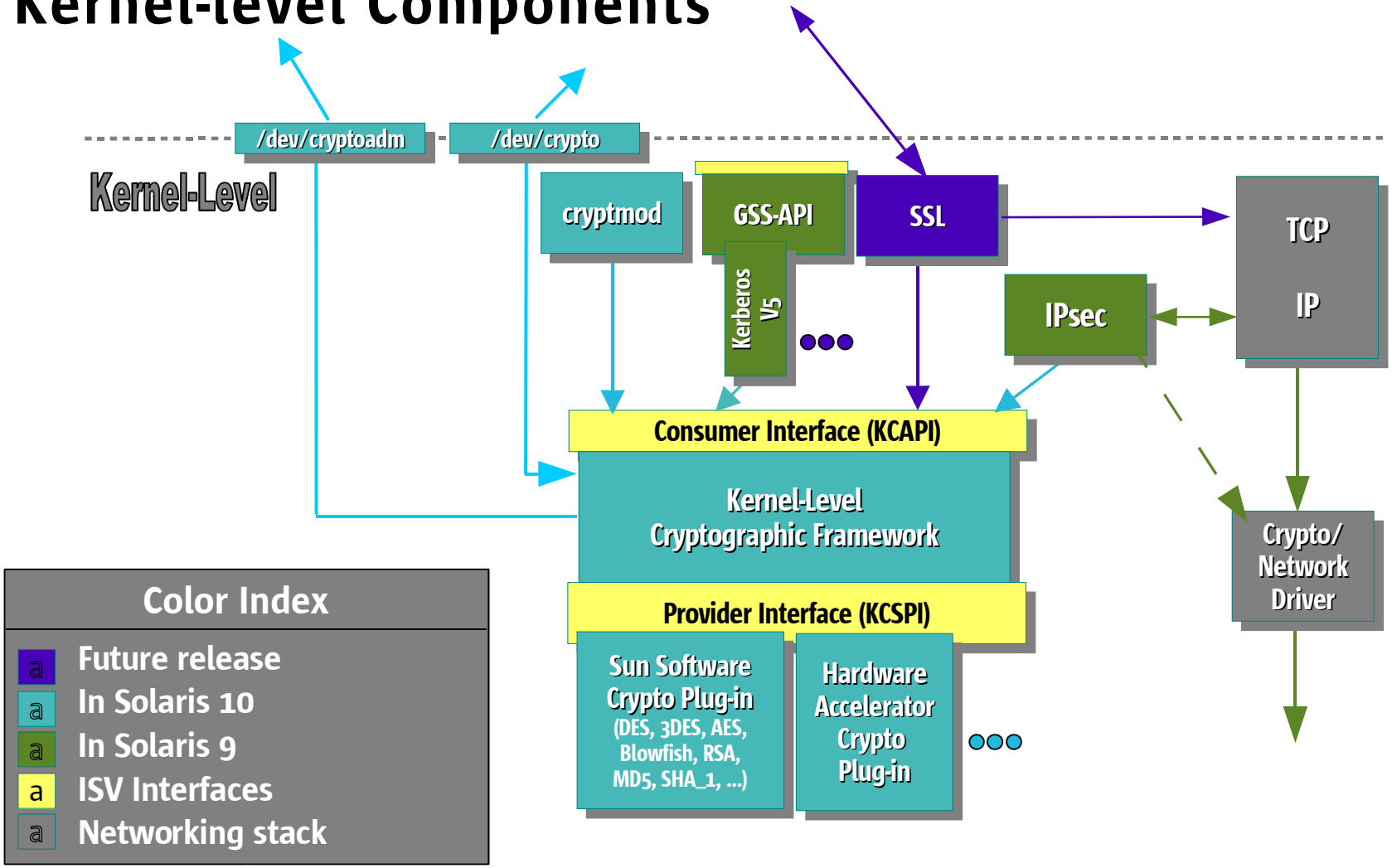
Network Security Architecture

User-Level Components



Network Security Architecture

Kernel-level Components

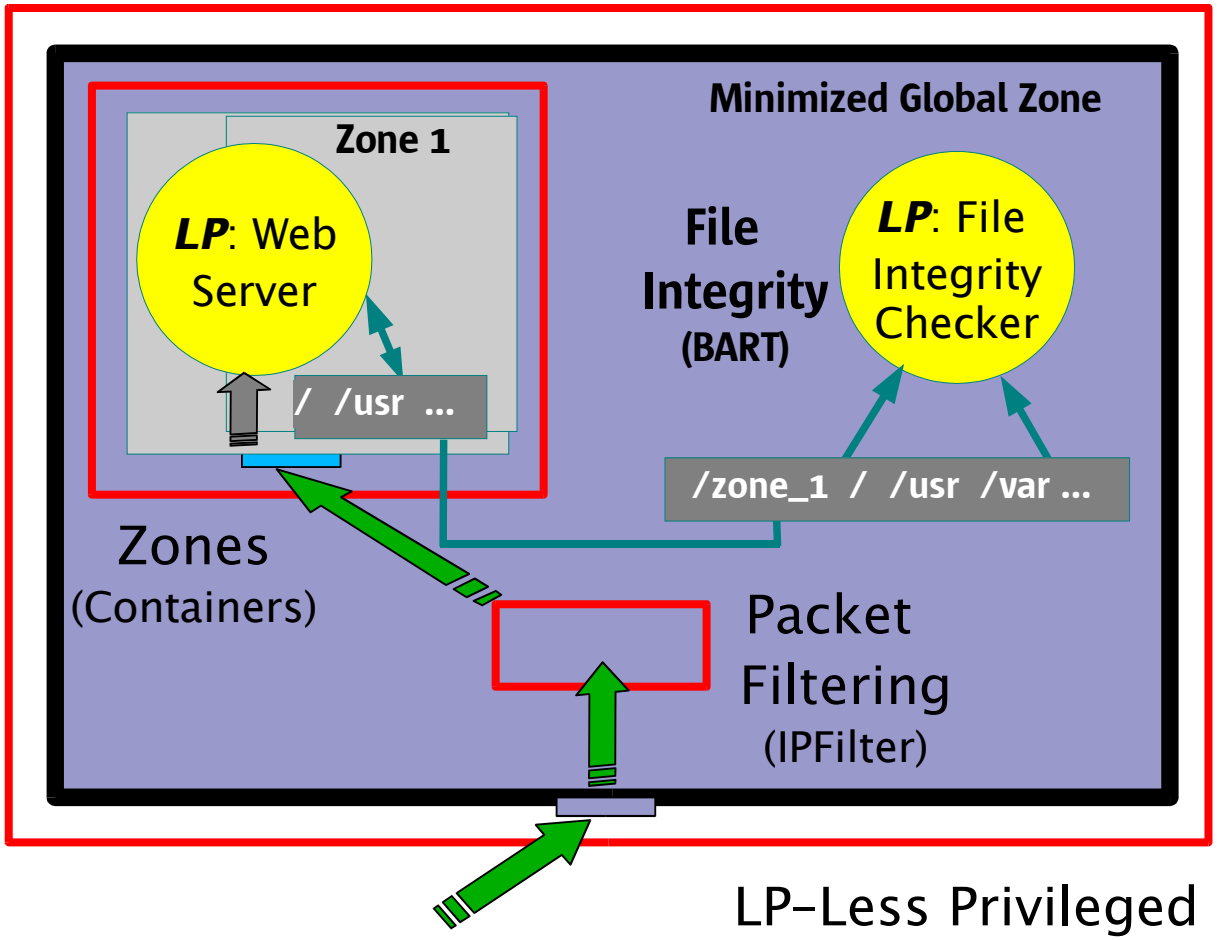


Solaris Packet Filter

- Based on Open Source IP Filter 4.x
- Stateful and stateless packet inspection

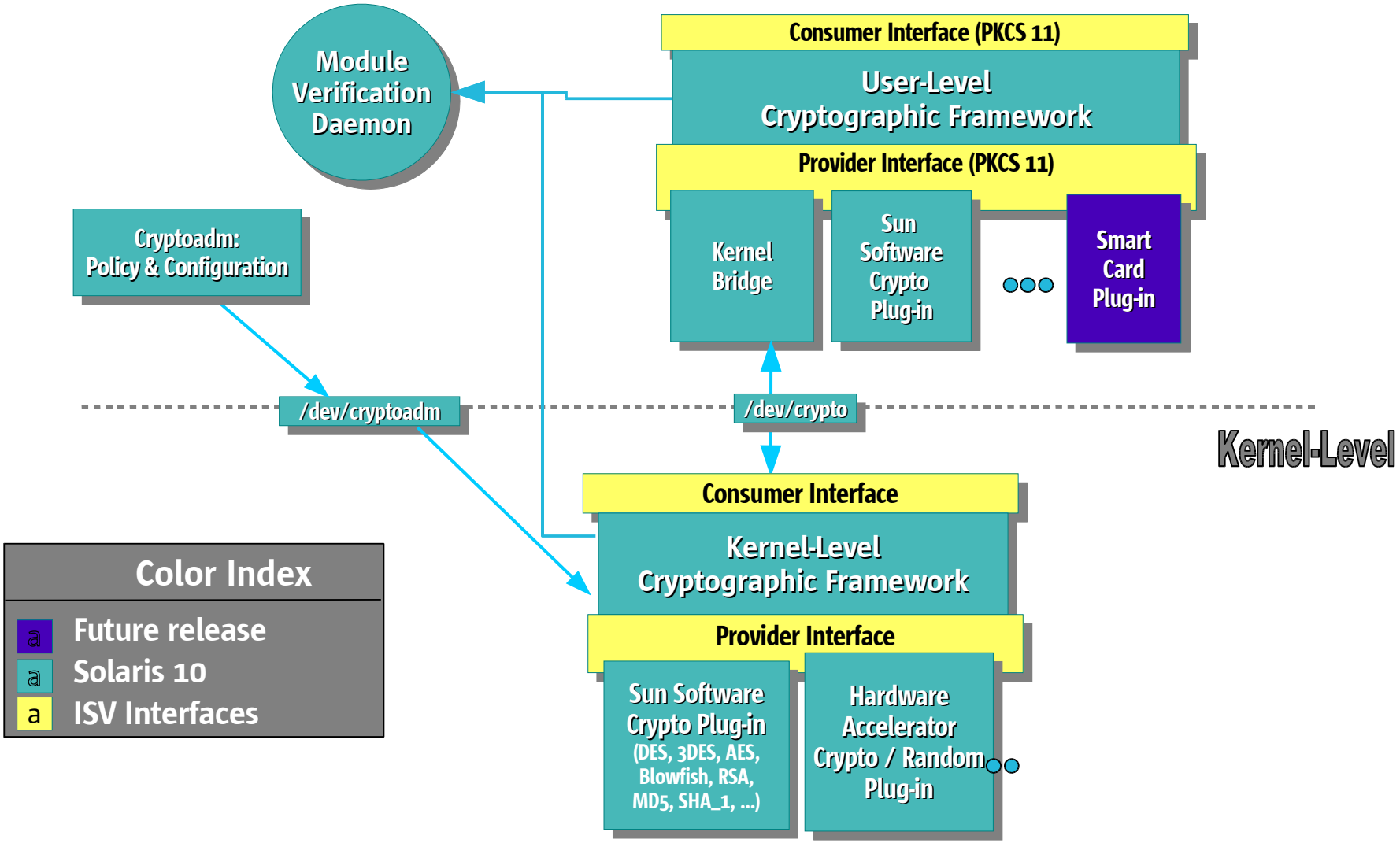
  Plumbs below IP module
- Text-based configuration (last match)
  - /etc/ipf/ipf.conf
  - /etc/ipf/ipnat.conf
  - Filter by: IP Addr (src,dst), Port, Interface, Direction, IPsec protection, CIPSO
  - Enforces: Block, pass or logging of packet
- Built in NAT and Port Address translation

Solaris 10 Protected System
Zones and BART



Can be as simple as running server in a zone and starting it with SMF

Cryptographic Framework



Crypto Framework Features

- Standards based pluggable framework
- Userland (PKCS#11) & kernel
- Administrative policy [cryptoadm(1m)]
  - provider/cryptographic algorithm,
    - eg enable only FIPS 140 algorithms
- End user commands
  - encrypt(1),digest(1),mac(1), pktool(1)
- Java JCE default provider is PKCS#11
- OpenSSL ENGINE for PKCS#11
  - Apache mod_ssl uses this by default

Bundled Userland Providers

- pkcs11_softtoken.so.1
  - Default PKCS#11 v2.11 software provider
  - DES, 3DES, AES, RC4 (<=128 bit); RSA, DSA, D-H, MD5, SHA-1, SSL HMAC
  - On disk (encrypted) persistant keystore
- pkcs11_softtoken_extra.so.1
  - Supports symmetric algorithms > 128-bit keys
  - Delivered via Encryption Kit in SUNWcry package
- Softtoken supports:
  - Asymmetric algorithms for signing & verification
  - Object and key management

Crypto Framework Consumers

- Kerberos, both kernel & userland code have been ported to the cryptographic framework.
- Userland performance numbers :

Krb5i	Krb5p
77.00%	54.00%
85.00%	53.00%
76.00%	50.00%
90.00%	56.00%

ftp put

	Krb5i	Krb5p
10 MB	87.00%	56.00%
20 MB	93.00%	58.00%
40 MB	91.00%	59.00%
100 MB	93.00%	56.00%

- ikrb = kerberos w/integrity checks using MD5 HMAC
- pkrb = kerberos with 3DES for privacy & MD5 HMAC

Crypto Framework Consumers

- Kernel performance numbers:

NFS Read

	krb5	krb5i	krb5p
10 MB	72.00%	63.00%	80.00%
20 MB	71.00%	62.00%	79.00%
40 MB	71.00%	63.00%	81.00%
100 MB	72.00%	61.00%	78.00%

NFS Write

	krb5	krb5i	krb5p
10 MB	80.00%	57.00%	72.00%
20 MB	77.00%	64.00%	75.00%
40 MB	74.00%	60.00%	75.00%
100 MB	77.00%	64.00%	77.00%

Crypto Framework Consumers

- IPsec
  - Uses kcf interfaces for IPsec AH and ESP
  - IPsec performance numbers ([Mb/s], TCP_STREAM between two SB1000):

	Stock IPsec	IPsec/kEF	Diff
esp-aes/none:TCP_STREAM	67.45	94.29	40%
esp-aes/md5:TCP_STREAM	54.72	71.02	30%
esp-3des/none:TCP_STREAM	20.21	37.83	87%
esp-3des/md5:TCP_STREAM	19.09	34.00	78%
esp-blowfish/none:TCP_STREAM	59.98	70.73	18%
esp-blowfish/md5:TCP_STREAM	51.02	57.49	13%
esp-none/md5:TCP_STREAM	143.23	146.03	2%
ah-md5:TCP_STREAM	132.89	132.15	-1%

Future Crypto Support

- SHA2
  - In OpenSolaris now, update of Solaris 10
- ECC support
  - Implemented but not yet shipped due to potential legal issues.
  - Support for Mozilla, Sun Java System WebServer
- FIPS 140-2 Evaluation
  - Not yet, is this important for you for software only @ level 2 ?

Kerberos Evolution

- Bundled Kerberos-aware applications
  - Telnet, ftp, rsh, rlogin, rdist, KDC
  - Mozilla, Apache, Secure Shell (via GSS-API)
- Enhanced interoperability and security
  - TCP and IPv6 Support
  - AES-128, AES-256, 3DES, RC4-HMAC
- Ease of deployment
  - *kclient* automated system setup
  - *pam_krb5_migrate* automated KDC population
- Incremental KDC DB propagation

Summary

- Role Based Access Control (RBAC)
- Process Least Privilege
- Zones (Solaris Containers)
- Packet Filter
- Service Management Framework
- Password enhancements
- Cryptographic Framework
- Kerberos enchancements

Questions?

Resources:

http://sun.com/solaris/security.jsp

Sun microsystems

Darren J Moffat

darren.moffat@sun.com

http://blogs.sun.com/darren

Sun microsystems