Enhancing Security Awareness and Control with DTrace

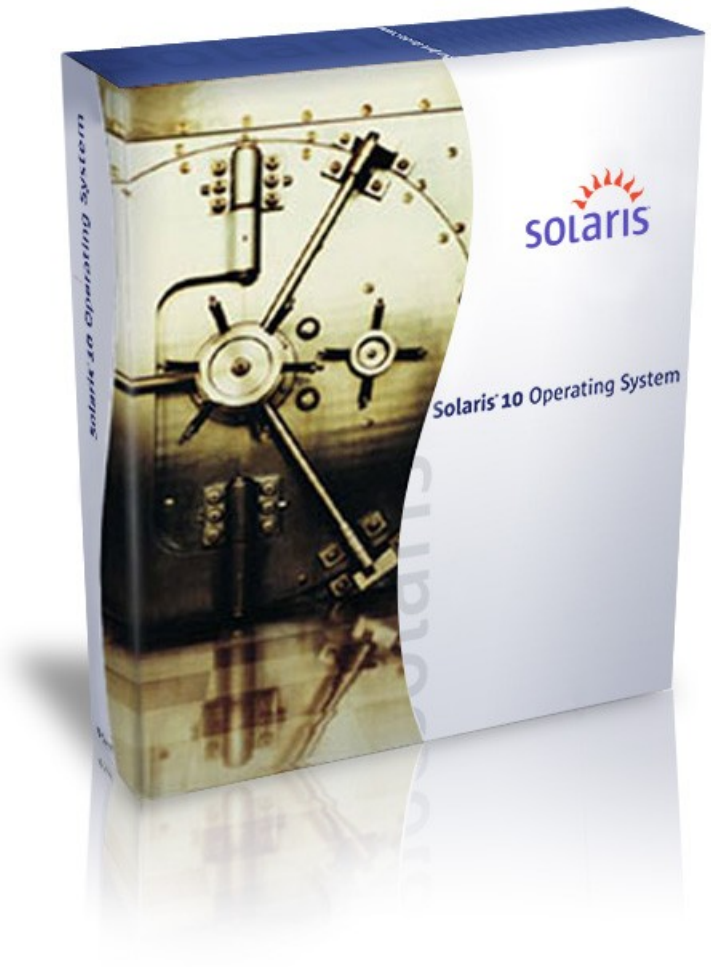
Glenn Brunette and Jon Haslam

Sun Microsystems, Inc.

Agenda

- Solaris 10 Security Controls
- Solaris Auditing
- DTrace && Security
- DTrace Security Examples
- DTrace Security Integration
- DTrace Security Automation
- References

Solaris 10 Security Controls



Reduced OS Installation Profiles

"Secure by Default"

User Rights Management

Process Rights Management

PAM, SMF, Crypto. Framework

Zones, ZFS, ACLs

IPFilter, TCP Wrappers

SSH, Kerberos, IPsec

Trusted Extensions

BART, sfpDB, Solaris Auditing

Solaris Auditing (In a Nutshell)

- Implements in-kernel detection and logging of administrative actions, user events and system calls.
- Groups individual audit events into “classes” that can be audited for success, failure or both.
- Supports both global and per-user audit policies as well as “global” or “per-zone” audit policies.
- Generates binary audit trail files that can be converted to both text or XML and can also deliver audited events to SYSLOG.

Solaris Auditing Example

```
$ auditreduce -m AUE_su -r joe | praudit -s
  file,2005-04-12 07:25:06.000 -04:00,
  header,97,2,AUE_su,,10.8.31.9,2005-04-12 07:28:30.220
  -04:00
  subject,joe,joe,other,joe,other,1834,3097759606,12114
  22 10.9.1.3
  text,bad auth. for user roleB
  return,failure,2
```

Example taken from the Sun BluePrint: Enforcing the Two-Person Rule Via Role-based Access Control in the Solaris 10 OS, http://www.sun.com/blueprints/0805/819-3164.pdf

Solaris Auditing Challenges

- Availability of good analysis and reporting tools.
  - Commercial (few and expensive) or “roll your own”
- Level of granularity with respect to audited events.
  - Global or per-user configurations
  - Audited events are all or nothing
- Reboot required to (initially) enable Solaris auditing.
- Failure to properly define an audit context can break things or cause events to not be reflected in audit trails.
  - e.g., OpenSSH versus `cron`

DTrace to the rescue!

- DTrace has great properties for monitoring security events:
  - Everything that happens on a system can be observed.
  - Monitoring configuration can be dynamically yet safely selected.
  - Rich language and API is used to define what should be monitored.
  - Level of granularity is often limited only by your imagination.
- DTrace is not for the faint of heart:
  - Requires intimate knowledge of implementation details.
  - DTrace will often require wrappers and/or post-processing.
- but, DTrace does provide functionality otherwise not available!

DTrace Destructive Actions

- DTrace was designed as an observation tool.
  - It was never meant for active system control.
- DTrace does support destructive operations:
  - Requires use of the `-w` option.
  - `stop`, `raise`, `breakpoint`, `panic`, `chill`, etc.
- DTrace does support indirect actions as well:
  - `system("...")`
- WARNING: `system()` action is not synchronous w.r.t enabling!
  - Processed at user level and bounded by the `switchrate` tunable
  - It is therefore not reliable enough for security control.

DTrace Security

- DTrace requires the use of privileges:
  - **`dtrace_kernel`** Allows DTrace kernel-level tracing.
  - **`dtrace_proc`** Allows DTrace process-level tracing. Allows process-level tracing probes to be placed and enabled in processes to which the user has permissions.
  - **`dtrace_user`** Allows DTrace user-level tracing. Allows use of the syscall and profile DTrace providers to examine processes to which the user has permissions.
- DTrace can optionally be used in a Solaris zone:
  - Solaris 10 11/06 or newer using the **`limitpriv`** keyword in `zonecfg(1M)`. **`dtrace_kernel`** is prohibited however.

DTrace Security Monitoring Examples

Caveats and Warnings

The following slides illustrate how a variety of common security monitoring problems can potentially be solved using DTrace.

All of the solutions and code that will be discussed are representative samples only and not in any way guaranteed to be complete or customer-ready.

Your mileage may vary. Void where prohibited by law. Blah blah blah...

DTrace Security Example #1

Solaris Privilege Profiling

- What and why?
  - Understand what processes are asserting what privileges.
  - Develop baselines to limit privileges used by services.
    - Using a baseline, privileges can be limited with RBAC, SMF, etc.
- How?
  - `sdt:::priv-ok` Privilege is successfully asserted.
  - `sdt:::priv-err` Privilege is not successfully asserted.
- Caveats and restrictions?
  - Global Zone only.
- Special thanks:
  - Darren Moffat

Solaris Privilege Profiling

```
web_svc zone:   # svcadm disable apache2
global zone:    # privdebug.pl -v -f -n httpd
web_svc zone:   # svcadm enable apache2
global zone:[output of privdebug command]
```

STAT	TIMESTAMP	PPID	PID	PRIV	CMD
USED	273414882013890	4642	4647	net_privaddr	httpd
USED	273415726182812	4642	4647	proc_fork	httpd
USED	273416683669622	1	4648	proc_fork	httpd
USED	273416689205882	1	4648	proc_fork	httpd
USED	273416694002223	1	4648	proc_fork	httpd
USED	273416698814788	1	4648	proc_fork	httpd
USED	273416703377226	1	4648	proc_fork	httpd

privdebug.pl is available from the OpenSolaris Security Community:
http://www.opensolaris.org/os/community/security/

DTrace Example #1
NTP Privilege Profiling

DTrace Security Example #2

Keystroke Logging

- Background:
  - Neither Solaris Auditing nor RBAC provide a way to log keystrokes.
  - Privileged users may require greater amounts of oversight.
- How?
  - `syscall::read:{entry|return}`
  - `syscall::write:{entry|return}`
- Caveats and restrictions?
  - None
- Special thanks:
  - Brendan Gregg (e.g., `shellsnoop`)

DTrace Example #2
Keystroke Logging

DTrace Security Example #3

Specific File Access Detection

- Background:
  - Solaris auditing will audit every file and directory access matching a given audit event (e.g., read, write, execute, etc.)
  - There is no way to specify which files you are interested in.
- How?
  - `syscall::open:{entry|return}`
  - `syscall::open64:{entry|return}`
- Caveats and restrictions?
  - None
- Special thanks:
  - Brendan Gregg (e.g., `opensnoop`)

DTrace Example #3
File Access Detection

DTrace Security Example #4

Auditing Specific Group Actions

- Background:
  - Solaris auditing can only monitor on a global or per-user basis.
  - Specific groups of users may warrant additional monitoring.
- How?
  - `fbt:genunix:execsetid:{entry|return}`
- Caveats and restrictions?
  - Global Zone only.

DTrace Example #4
Audit Group Actions

DTrace Security Example #5

Detect Outbound Socket Connections

- Background:
  - Some services should never need to initiate outbound network connections. There is no way to specifically target these services using Solaris auditing.
- How?
  - `fbt:ip:tcp_connect:entry`
- Caveats and restrictions?
  - Global Zone only.

DTrace Example #5
Outbound Socket Connections

DTrace Security Example #6

Detect Promiscuous Mode Transitions

- Background:
  - Attackers often enable network sniffers after compromising a system in order to collect passwords and other data that could be useful for theft or for staging follow-up attacks.
- How?
  - `fbt::mac_promisc_set:entry`
- Caveats and restrictions?
  - Global Zone only.
- Special thanks:
  - Efi Batchev

DTrace Example #6
Promiscuous Mode

Automating and Integrating

- Automating DTrace Security Monitoring
  - SMF! (e.g., Solaris Secure by Default)
- Automating Log File Rotation
  - `logadm(1M)`
- Integrating with Security Event Management Platforms
  - Redirect DTrace output to a file:
    - `freopen("/path/to/file") / ftruncate()`
    - `printf("...")`
  - Send DTrace messages to SYSLOG:
    - `system("/usr/bin/logger ...")`

Summary

1. DTrace is an incredibly powerful tool for observing security events on Solaris systems.
2. DTrace complements Solaris Auditing providing a way to observe more targeted security events.
3. All the cool kids are using DTrace. You should too! Let us know what ideas you come up with!

For More Information

- Sun Security Home
  - http://www.sun.com/security
- OpenSolaris Security Community
  - http://www.opensolaris.org/os/community/security
- OpenSolaris DTrace Community
  - http://www.opensolaris.org/os/community/dtrace
- Solaris Internals – Mauro & McDougall
  - http://www.solarisinternals.com
- Brendan Gregg's DTrace Tools
  - http://www.brendangregg.com/dtrace.html

Enhancing Security Awareness and Control with DTrace

Glenn Brunette
glenn.brunette@sun.com
http://blogs.sun.com/gbrunett

Jon Haslam
jonathan.haslam@sun.com
http://blogs.sun.com/jonh