



Security

---

wangth

# FreeBSD Security Advisories

---

❑ <http://www.freebsd.org/security/advisories.html>

## FreeBSD Security Advisories

This web page contains a list of released FreeBSD Security Advisories. See the [FreeBSD Security Information](#) page for general security information about FreeBSD.

Issues affecting the FreeBSD Ports Collection are covered in [the FreeBSD VuXML document](#).

Date	Advisory name
2019-11-12	<a href="#">FreeBSD-SA-19:26.mcu</a>
2019-11-12	<a href="#">FreeBSD-SA-19:25.mcepsec</a>

# FreeBSD Security Advisories

## ❑ Advisory

- Security information

## ❑ Where to find it

- Web page (Security Advisories Channel)

➤ <http://www.freebsd.org>



The screenshot shows the FreeBSD Project website. The 'SECURITY ADVISORIES' section is circled in green and contains the following information:

Date	Advisory Title
2006-12-06	<a href="#">FreeBSD-SA-06:26.gtar</a>
2006-12-06	<a href="#">FreeBSD-SA-06:25.kmem</a>
2006-11-09	<a href="#">FreeBSD-SA-06:24.libarchive</a>

# FreeBSD Security Advisories

## ❑ Where to find it

- freebsd-security-notifications Mailing list
  - <http://lists.freebsd.org/mailman/listinfo/freebsd-security-notifications>

### Subscribing to freebsd-security-notifications

Subscribe to freebsd-security-notifications by filling out the following form. You will be sent email requesting confirmation, to prevent others from gratuitously subscribing you. This is a hidden list, which means that the list of members is available only to the list administrator.

Your email address:

Your name (optional):

You may enter a privacy password below. This provides only mild security, but should prevent others from messing with your subscription. **Do not use a valuable password** as it will occasionally be emailed back to you in cleartext.

If you choose not to enter a password, one will be automatically generated for you, and it will be sent to you once you've confirmed your subscription. You can always request a mail-back of your password when you edit your personal options.

Pick a password:

Reenter password to confirm:

Which language do you prefer to display your messages?

English (USA)

Would you like to receive list mail batched in a daily digest?

No  Yes

Subscribe

# FreeBSD Security Advisories

## ❑ Example

- mcephsc

```
-----  
FreeBSD-SA-19:25.mcephsc                               Security Advisory  
                                                         The FreeBSD Project
```

```
Topic:          Machine Check Exception on Page Size Change
```

```
Category:       core
```

```
Module:         kernel
```

```
Announced:     2019-11-12
```

```
Credits:        Intel
```

```
Affects:        All supported versions of FreeBSD.
```

```
Corrected:      2019-11-12 18:03:26 UTC (stable/12, 12.1-STABLE)  
                2019-11-12 18:13:04 UTC (releng/12.1, 12.1-RELEASE-p1)  
                2019-11-12 18:13:04 UTC (releng/12.0, 12.0-RELEASE-p12)  
                2019-11-12 18:04:28 UTC (stable/11, 11.3-STABLE)  
                2019-11-12 18:13:04 UTC (releng/11.3, 11.3-RELEASE-p5)
```

```
CVE Name:       CVE-2018-12207
```

# FreeBSD Security Advisories

---

## ❑ CVE-2018-12207

- <https://nvd.nist.gov/vuln/detail/CVE-2018-12207>

## 🚩 CVE-2018-12207 Detail

### MODIFIED

---

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

## Current Description

Improper invalidation for page table updates by a virtual guest operating system for multiple Intel(R) Processors may allow an authenticated user to potentially enable denial of service of the host system via local access.

**Source:** MITRE

# FreeBSD Security Advisories

---

## ❑ Example

- Problem Description

### I. Background

The Intel machine check architecture is a mechanism to detect and report hardware errors, such as system bus errors, ECC errors, parity errors, and others. This allows the processor to signal the detection of a machine check error to the operating system.

### II. Problem Description

Intel discovered a previously published erratum on some Intel platforms can be exploited by malicious software to potentially cause a denial of service by triggering a machine check that will crash or hang the system.

### III. Impact

Malicious guest operating systems may be able to crash the host.

# FreeBSD Security Advisories

---

## ❑ Example

- Workaround

### IV. Workaround

No workaround is available. Systems not running untrusted guest virtual machines are not impacted.



# FreeBSD Security Advisories

## ❑ Example

- Solution
  - Upgrade to
  - Source code patch
  - Binary patch

### V. Solution

Upgrade your vulnerable system to a supported FreeBSD stable or release / security branch (releng) dated after the correction date, and reboot.

Perform one of the following:

1) To update your vulnerable system via a binary patch:

Systems running a RELEASE version of FreeBSD on the i386 or amd64 platforms can be updated via the `freebsd-update(8)` utility:

```
# freebsd-update fetch
# freebsd-update install
# shutdown -r +10min "Rebooting for a security update"
```

2) To update your vulnerable system via a source code patch:

The following patches have been verified to apply to the applicable FreeBSD release branches.

a) Download the relevant patch from the location below, and verify the detached PGP signature using your PGP utility.

```
[FreeBSD 12.1]
# fetch https://security.FreeBSD.org/patches/SA-19:25/mcepsc.12.1.patch
# fetch https://security.FreeBSD.org/patches/SA-19:25/mcepsc.12.1.patch.asc
# gpg --verify mcepsc.12.1.patch.asc
```

```
[FreeBSD 12.0]
# fetch https://security.FreeBSD.org/patches/SA-19:25/mcepsc.12.0.patch
# fetch https://security.FreeBSD.org/patches/SA-19:25/mcepsc.12.0.patch.asc
# gpg --verify mcepsc.12.0.patch.asc
```

```
[FreeBSD 11.3]
# fetch https://security.FreeBSD.org/patches/SA-19:25/mcepsc.11.patch
# fetch https://security.FreeBSD.org/patches/SA-19:25/mcepsc.11.patch.asc
# gpg --verify mcepsc.11.patch.asc
```

b) Apply the patch. Execute the following commands as root:

```
# cd /usr/src
# patch < /path/to/patch
```

c) Recompile your kernel as described in [URL:https://www.FreeBSD.org/handbook/kernelconfig.html](https://www.FreeBSD.org/handbook/kernelconfig.html) and reboot the system.

# Common Security Problems

---

## ❑ Software bugs

- FreeBSD security advisor
- pkg audit
  - pkg-audit(8)

## ❑ Unreliable wetware

- Phishing site

## ❑ Open doors

- Account password
- Disk share with the world

# pkg audit (1)

---

## ❑ pkg audit

- Checks installed ports against a list of security vulnerabilities
- pkg audit -F
  - -F: Fetch the current database from the FreeBSD servers.

## ❑ Security Output

# pkg audit (2)

## ❑ pkg audit -F

```
Fetching vuln.xml.bz2: 100% 694 KiB 710.2kB/s 00:01
libxml2-2.9.4 is vulnerable:
libxml2 -- Multiple Issues
CVE: CVE-2017-9050
CVE: CVE-2017-9049
CVE: CVE-2017-9048
CVE: CVE-2017-9047
CVE: CVE-2017-8872
WWW: https://vuxml.FreeBSD.org/freebsd/76e59f55-4f7a-4887-bcb0-11604004163a.html

1 problem(s) in the installed packages found.
```

## ❑ <http://www.freshports.org/<category>/<portname>>

- <https://www.freshports.org/databases/postgresql96-server/>

# pkg audit (3)



We also have a status page: <https://freshports.wordpress.com/>

## Port details

**postgresql96-server** PostgreSQL is the most advanced open-source database available anywhere

9.6.6 [databases](#)  $\Sigma=5$  🔍 🦠 🦠 🦠

Maintainer: [pgsql@FreeBSD.org](mailto:pgsql@FreeBSD.org) 🔍

Port Added: 05 Sep 2016 11:15:47

License: PostgreSQL

PostgreSQL is a sophisticated Object-Relational DBMS, supporting almost all SQL constructs, including subselects, transactions, and user-defined types and functions. It is the most advanced open-source database available anywhere. Commercial Support is also available.

The original Postgres code was the effort of many graduate students, undergraduate students, and staff programmers working under the direction of

# Common trick

---

## ❑ Tricks

- ssh scan and hack
  - ssh guard
  - sshit
  - ...
- Phishing
- XSS & SQL injection
- ...

## ❑ Objective

- Spam
- Jump gateway
- File sharing
- ...

# Process file system - procfs

```
last pid: 8103; load averages: 0.00, 0.03, 0.04
102 processes: 1 starting, 1 running, 100 sleeping
CPU states: 0.2% user, 0.0% nice, 1.7% system, 0.7% interrupt, 97.4% idle
Mem: 305M Active, 1402M Inact, 215M Wired, 81M Cache, 112M Buf, 3016K Free
Swap: 4096M Total, 352K Used, 4096M Free
```

PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	C	TIME	WCPU	COMMAND
4576	tyhsieh	1	76	0	1964K	1652K	select	1	56:05	0.00%	httpd
4566	tyhsieh	1	76	0	1672K	1360K	select	0	6:13	0.00%	httpd
4584	tyhsieh	1	76	0	1996K	1052K	select	0	1:24	0.00%	httpd

## ❑ Procfs

- A view of the system process table
- Normally mount on /proc
- `mount -t procfs proc /proc`

```
hsec[/proc/4566] -chiahung- ls -al
total 0
dr-xr-xr-x  1 tyhsieh  hsec   0 Jan  3 13:53 ./
dr-xr-xr-x  1 root    wheel  0 Jan  3 13:53 ../
-r--r--r--  1 tyhsieh  hsec   0 Jan  3 13:53 cmdline
--w-----  1 tyhsieh  hsec   0 Jan  3 13:53 ctl
-r--r--r--  1 tyhsieh  hsec   0 Jan  3 13:53 etype
lr--r--r--  1 tyhsieh  hsec   0 Jan  3 13:53 file@ -> /home/tyhsieh/.etcdir/.etcvar/.etcexec/.etcvar/httpd
-r--r--r--  1 tyhsieh  hsec   0 Jan  3 13:53 map
-r--r--r--  1 tyhsieh  hsec   0 Jan  3 13:53 rlimit
-r--r--r--  1 tyhsieh  hsec   0 Jan  3 13:53 status
```

# Simple SQL injection example

---

## ❑ Username/password authentication

```
SELECT * FROM usrTable  
WHERE user =  
AND pass = ;
```

## ❑ No input validation

```
SELECT * FROM usrTable  
WHERE user = 'test'  
AND pass = 'a' OR 'a' = 'a'
```



# setuid program

---

## ❑ passwd

```
zfs[~] -chiahung- ls -al /usr/bin/passwd  
-r-sr-xr-x 2 root wheel 8224 Dec 5 22:00 /usr/bin/passwd
```

- /etc/master.passwd is of mode 600 (-rw-----) !

## ❑ Setuid shell scripts are especially apt to cause security problems

- Minimize the number of setuid programs

```
/usr/bin/find / -user root -perm -4000 -print |  
/bin/mail -s "Setuid root files" username
```

- Disable the setuid execution on individual filesystems
  - **-o nosuid**



# Why not su nor sudo?

## ❑ Becoming other users

- A pseudo-user for services, sometimes shared by multiple users

```
User_Alias newsTA=wangyr  
Runas_Alias NEWSADM=news  
newsTA ALL=(NEWSADM) ALL
```

- `sudo -u news -s` (?) **Too dirty!**
- `/etc/inetd.conf`
  - `login stream tcp nowait root /usr/libexec/rlogind rlogind`
- `~notftpadm/.rhosts`
  - `localhost wangyr`
- `rlogin -l news localhost`

# Security tools

---

- nmap
- john, crack
- PGP
- CA
- ...
  
- Firewall
- TCP Wrapper
- ...

# TCP Wrapper

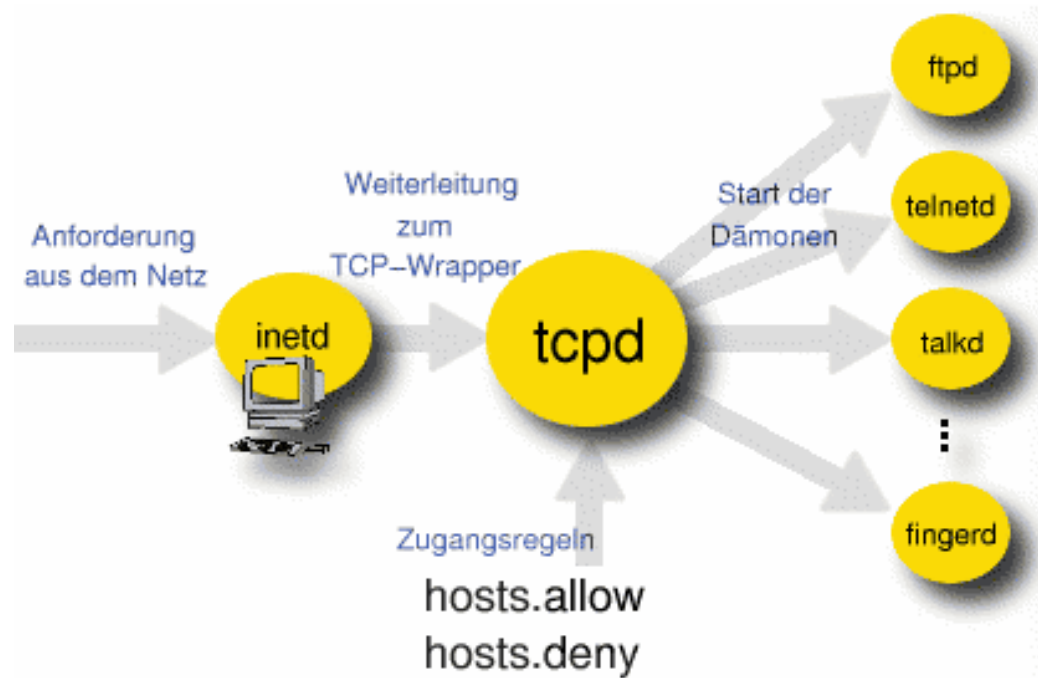
---

- ❑ There are something that a firewall will not handle
  - Sending text back to the source
  
- ❑ TCP wrapper
  - Extend the abilities of `inetd`
    - Provide support for every server daemon under its control
  - Logging support
  - Return message
  - Permit a daemon to only accept internal connetions

# TCP Wrapper

## □ TCP Wrapper

- Provide support for every server daemon under its control



# TCP Wrapper

- ❑ To see what daemons are controlled by inetd, see `/etc/inetd.conf`

```
#ftp  stream tcp    nowait root  /usr/libexec/ftpd  ftpd -l
#ftp  stream tcp6   nowait root  /usr/libexec/ftpd  ftpd -l
#telnet stream tcp    nowait root  /usr/libexec/telnetd telnetd
#telnet stream tcp6  nowait root  /usr/libexec/telnetd telnetd
shell stream tcp    nowait root  /usr/libexec/rshd   rshd
#shell stream tcp6  nowait root  /usr/libexec/rshd   rshd
login stream tcp    nowait root  /usr/libexec/rlogind rlogind
#login stream tcp6  nowait root  /usr/libexec/rlogind rlogind
```

- ❑ TCP wrapper should not be considered a replacement of a good firewall. Instead, it should be used in conjunction with a firewall or other security tools

# TCP Wrapper

## ❑ To use TCP wrapper

1. inetd daemon must start up with “-Ww” option (default)

Or edit /etc/rc.conf

```
inetd_enable="YES"  
inetd_flags="-wW"
```

- Edit /etc/hosts.allow

➤ Format:

daemon:address:action

- daemon is the daemon name which inetd started
- address can be hostname, IPv4 addr, IPv6 addr
- action can be “allow” or “deny”
- Keyword “ALL” can be used in daemon and address fields to means everything



# /etc/hosts.allow

---

## ❑ First rule match semantic

- Meaning that the configuration file is scanned in ascending order for a matching rule
- When a match is found, the rule is applied and the search process will stop

## ❑ example

```
ALL: localhost, loghost @adm_cc_cs : allow
ptelnetd pftpd sshd: @sun_cc_cs, @bsd_cc_cs, @linux_cc_cs : allow
ptelnetd pftpd sshd: zeiss, chbsd, sabsd : allow
identd : ALL : allow
portmap : 140.113.17. ALL : allow
sendmail : ALL : allow
rpc.rstatd : @all_cc_cs 140.113.17.203: allow
rpc.rusersd : @all_cc_cs 140.113.17.203: allow
ALL : ALL : deny
```

# /etc/hosts.allow

## ❑ Advance configuration

- External commands (**twist option**)

➤ twist will be called to execute a shell command or script

```
# The rest of the daemons are protected.
telnet : ALL \
        : severity auth.info \
        : twist /bin/echo "You are not welcome to use %d from %h."
```

- External commands (**spawn option**)

➤ spawn is like twist, but it will not send a reply back to the client

```
# We do not allow connections from example.com:
ALL : .example.com \
        : spawn (/bin/echo %a from %h attempted to access %d >> \
        /var/log/connections.log) \
        : deny
```

# /etc/hosts.allow

---

- Wildcard (**PARANOID** option)
  - Match any connection that is made from an IP address that differs from its hostname

```
# Block possibly spoofed requests to sendmail:  
sendmail : PARANOID : deny
```

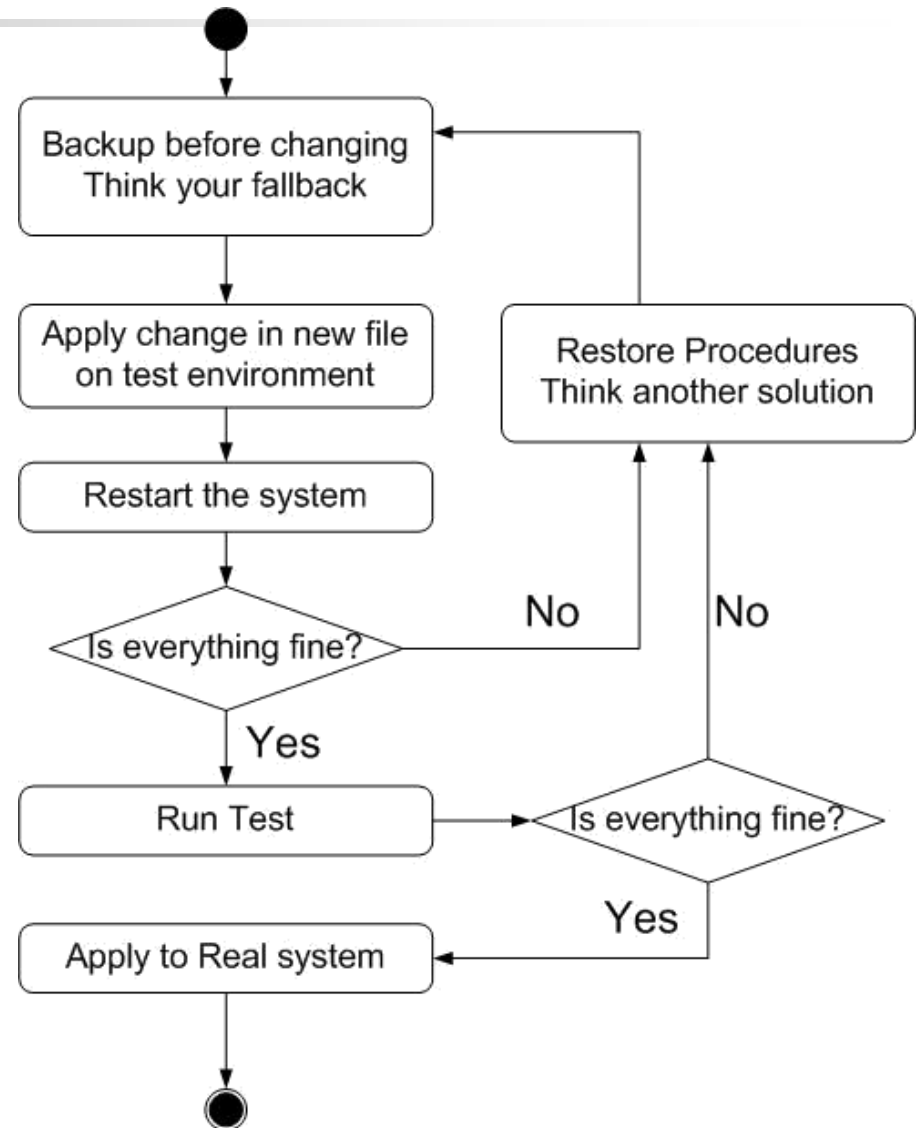
## ☐ See

- man 5 hosts\_access
- man 5 hosts\_options

# When you perform any change.

## □ Philosophy of SA

- Know how things really work.
- Plan it before you do it.
- Make it reversible
- Make changes incrementally.
- Test before you unleash it .



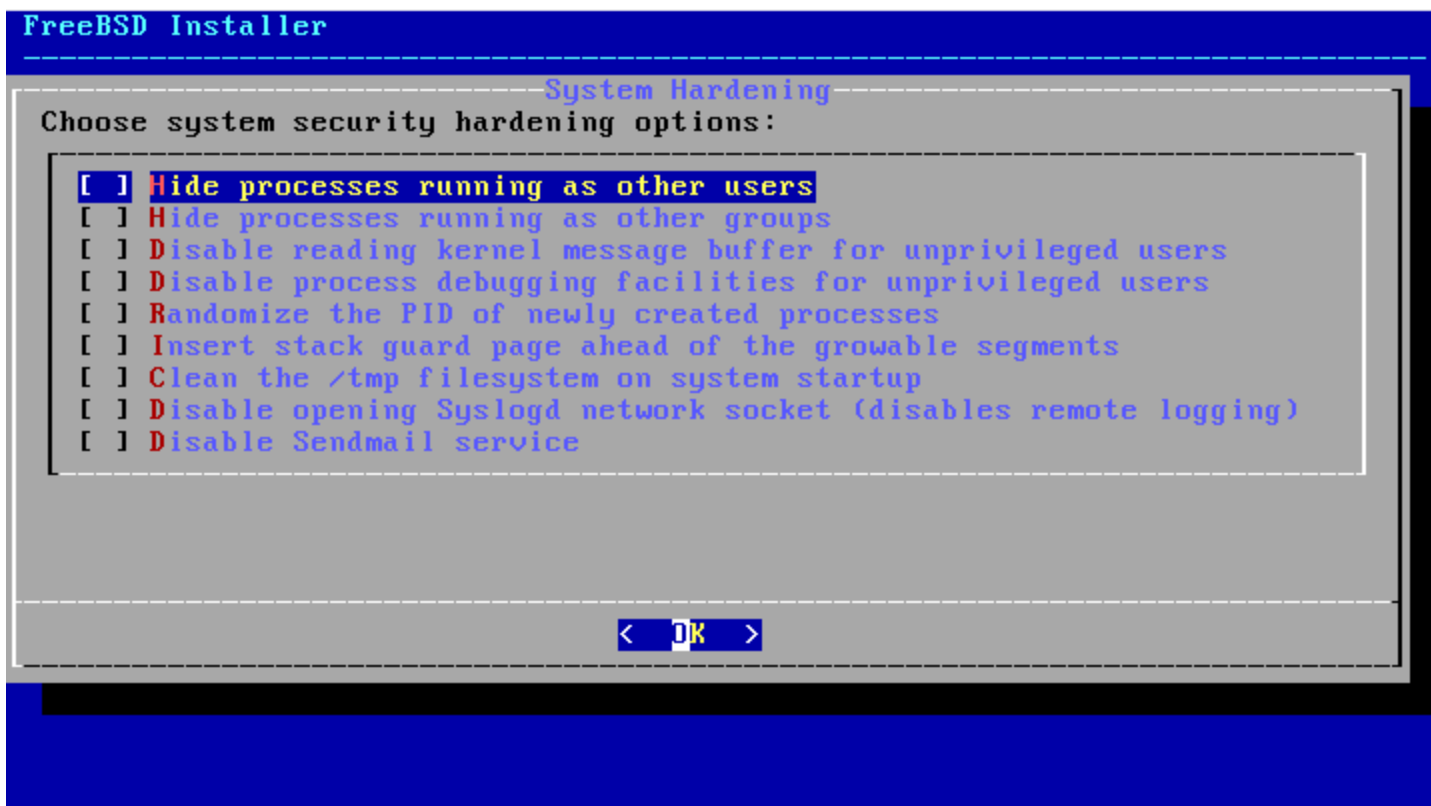


# Appendix

---

# System Security Hardening Options (1/3)

- ❑ Include various system hardening options during installation since FreeBSD 11.0-RELEASE



- /usr/src/usr.sbin/bsdinstall/scripts/hardening

# System Security Hardening Options (2/3)

---

- ❑ Hide processes running as other users
  - `security.bsd.see_other_uids=0`
  - Type: Integer, Default: 1
- ❑ Hide processes running as other groups
  - `security.bsd.see_other_gids=0`
  - Type: Integer, Default: 1
- ❑ Disable reading kernel message buffer for unprivileged users
  - `security.bsd.unprivileged_read_msgbuf=0`
  - Type: Integer, Default: 1
- ❑ Disable process debugging facilities for unprivileged users
  - `security.bsd.unprivileged_proc_debug=0`
  - Type: Integer, Default: 1

# System Security Hardening Options (3/3)

- ❑ Randomize the PID of newly created processes
  - `kern.randompid=$(jot -r 1 9999)`
    - Random PID modulus
  - Type: Integer, Default: 0
- ❑ Insert stack guard page ahead of the growable segments
  - `security.bsd.stack_guard_page=1`
  - Type: Integer, Default: 0
- ❑ Clean the /tmp filesystem on system startup
  - `clear_tmp_enable="YES" (/etc/rc.conf)`
- ❑ Disable opening Syslogd network socket (disables remote logging)
  - `syslogd_flags="-ss" (/etc/rc.conf)`
- ❑ Disable Sendmail service
  - `sendmail_enable="NONE" (/etc/rc.conf)`