Pissing off the bad guys by porting grsecurity to HardenedBSD

Shawn Webb
HardenedBSD Cofounder

About Me

- Cofounder of HardenedBSD
- Infosec professional
- Former ClamAV core developer
- Open source enthusiast
- ZFS fanboy
- Tor relay operator

Agenda

- The current state of FreeBSD security
- HardenedBSD history, part 1
- Exploit mitigation discussion
- HardenedBSD history, part 2
- Case Study
- Dive into HardenedBSD’s exploit mitigations
- The future of HardenedBSD

Current state of FreeBSD

- FreeBSD 4.0, released in 2000, included jails
- Gimped stack canary
- No ASLR, W^X, or other exploit mitigations
- Sandboxing with Capsicum
- State of security: era 1999 to 2001
- 24+ year old bug in libc
  - Fix by FreeBSD introduced new vulnerability

Current State of FreeBSD

- Multiple bhyve vulnerabilities providing hypervisor escapes
- Vulnerabilities against freebsd-update and portsnap reported over four years ago
- Weaknesses in seeding entropy provider
- Unwillingness to implement proven exploit mitgations. “ASLR is dead!”

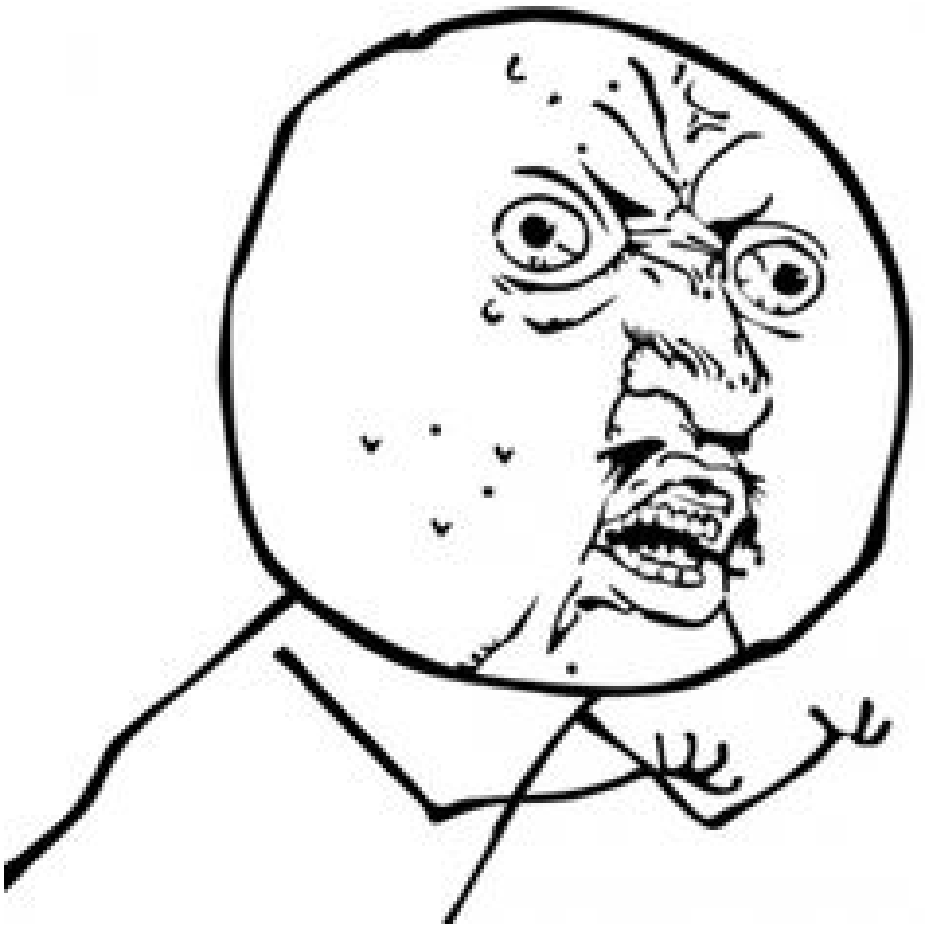
Current State of FreeBSD

- Capsicum + jails thought to be end-all-be-all of exploit mitigations
  - Capsicum is great for sandboxing
  - Suffers from data-only attacks
  - Difficult to use
  - Applications must be written with sandboxing in mind

Current State of FreeBSD

- FreeBSD knows about many weaknesses and insecurities within FreeBSD, but doesn’t want the bad PR that comes from fixing them
  - Fixing them (good) would mean publicly acknowledging the lack of them for over 15 years (bad).
- NSA must love FreeBSD
  - 07 Mar 2017 leaked CIA docs (Vault7) show some love

Current State of FreeBSD



Why not Linux?

- Linux is a hack
- BSD is an engineering marvel
- GPL holds innovation hostage
- State of security in Linux sucks, too
  - Not nearly as bad as FreeBSD
- Must apply third-party patch for true security (grsecurity)

About HardenedBSD

- “Spork” of FreeBSD
  - Merge from FreeBSD every six hours
- Porting grsecurity to HardenedBSD
  - 26 Apr 2017 – grsecurity goes private
    - Affects many Linux distributions
    - Doesn’t affect HardenedBSD

HardenedBSD History, Part 1

- Only exploit mitigation in FreeBSD: partial stack cookie
- 2013, Oliver Pinter and I independently notice lack of exploit mitigations
- 2013, Oliver Pinter and I work together on the first exploit mitigation: ASLR
- HardenedBSD born in 2014
- Attempt to upstream ASLR failed

About secadm

- Toggle exploit mitigations
- Abuses MAC framework awesomely
  - BTW, MAC framework = awesome rootkit framework
- Extra features: TPE, Integriforce, application whitelisting

Exploit Mitigations – PaX ASLR

- Address Space Layout Randomization
- Deltas to randomize where objects are loaded in memory
- Foundational to other exploit mitigations
- Originally meant to be temporary solution
  - Permanent solution: PaX RAP
- Showing weaknesses in some cases, but not broken

Exploit Mitigations - SEGVGUARD

- Inspired by PaX
- Prevents ASLR bruteforce attacks
- Daemons/applications that autorestart
- Prevents execution for <x> seconds after <y> crashes during <z> window

Exploit mitigations – W xor X

- Memory mappings cannot be both Writable and eXecutable
- PaX model:
  - If historically writable, never executable
  - If historically executable, never writable
- OpenBSD model:
  - Only protect at mmap time
  - Bypass with mprotect

Exploit Mitigations – W xor X

- HardenedBSD uses PaX model
  - I term it “Strict W^X”
- Some applications dislike strict W^X
  - Those with a JIT
  - Firefox almost doing it right

Exploit Mitigations - SafeStack

- Separate safe stack and unsafe stack
- Buffers, undesireable overflow objects, on unsafe stack
- Integers, return pointer on safe stack
- Requires ASLR and W^X to be effective
- HardenedBSD only BSD to enable SafeStack

Exploit Mitigations - CFI

- Prevent unwanted transfer of control to undefined or arbitrary code
- Microsoft Control Flow Guard (CFG)
  - Weaker variant of CFI
- PaX RAP
  - Stronger variant of CFI
  - Requires GPLv3 and is patent-pending
- Cross-DSO CFI requires ASLR and W^X
- HardenedBSD only BSD to enable CFI

Exploit Mitigations – TPE & Integriforce

- TPE: Only execute applications in root-owned, non-world and non-group writable directories
- Integriforce: executable file integrity enforcement
  - Inspired by NetBSD verified exec
  - Can be used as application whitelist system
- Integriforce rules distributed with binary updates (signed hashes of every binary and shared library in base!)

Exploit Mitigations - Hardening

- Hardening the following subsystems:
  - [lin]procfs
  - IPv4 and IPv6 stack
  - Boot process
  - Kernel modules
  - System control (sysctl) nodes
  - System messages (dmesg)
  - Kernel info leaks

About Toggles

- Providing toggles is fine
  - Set via root only
- SELinux is bad
  - Having too many toggles is bad
  - Hard-to-configure toggles are bad
- HardenedBSD desktop:
  - Only need to toggle half of W^X for Firefox
    - Two secadm rules

HardenedBSD History

- 18 Mar 2015 – NOEXEC introduced
- 24 Apr 2015 - Integriforce
- 04 Jul 2015 – ASLR completed
- 07 Sep 2015 – arm64 support
- 31 Dec 2015 – Secure binary updater
- 25 Feb 2016 – Prevent RTLD from creating executable per-thread stacks (amd64)
- 26 Mar 2016 – LibreSSL in base

HardenedBSD History

- 06 Jun 2016 – OPNsense publishes build with HardenedBSD ASLR enabled
- 11 Jun 2016 – RISC-V support
- 08 Aug 2016 – PIE, RELRO, BIND_NOW for ports
- 27 Nov 2016 – SafeStack in base
  - SafeStack being added to individual ports even today
- 20 Dec 2016 – SEGVGUARD becoming more stable
- 02 Mar 2017 – non-Cross-DSO CFI in base

Case Study

- NTP CVEs announced this week
- Multiple stack buffer overflows
- Fully mitigated with combination of:
  - ASLR
  - W^X
  - SafeStack
  - CFI
- Easy to exploit on FreeBSD
- Hard (impossible?) on… HardenedBSD

Case Study

```
1544 static void
1545 ctl_putstr(
1546         const char *    tag,
1547         const char *    data,
1548         size_t          len
1549         )
1550 {
1551         char buffer[512];
1552         char *cp;
1553         size_t tl;
1554
1555         tl = strlen(tag);
1556         memcpy(buffer, tag, tl);
1557         cp = buffer + tl;
1558         if (len > 0) {
1559                 INSIST(tl + 3 + len <= sizeof(buffer));
1560                 *cp++ = '=';
1561                 *cp++ = '"';
1562                 memcpy(cp, data, len);
1563                 cp += len;
1564                 *cp++ = '"';
1565         }
1566         ctl_putdata(buffer, (u_int)(cp - buffer), 0);
1567 }
```

CVE 2017-6458: contrib/ntp/ntpd/ntp_control.c @ ntp 4.2.8p9 (3298f99b1994fc7807e46ed367569072b1921960)

Deep Dive

- Code and stuff and things

The Future of HardenedBSD

- FS EXTATTR support for exploit mitigation toggles
- secadm in base
- Critical: documentation
- Full audit of the linuxulator
  - VDSO cannot be randomized
- PaX UDEREF

The Future of HardenedBSD

- System call hardening
- More sysctl and KPI hardening
- Further work with OPNsense
  - OPNTor, anyone?
- Cross-DSO CFI
- SROP mitigation
- CFI and SafeStack ported to arm64

The Future of HardenedBSD