Projekt TrustedBSD jako klucz do bezpieczeństwa systemu FreeBSD

Wojciech A. Koszek
IX LO. im. C.K. Norwida w Częstochowie
dunstan@FreeBSD.czest.pl
SKI2005
09.06.2005

Główne funkcje jądra:

- Nadzór nad pracą procesora/procesorów
- Zarządzanie zadaniami
- Zarządzenie pamięcią – podsystem VM
  - Podział między pamięć poszczególnych procesów
  - Podział na przestrzeń jądra i użytkownika
- Kontrola zasobów maszyny (instrukcje CPU, przerwania, porty I/O)

Systemy operacyjne kiedyś..

- Jeden użytkownik z uprawnieniami nadzorcy
- Brak mechanizmów ochrony zasobów
- Brak potrzeby stosowania jakiegokolwiek modelu bezpieczeństwa
- Ograniczony dostęp do nowoczesnych technologii

Systemy operacyjne dziś

- Wielu użytkowników
  - Hierarchiczny podział uprawnień
- Wielostopniowa kontrola zasobów
  - Mechanizmy wirtualizacji zasobów
- Konieczność stosowania mechanizmów bezpieczeństwa

Mechanizmy kontroli w jądrze systemu?

- Doskonały dostęp do wszystkich zasobów
- Brak ograniczeń z powodu pracy w trybie chronionym, najwyższy poziom uprzywilejowany (x86 – ring0)
- Limitowanie działań użytkownika to właśnie kontrola w warstwie jądra

Posix.1e

- Access Control Lists
- Auditing
- Capabilities
- Mandatory Access Control
- Information Labeling

Bezpieczeństwo niesie za sobą

- Zapotrzebowanie na konkretną funkcjonalność:
  - Dodatkowe dane w strukturach systemu operacyjnego
  - Kontrola na podstawie większej ilości kryteriów

Dlaczego FreeBSD?

- Stabilność (najdłuższy uptime według Netcraft - 26 na 50 systemów to FreeBSD)
- Bezpieczeństwo (jail(), securelevels, TrustedBSD, GBDE, (Fast)IPSec, OpenPAM, PF/IPF/ipfw2)
- Zaufanie (MacOSX, DARPA, NSA, NAI LABS)
- Dynamiczny rozwój, czytelność kodu
- Doskonałe perspektywy na przyszłość (implementacja nowych rozwiązań opłacalna)

Główny cel TrustedBSD: Posix.1e we FreeBSD

- Kod FreeBSD z gałęzi 5.x
- Sposób rozwoju zbliżony do FreeBSD
- Możliwe przeniesienie na inne systemy z rodziny BSD

Stan prac nad TrustedBSD

- Integracji uległy:
  - MAC
  - Rozszerzone atrybuty systemu plików
  - ACL
- W trakcie rozwoju:
  - Capabilities
  - Auditing (zaprezentowany na BSDCan2005)

Mandatory Access Control: wywołania systemowe

/usr/src/sys/kern/kern_exec.c:

```
[..]
static int
do_execve(td, fname, argv, envv, mac_p)
[..]
#ifdef MAC
        error = mac_execve_enter(imgp, mac_p);
        if (error) {
[..]
#endif
[..]
#ifdef MAC
       mac_execve_exit(imgp);
[..]
```

Mandatory Access Control: warstwa procesów

```
[..]
int
cr_cansee(struct ucred *u1, struct ucred *u2)
[..]
#ifdef MAC
        if ((error = mac_check_cred_visible(u1, u2)))
return (error);
#endif
[..]
```

Rozszerzone atrybuty systemu plików?

- Sama nazwa pliku jest niewystarczająca
- Liczne problemy ze śledzeniem zmian:
  - Problemy z integralnością bazy danych dotyczących zmian
- Integracja z własnym oprogramowaniem:
  - Niezwykle rozbudowane możliwości

ExtAttr we FreeBSD

- Natywne wsparcie dla rozszerzonych atrybutów przez UFS2

```
$ setextattr user CK_SHA1
`sha1 -q FILE` FILE
$ lsextattr user FILE
FILE    CK_SHA1
$ getextattr user CK_SHA1
```

Access Control Lists

- Roszerzenie standardowego mechanizmu zabezpieczeń (*OGU+RWX*)
- Możliwość zdefiniowania szczegółowych uprawnień: odczyt dla użytkownika "dunstan", wykonywanie dla grupy "admin", użytkownicy "user1" i "user2" tylko odczyt.

Wykorzystanie istniejącej funkcjonalności

- moduły z polityką ładowane poprzez */boot/loader.conf*
- użytkownicy oznaczani etykietą poprzez klasy logowania w */etc/login.conf*

W trakcie tworzenia

- Capabilities
- Auditing
- Dokumentacja

Capabilities

- Ponieważ UID == 0 to czasami zbyt wiele
- Wystarczy możliwość "zbindowania" portu poprzez *bind(2)*, utworzenia gniazda dzięki *socket(2)*
- Minimalizacja strat w przypadku przejęcia kontroli nad aplikacją

Auditing

- Logowanie interesujących zdarzeń (zasięg użytkownika i systemu)
- Brak dostępu do rekordów audytowania przez aplikację audytowaną

TrustedBSD nie jest dobre na wszystko!

- Ataki na implementacje podsystemów jądra:
  - PR: kern/77748 - Local DoS w funkcji if_clone_list()
  - PR: kern/77421 - Local DoS w funkcji ifconf()

Podsumowanie

- System posiada narzędzia konfiguracji (*setfacl(1)*, *setpmac(8)*, *setfmac(8)*, *ugidfw(8)*)
- Potrzebna jedynie rekompilacja jądra:
  - Niezbędne opcje opisane na stronach podręcznika *mac(3)*
- Audytowanie zdarzeń dostępne już niebawem

Koniec

- Dziękuję za uwagę. Zapraszam do zadawania pytań