Bezpieczeństwo oprogramowania

Prezentacja na systemy operacyjne 2006/07

Grzegorz Kulewski

styczeń 2007

Błędy w oprogramowaniu

- niedostateczne sprawdzanie danych wejściowych i uprawnień,
- błędy w konstrukcji aplikacji,
- nadmierne zaufanie do użytkowników i środowiska,
- niepotrzebny determinizm programów,
- sytuacje wyścigów,
- brak limitów alokacji zasobów

Najpopularniejsze metody ataków

- przepełnienie bufora,
- ataki typu *injection*,
- *cross site scripting*,
- wykorzystywanie sytuacji wyścigów,
- ataki DOS i DDOS

Błąd przepełnienia bufora

- **Nazwa angielska?** buffer overflow, smash stack attack
- **Cel ataku?** najczęściej demon lub program działający z większymi uprawnieniami niż nasze
- **Sposób przeprowadzenia?** ręczny lub (bardzo często) automatyczny
- **Możliwe skutki?** przejęcie uprawnień ofiary i/lub wykonanie dowolnego kodu
- **Zagrożenie?** bardzo wysokie, „błąd dekady lat ’90”, ale niestety i obecnej

Błąd przepełnienia bufora

Mechanizm ataku

Jeżeli w funkcji znajduje się bufor, którego zakres nie jest prawidłowo sprawdzany to atakujący może **doprowadzić do jego przepełnienia** i (w większości implementacji języka C i wielu innych języków) **nadpisać inne dane funkcji i/lub adresy powrotu dostarczonymi przez siebie danymi**.

Atakujący zmienia tą metodą stan programu „w locie”, w sposób nieprzewidywalny dla programisty piszącego kod.

Błąd przepełnienia bufora
Sposoby obrony

- **pisanie bezpiecznego kodu**,
- wyłączenie praw do wykonywania stosu,
- randomizowanie adresów stosu i bibliotek,
- stosowanie łatek typu PaX, GrSecurity itp.,
- dodatkowe wsparcie w kompilatorach (SSP, gcc-4.1 i inne)

Dla administratorów

- posiadać wiedzę i doświadczenie praktyczne,
- aktualizować oprogramowanie,
- usuwać niepotrzebne oprogramowanie, strony i dane,
- nie stosować bez sprawdzenia konfiguracji domyślnych,
- nie ufać użytkownikom i ich niepotrzebnie nie informować,
- tworzyć kopie zapasowe,
- nie wyłączać bezpieczeństwa, nawet na chwile,
- oddzielić i izolować usługi, serwery i sieci,
- testować swoją konfigurację (testy penetracyjne, audyty)

Dla programistów

- **sprawdzać dane wejściowe**
- „wszystko co nie jest dozwolone jest zabronione”,
- dobrze projektować oprogramowanie, unikać bałaganu,
- nie iść za bardzo na rękę użytkownikom,
- unikać niepotrzebnego determinizmu