SELinux

SELinux – Security Enhanced Linux

czyli

Linux o podwyższonym bezpieczeństwie

Najkrócej mówiąc...

- SELinux jest systemem z MAC (Mandatory Access Control), który realizuje politykę RBAC (Role Based Access Control) za pomocą DTAC (Dynamically Typed Access Control).
- Ideą SELinuksa jest odebranie procesom tych uprawnień, które nie są im potrzebne.
- Dzięki temu zniszczenia, jakie mogą zostać poczynione w razie włamania do systemu, będą znikome (ograniczone do uprawnień jakie posiada proces, w którym błędy wykorzystano do włamania).

Historia SELinuksa

- Projekt DTMach (Distributed Trusted Mach) – rozwijał rozwiązania wspomagające bezpieczeństwo – m.in. MAC (Mandattory Access Control). Zapoczątkowany w 1992 roku.
- Projekt Flux – rozwijał dla systemu Fluke (który przejął rozwiązania z DTMach) architekturę FLASK.
- Projekt SELinux – zintegrował architekturę FLASK z jądrem Linuksa.
- Prace nad SELinuxem są sponsorowane przez NSA (Natonal Security Agency) a prowadzi je SCC (Secure Computing Corporation).

Główne części składowe SELinuksa

- ***Jądro*** - SELinux jest modułem LSM (Linux Security Modules). Dzięki temu może korzystać z interfejsów, dających mu kontrolę nad dostępem do obiektów systemu (plików, katalogów, gniazd, urządzeń itp.), co daje SELinuksowi możliwość wymuszenia własnej polityki bezpieczeństwa (zdefiniowanej w Policy).
- ***Modyfikacja niektórych programów*** - Dla prawidłowego działania SELinuksa niezbędne jest zmodyfikowanie programów, mających kluczowe znaczenie dla bezpieczeństwa systemu (należy rozszerzyć je o obsługę SELinuksa). Można to zrealizować na etapie instalacji tych programów poprzez: nałożenie łat na ich źródła lub pobranie już zmodyfikowanych wersji tych programów. Do programów wymagających takiej modyfikacji zaliczają się m.in. ls, ps, login, ssh i xdm.
- ***Policy*** - Policy to zbiór reguł określających prawa wykonywania działań przez użytkownika, prawa dostępu do obiektów systemu oraz zachowanie samego systemu.

MAC – Mandatory Access Control

- Główna zasada: użytkownik nie decyduje o prawach dostępu do obiektów i zabezpieczeniach!
- Są one definiowane odgórnie przez administratora. (w przypadku SELinuksa są zapisane w postaci reguł w *Policy*).

RBAC – Role Based Access Control

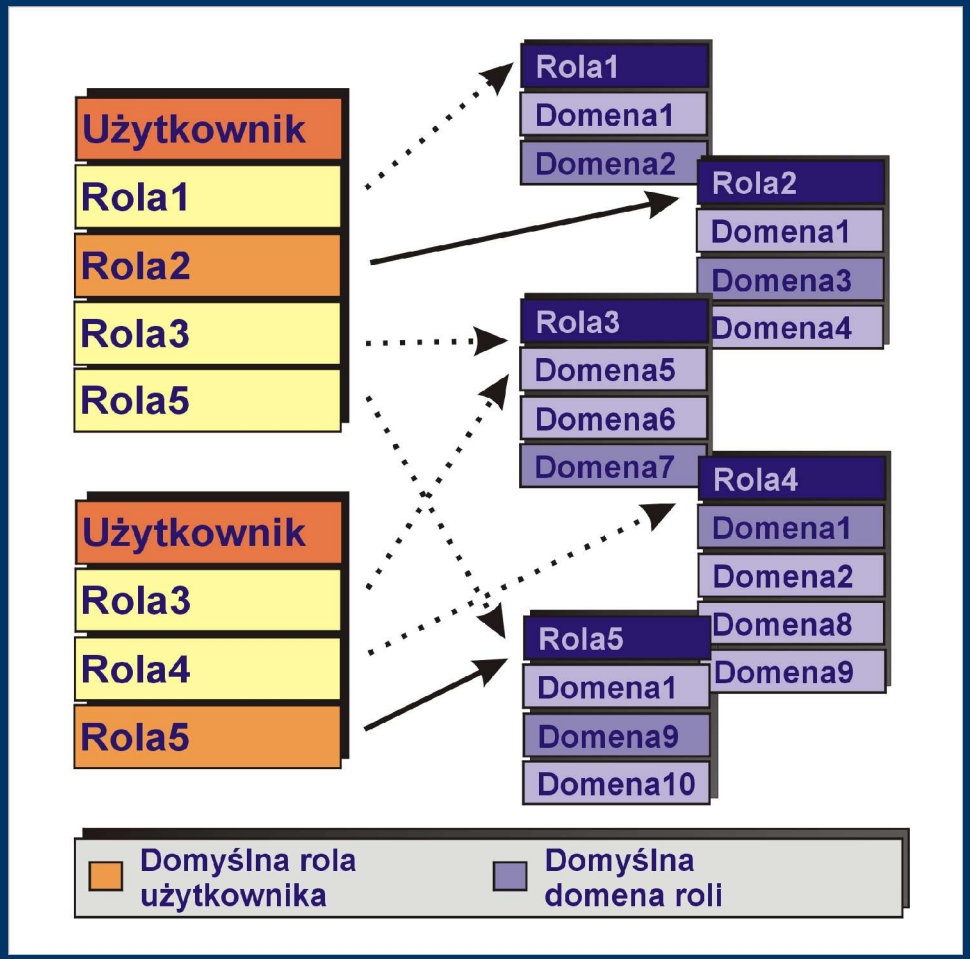
- System kontroli dostępu oparty na rolach.
- Rola – zestaw praw do wykonywania określonych działań.
- Może zachodzić potrzeba zmiany roli w ramach, której wykonywane jest działanie.
- Standardowy Unix jest systemem z RBAC.
- SELinux daje każdemu użytkownikowi zestaw ról, z czego jedna z nich jest rolą domyślną. Wielu użytkowników może mieć te same role.
- SELinux pozwala na dokładniejsze dostosowanie ról do konkretnych potrzeb.
- Zasady RBAC w SELinuksie realizowane są za pomocą DTAC. Każda rola posiada zestaw domen dostępu.

DTAC – Dynamically Typed Access Control

- Domenowy system kontroli dostępu.
- Każdy obiekt w systemie posiada swój typ.
- Typ ten jest narzucany za pomocą odgórnych zasad (zawartych w *Policy*) . Użytkownik nie ma na nie wpływu.
- System z SELinuxem jest przygotowany nawet na kilkaset tysięcy takich zasad.
- Dla każdego typu tworzone są reguły, które definiują zachowanie systemu w momencie wykonywania akcji na obiektach tego typu.
- Firma IBM pracuje nad narzędziem, pozwalającym na sprawdzenie poprawności i spójności zestawu narzuconych reguł (zawartych w *Policy*).

Działanie SELinuksa

- Każdy użytkownik systemu może posiadać wiele ról. Jedna z tych ról jest domyślna dla danego użytkownika.
- Każda rola dysponuje zestawem domen. Jedna z tych domen jest domyślna dla danej roli.
- Role mogą być wspólne dla wielu użytkowników a domeny mogą być wspólne dla wielu ról.
- W danym momencie użytkownik wykonuje dokładnie jedną rolę w jednej domenie.
- Dla każdej pary (domena roli użytkownika, typ obiektu) zdefiniowane są reguły zachowania systemu.



Działanie SELinuksa - cd.

- Standardowym modelem uprawnień w Linuksie jest UGO (User, Group, Others). SELinux nie rozszerza tego modelu – tzn. nie nadaje uprawnień, których nie miał użytkownik w systemie bez SELinuksa.
- Dzięki czemu, jeśli nawet zasady *Policy* nie ograniczałyby w żaden sposób uprawnień użytkownika (pozwalałyby na wszystko), wówczas poziom bezpieczeństwa systemu operacyjnego jest taki jak zwykłego Linuksa (czyli nie taki zły!)

Kontekst bezpieczeństwa

- Kontekst bezpieczeństwa procesu:

```
użytkownik:rola:domena
```

- a więc na przykład:

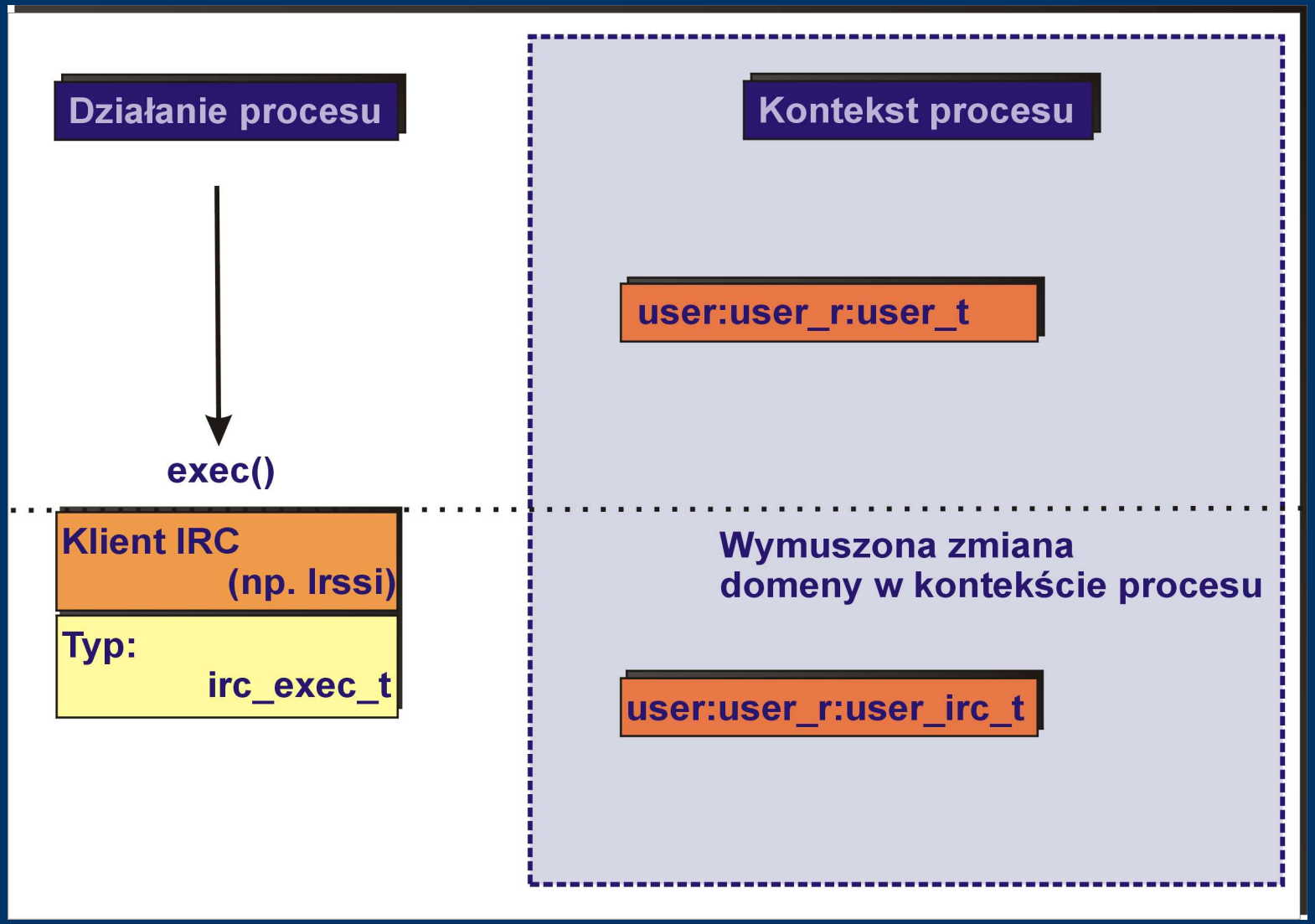
```
pawel:user_r:user_t
```

- Tak naprawdę także obiekty w systemie mają pełen kontekst bezpieczeństwa np. system_u:object_r:http_port_t może być kontekstem bezpieczeństwa portu TCP. Ale istotna jest tylko ta ostatnia część - typ.

Rodzaje reguł SELinuksa

- Najczęściej występujące rodzaje reguł:
  - Definiowanie operacji dozwolonych (wykonywania, czytania, pisania i innych).
  - Definiowanie dozwolonych zmian roli w ramach użytkownika (listy ról dozwolonych dla użytkownika).
  - Definiiowanie dozwolonych zmian domeny w ramach roli (listy domen dozwolonych dla roli).
  - Definiowanie wymuszonych zmian roli przy wykonywaniu operacji na obiekcie lub wykonywaniu programu.
  - Definiowanie wymuszonych zmian domeny przy wykonywaniu operacji na obiekcie lub wykonywaniu programu.
- W standardowej *Policy* odchodzi się od wymuszeń zmian roli (są one zastępowane wymuszeniami zmiany domeny).

Przykład – klient IRC



Definicje Type Enforcement

- Definicje wymuszonych przejść:
  - domain_auto_trans(initrc_t, sshd_exec_t, sshd_t)
    Makro, które jest regułą oznaczającą, że jeśli proces w domenie initrc_t uruchomi program z pliku o typie sshd_exec_t, to proces wykonujący ten program będzie działał w domenie sshd_t.
  - file_type_auto_trans(sshd_t, tmp_t, sshd_tmp_t)
    Makro, które jest regułą oznaczającą, że jeśli proces w domenie sshd_t otworzy (lub utworzy) plik o typie tmp_t, wówczas zmieni się typ tego pliku na sshd_tmp_t.

Przykład – przeglądarka stron WWW

- Skomplikowany program = błędy.
- Wykorzystanie błędów = zagrożenie dla naszego systemu.
- Plik wykonywalny przeglądarki internetowej ma typ netscape_exec_t.
- Po wykonaniu funkcji exec() proces, który ją wykonał przechodzi do domeny user_netscape_t.
- Proces będąc w domenie user_netscape_t ma prawo:
  - tworzyć pliki o typach user_netscape_t oraz user_netscape_rw,
  - wykonywać pobrane programy, ale wyłącznie w ramach domeny user_netscape_t,
  - drukować (wykonywać program lpr).

Definicje Type Enforcement - cd.

- Definicje dozwolonych przejść:
  - domain_trans(sshd_t, shell_exec_t, sysadm_t)
    Makro, które jest regułą pozwalającą procesowi działającemu w domenie sshd_t, który uruchomił program z pliku o typie shell_exec_ts na zmianę domeny na sysadm_t.
  - type_change user_t tty_device_t:chr_file user_tty_device_t;
    Reguła pozwalająca procesowi działającemu w domenie user_t na zmianę typu pliku tty_device_t na user_tty_device_t.

Zalety SELinuksa

- Rozwiązanie to jest:
  - całościowe – nie koncentruje się ani na usunięciu wybranych błędów w polityce bezpieczeństwa systemu, ani na rozwiązywaniu problemów wynikających z przyjętej konstrukcji systemu, ale proponuje kompletne rozwiązanie.
  - elastyczne – administrator może dostosować reguły w Policy do aktualnych potrzeb.
  - rozszerzalne – administrator może dołączyć nowe reguły do Policy.
  - ciągle rozwijane – rozwojem tego rozwiązania zajmuje się firma SCC (Secure Computing Corporation), której prace są finansowane przez NSA.
  - standaryzowane – mechanizm MAC wykorzystujący etykietowanie obiektów jest zgodny z normą POSIX.6.

Test Russella Cokera

- Russell Coker – developer Debiana udostępnił w Internecie maszynę i upublicznił hasło roota.
- Pomimo tego, że wpisy w *Policy* były standardowe – nikomu nie udało się niczego popsuć w systemie.