SIO2Jail

Wojciech Dubiel
Tadeusz Dudkiewicz
Przemysław Jakub Kozłowski
Maciej Wachulec

Motywacja

Konkursy algorytmiczne wymagają specyficznego środowiska do uruchamiania zgłaszanych programów

- Pomiar czasu / złożoności obliczeniowej
  - Najlepiej deterministyczny
- Zapewnienie powtarzalnego wykonania
- Sandbox zabezpieczający przed
  - naruszeniem zasad – np. tworzeniem wielu wątków, obejściem limitu pamięci
  - ingerencją w system sprawdzający – np. czytanie pliku zawierającego poprawne rozwiązanie, modyfikacja wyniku pomiaru czasu
  - uzyskaniem kontroli nad serwerem sprawdzającym i wykorzystaniem do dalszych ataków

Motywacja

Obecne rozwiązanie – `oitimetool`:

- skupia się głównie na mierzeniu czasu, sandboxowanie traktuje drugorzędnie
- przepisuje kod maszynowy w momencie pierwszego wykonania
- używa do tego biblioteki PIN
  - własnościowa biblioteka Intela do instrumentacji kodu
  - nie działa z nowymi kernelami
  - z założenia nie służy do uruchamiania niezaufanego kodu
- program zawodnika jest w tej samej przestrzeni adresowej co oitimetool
  - trudno odróżnić błąd programu zawodnika od błędu sprawdzaczki
  - prowadzi to do podatności pozwalających na ucieczkę
- sandboxowanie wyłącznie przez blokowanie syscalli
  - mocno związane z implementacją biblioteki standardowej
  - ma zhardcodowaną politykę
  - trudno dodać wsparcie dla nowych języków

Dostępne technologie

- perf
- seccomp-bpf
- ptrace
- namespaces
  - mount namespaces
  - PID namespaces
  - UTS namespace
  - IPC namespace
  - user namespace
- rlimit
- cgroup

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci
- trafienia i nietrafienia w cache

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci
- trafienia i nietrafienia w cache
- itd.

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci
- trafienia i nietrafienia w cache
- itd.

Perf – API w Linuxie do czytania tych liczników

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci
- trafienia i nietrafienia w cache
- itd.

Perf – API w Linuxie do czytania tych liczników

- odczyt liczby zdarzeń dotyczących wybranego procesu

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci
- trafienia i nietrafienia w cache
- itd.

Perf – API w Linuxie do czytania tych liczników

- odczyt liczby zdarzeń dotyczących wybranego procesu
- ...lub grupy procesów

Dostępne technologie – perf

Nowoczesne procesory mają liczniki sprzętowe, zliczające:

- cykle w których procesor nie spał
- wykonane instrukcje
- odwołania do pamięci
- trafienia i nietrafienia w cache
- itd.

Perf – API w Linuxie do czytania tych liczników

- odczyt liczby zdarzeń dotyczących wybranego procesu
- ...lub grupy procesów
- z podziałem na przestrzeń jądra i użytkownika

Dostępne technologie – seccomp-bpf

Seccomp – mechanizm filtrowania wywołań systemowych w Linuxie

- przygotowujemy program filtrujący zapisany w BPF
- przekazujemy go do jądra
- jądro sprawdza filtr przy każdym wywołaniu systemowym
  - bardzo wcześnie, zanim przejdzie do kodu odpowiedzialnego za konkretny numer wywołania – ograniczamy powierzchnię ataku
  - bardzo szybko – wszystko dzieje się w jądrze, nie trzeba przełączać kontekstu

Dostępne technologie – ptrace

Ptrace – standardowy interfejs POSIXowy do debugowania procesów. Pozwala m.in na:

- zatrzymanie procesu
  - przed wejściem w wywołanie systemowe
  - po wyjściu z wywołania systemowego
  - po wystąpieniu sygnału TRAP
  - po zwróceniu odpowiedniej wartości przez filtr seccomp
- odczyt rejestrów i przestrzeni adresowej zatrzymanego procesu
- zapis rejestrów i przestrzeni adresowej zatrzymanego procesu

Dostępne technologie – namespaces

Dostępne technologie – rlimit

rlimit (a.k.a. ulimit) – POSIXowy mechanizm określania maksymalnego rozmiaru zasobów zużywanych przez proces. W Linuxie pozwala ograniczyć m.in.:

- rozmiar przestrzeni adresowej
- rozmiar stosu
- czas działania programu
- liczbę otwartych plików

Dostępne technologie – cgroup