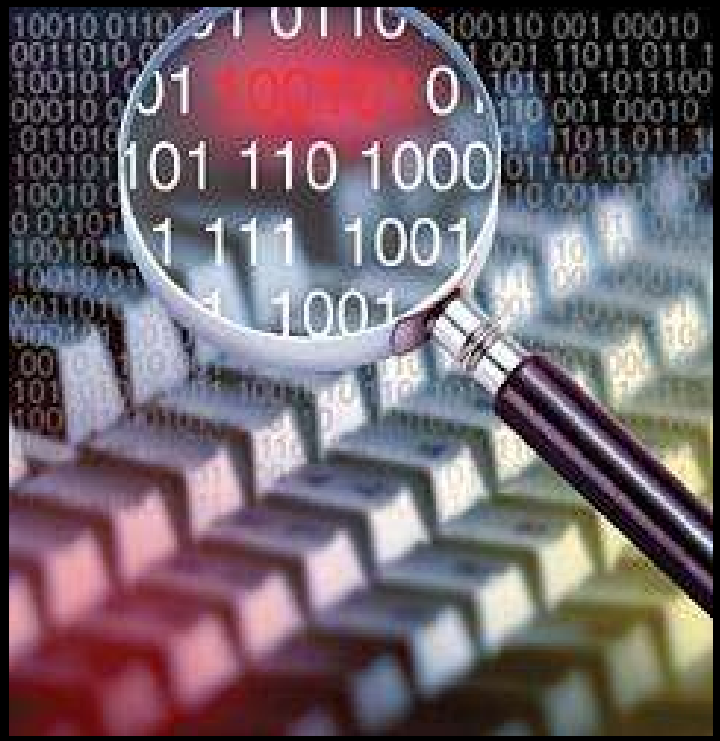



# COMPUTER INTRUSION DETECTION

Izabela Adamczyk

Wrocław 2005

# Problem

- informacja od wieków
  - daje władzę
  - w nieodpowiednich rękach jest niebezpieczna
- komputeryzacja
  - nowy sposób zarządzania informacją
  - system nie jest doskonały (administratorzy czuwają)
  - powszechny dostęp (sieci, Internet)
  - nieznane pułapki
-

# IDS

- **Intrusion** (wtargnięcie)

  nieautoryzowany dostęp do systemu

  niedozwolony sposób korzystania z systemu

- **IDS** (Intrusion Detection System)

  - pomaga administratorowi w wykrywaniu wtargnięć

    - wykrywa

    - włącza alarm

  - nie próbuje powstrzymać intruza

  - nie może nadużywać
    cierpliwości administratora)
    (Hannibal ad portas!)

# Rodzaje IDS

**Network Based IDS**
analiza pakietów

**Host Based IDS**
śledzenie programów, użytkowników, obciążenia syetmu

IDS

Application Based IDS
atak na aplikację (np.bazę danych)

Hybrydowy IDS

# Metody budowy IDS

**SPOSOBY ANALIZY DANYCH**

## Analiza anomalii

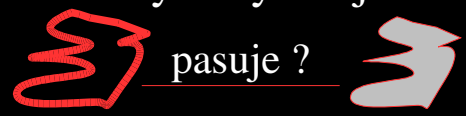
(anomaly detection)

- tworzony profil na bazie *normalnego* działania systemu,
  *odstające* zachowania to atak
- trudno tworzyć profile
- system czasem zachouje się nienormalnie, coć to nie atak
- konieczność tworzenie odrębnych profili dla różnych systemów, aplikacji, użytkowników
- kosztowne

## Analiza sygnatur

(signature, misuse detection)

- tworzony model *ataku*, *podobne* zachowania to atak
- tak działa antywirus
- nie sprawdza się w nowych sytuacjach




# Przykładowy IDS

Host Based IDS
+
Analiza Anomalii
+
Sequences of System Calls

# Sequences of System Calls

- brak bezpośredniego dostępu programów do dysku itp
- odwołania przez funkcje systemowe (`open()-read`)
- sekwencje

  `setpgrp()→ioctl()→setpgrp()→ioctl()`
- intruz też musi korzystać z sekwencji
- problem

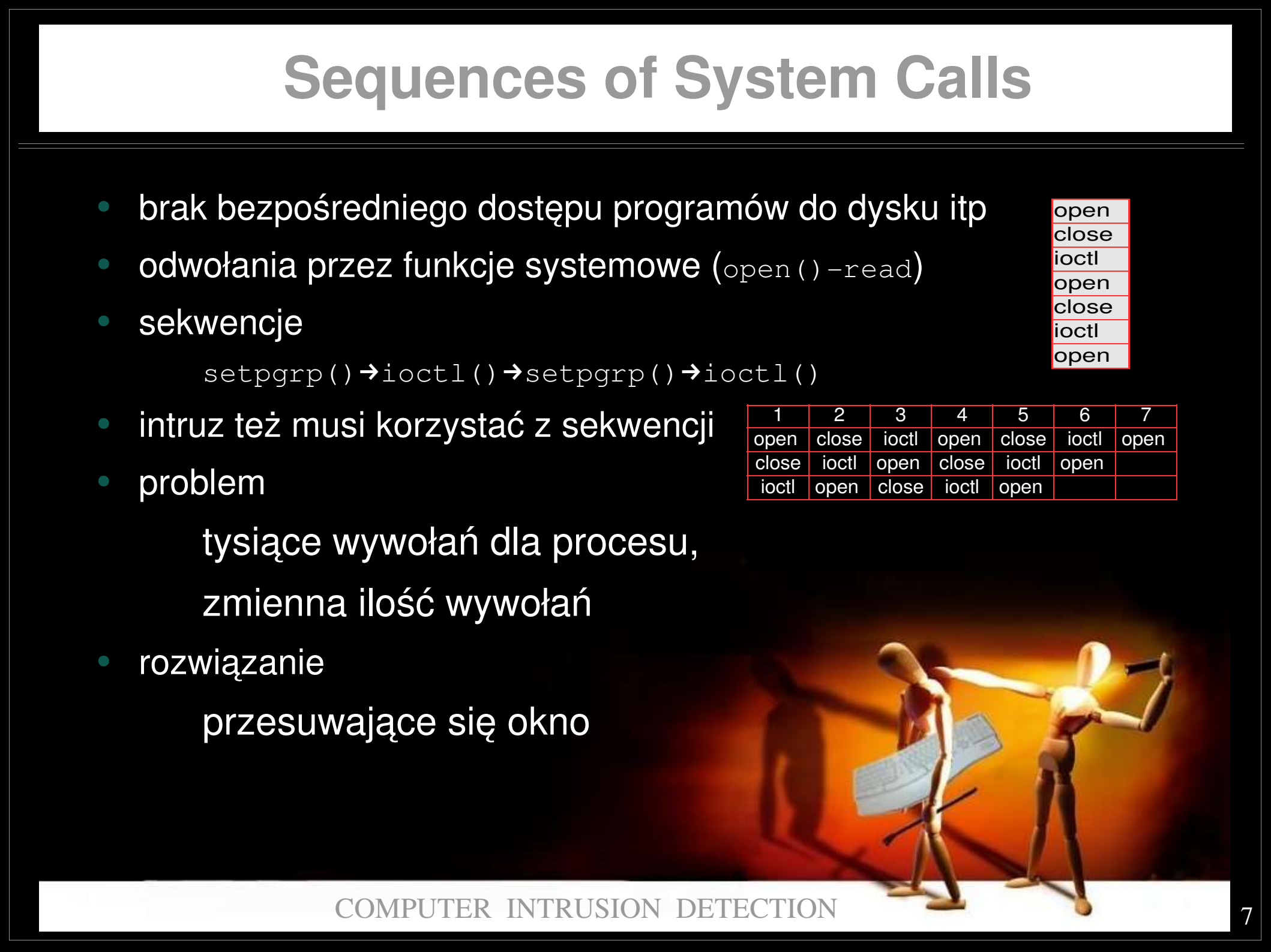
  tysiące wywołań dla procesu,

  zmienna ilość wywołań
- rozwiązanie

  przesuwające się okno

| open |
|---|
| close |
| ioctl |
| open |
| close |
| ioctl |
| open |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| open | close | ioctl | open | close | ioctl | open |
| close | ioctl | open | close | ioctl | open | |
| ioctl | open | close | ioctl | open | | |

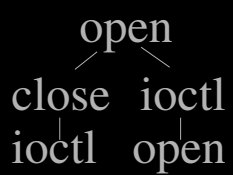


# Sequences of System Calls

- dla zwiększenia efektywności zastosowano drzewiastą strukturę danych
- redukcja rozmiaru danych

  szacowana ilość krotek: **O(N*k)**

  N-liczba normalnych zachowań, k-rozmiar okna




  w rzeczywistości rozmiar jest mniejszy (drzewa sekwencji)

  Forrest (baza dla SENDMAIL)

  1318 unikalnych sekwencji długości 10

  7578 węzłów w lesie (zbiorze drzew sekwencji)

  bez drzew: 13180 węzłów

# Sequences of System Calls

- tworzona baza normalnych zachwań systemu (fragmenty sekw.)
- nowe sekwencje są testowane, obliczana jest ilość niezgodności
- w praktyce niemożliwe jest zebranie bazy wszystkich zachowań normalnych-trzeba trochę poluzować warunek niezgodności
- obliczane są odległości Hamminga (ilość niezgodności w oknie) by sprawdzić jak odległa jest dana sekwencja od normalnej
- obliczana jest minimalna odległość Hamminga:
  $d_{min}(i)=\min\{d(i,j)$ dla każdej normalnej sek. $j\}$,
  czyli sprawdzamy jak odległa jest dana sekwencja od 'bazy'
- klasy zachowań:
  normalne,
  legalne (dopuszczalne),
  nienormalne

# Sequences of System Calls

- aby wykrć wtargnięcie

  **przynajmniej jedna** sekwencja musi być uznana za anomalię, czyli ma $\mathbf{d_{min}>0}$ (im większe dmin, tym większa szansa że to intruz)

  obliczana jest maksymalna $d_{min}$ dla sesji:

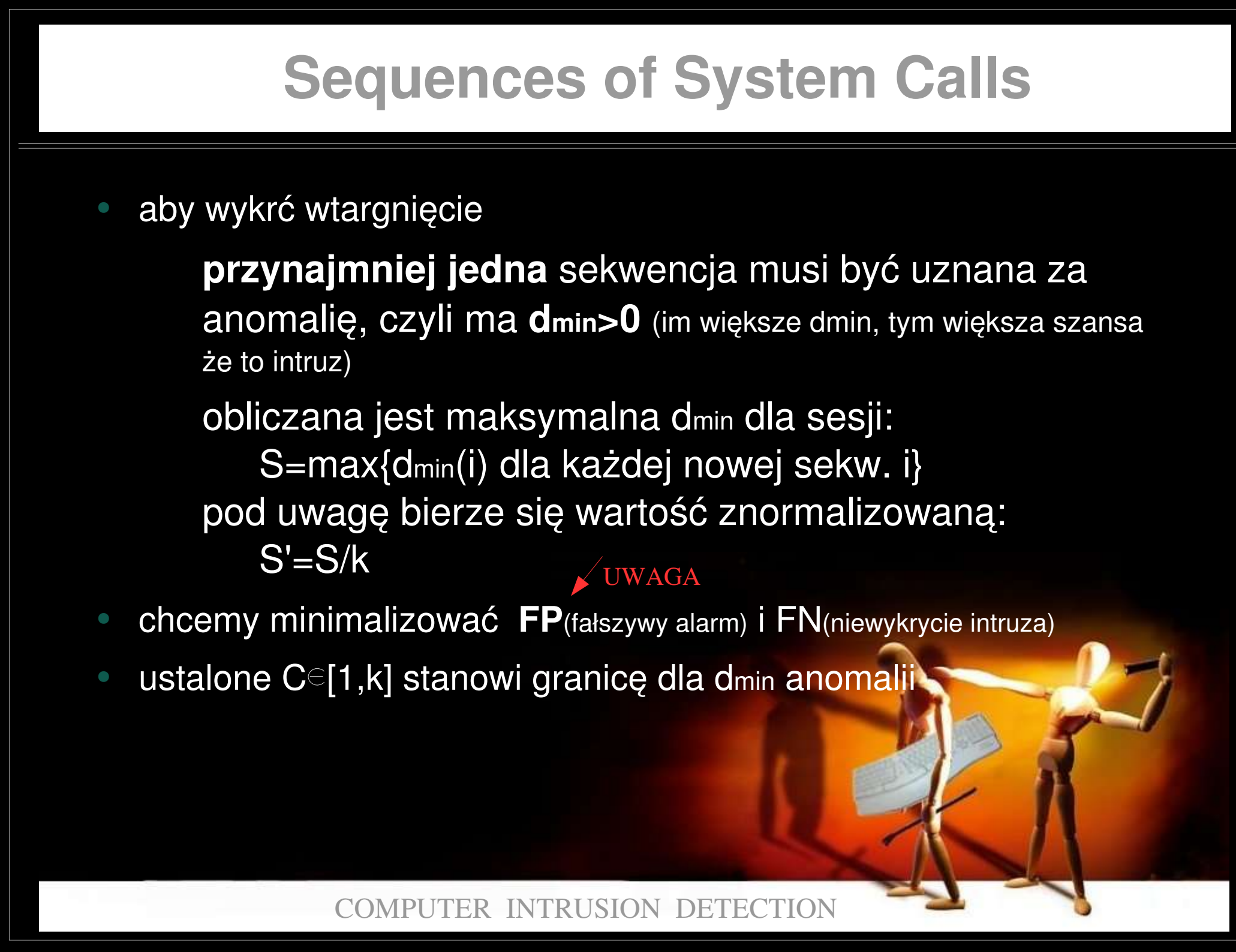
  $S=\max\{d_{min}(i)$ dla każdej nowej sekw. $i\}$

  pod uwagę bierze się wartość znormalizowaną:

  $S'=S/k$

UWAGA

- chcemy minimalizować **FP**(fałszywy alarm) i FN(niewykrycie intruza)
- ustalone $C \in [1,k]$ stanowi granicę dla $d_{min}$ anomalii



# Sequences of System Calls

- SENDMAIL vs INNE PROCESY
  dla k<6 -trudno rozpoznawalne

  k>30 złożone obliczeniowo

  k=10 ——————————►
- Wtargnięcia

udane ataki

| atak | # niezgodności | %niezgodności | S' |
|---|---|---|---|
| syslogd local | 248 | 17 | 0.7 |
| syslogd remote | 539 | 30 | 0.7 |
| sunsendmailcp | 92 | 25 | 0.6 |
| decode | 7 | 1 | 0.2 |
| lprcp | 242 | 9 | 0.5 |

nieudane ataki

| atak | # niezgodności | %niezgodności | S' |
|---|---|---|---|
| sm565a | 54 | 22 | 0.6 |
| sm5x | 472 | 33 | 0.6 |
| forward loop | 86 | 16 | 0.5 |

| proces | # niezgodności | %niezgodności | S' |
|---|---|---|---|
| ls | 42 | 75 | 0.6 |
| ls -l | 134 | 91 | 1 |
| ls -a | 44 | 76 | 0.6 |
| ps | 539 | 97 | 0.6 |
| ps -ux | 1123 | 99 | 0.6 |
| finfer | 67 | 83 | 0.6 |
| ping | 41 | 57 | 0.6 |
| ftp | 271 | 90 | 0.7 |
| pine | 430 | 77 | 1 |

Ergo: można zauważyć różnice w zachowaniu

# Sequences of System Calls

- False positives

  testy przeprowadzono dla `lpr` (dane z MIT i UNM)

  okno: k=10

  C=4 ($d_{min}(i)>C$)

| normal | test | FP/day | FP-job |
|---|---|---|---|
| 700 | 2066 | 2 | 1/100 |
| 1400 | 1366 | 1 | 5/1000 |

znalezione powody FP:

duże zadania przepełniały dysk,

próby druku dla /dev/printer/

-nie ma takiego urządzenia

# Network Intrusion Detection

- WWW – serie pakietów
- TCP/IP

  IP - bez potwierdzenia,

  źródło IP, cel IP

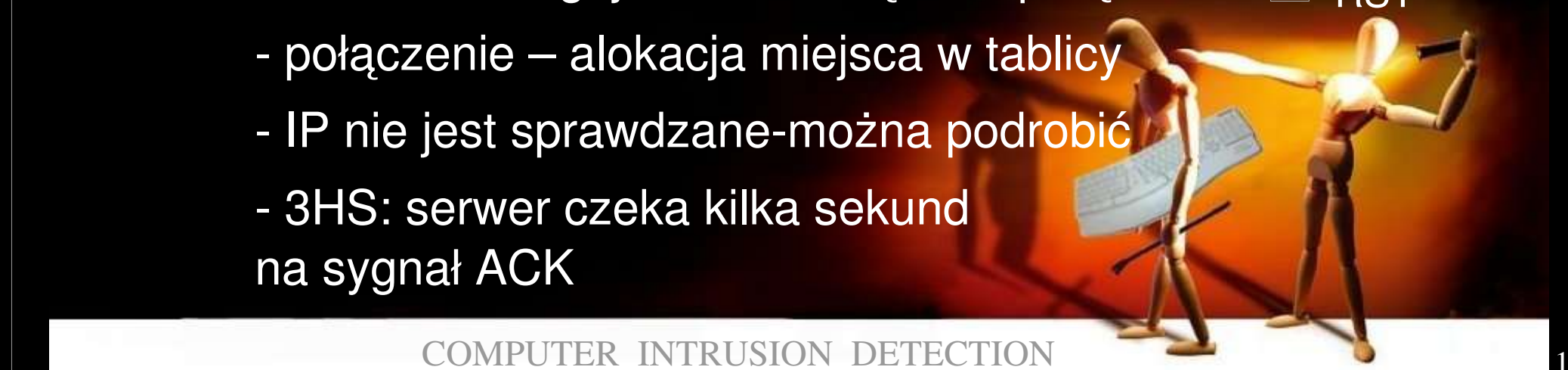
  TCP - potwierdzenie, 2 kanały,
  porty (identyfikacja aplikacji), flagi (3HS), sekwencje
- atak

  - serwer obsługuje skończoną ilość połączeń

  - połączenie – alokacja miejsca w tablicy

  - IP nie jest sprawdzane-można podrobić

  - 3HS: serwer czeka kilka sekund
  na sygnał ACK

# Network Intrusion Detection

- Network Sensor
  - analizuje pakiety
  - bezpośrednio przed firewllem
  - pozwala dostrzegać nieobsłużone połączenia (*backscatters)*
  - obserwuje tylko te *backscatters* które należą do sieci
- Streaming Data
  - ogromne ilości danych
  - większość metod statystycznych działa dla ustalonego rozmiaru danych (n)
  - tutaj ciągle napływają nowe (modele on the fly)
  - trudno ustalić średni przepływ danych

# Network Intrusion Detection

- tworzone są profile
  - użytkowników
    - autoryzacja tylko przy podawaniu hasła
    - profilowanie sprawdza użytkownika w trakcie pracy
  - aplikacji
- analizowane
  - user command sequences (UNIX)
  - window titles (WINDOWS) (tytuł, email, www)
  - sposób pisania (ssh)

    (może działać przy hasłach,
    w czasie pracy-niekoniecznie)

# Network Intrusion Detection

- Przykładowe dane dla uzytkowników Windows NT

| user | session | login length | 1st app | last app | #apps | #wins | #titles |
|---|---|---|---|---|---|---|---|
| u1-h19 | 3 | 30 794 | msoffice | msoffice | 6 | 13 | 134 |
| u1-h19 | 5 | 28 788 | msoffice | msoffice | 8 | 15 | 194 |
| u1-h19 | 6 | 19 902 | msoffice | msoffice | 10 | 25 | 267 |
| u1-h5 | 1 | 3 472 | explorer | explorer | 3 | 6 | 34 |
| u19-h10 | 6 | 16 886 | msoffice | msoffice | 7 | 8 | 133 |
| u8-h6 | 4 | 1 207 | outlook | outlook | 5 | 7 | 166 |

u-user
h-host

najpopularniejsze tytuły

| # | #session | Window title |
|---|---|---|
| 7002 | 425 | Inbox-Microsoft Outlook |
| 2525 | 411 | Program Manager |
| 2188 | 215 | Microsoft Word |
| 792 | 126 | Netscape |
| 704 | 156 | Print |

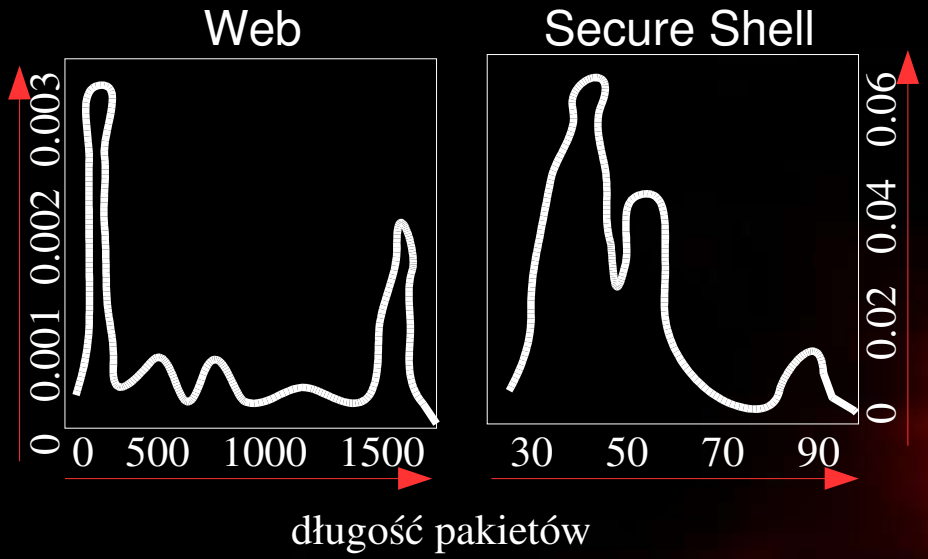
# Network Intrusion Detection

- metody wnioskowań

  - przesyłane dane np. w czasie godziny tworzą charakterystykę aplikacji (WWW, ssh)

  - analizować można wykresy, wizualizacje

  - dane (user, sesja, długość sesji ... ) - analizowano za pomocą różnych metod statystycznych




# Uwagi

- SOM

  nie pamięta kontekstu, więc kolejka(sekwencje połączeń)

- SVM vs NN

  SVM: praktyczne dla dużych danych, szybkie (nauka, działanie), skalowalne, czy atak

  NN: wykazano skuteczność w rzeczywistości, jaki atak

- RSN

  MLP nie pamięta kontekstu, RSN-tak

- Feature Transform

  wyrzuć nieistotne cechy

  znajdź funkcję fransformującą dane (do nowej przestrzeni) (np. przy uzyciu NN)

  zastosuj alg. klastrowania (np. k-means)

- Uwaga: skoordynowane ataki zaburzają wyniki

# Literatura

- **Network Intrusion Detection** *D.J.Marchette*
- **Intrusion Detection Using Neural Networks and Support Vector Machines** *S.Mukkamala, G.Janoski, A.Sung*
- **Dynamic Intrusion Detection Using Self Organizing Maps** *P.Lichodziejewski, A.Nur Zincir-Heywood, M.I. Heywood*
- **Analisis Techniques for Detcting Coordinated Attacks and Probes** *J.Green, D.Marchette, S.Northcutt, B.Ralph*
- **Intrusion Detection Based on Feature Transform Using Neural Networks** *Wonil Kim, Se-Chang Oh, Kyoungro Yoon*
- **Intrusion Detection Using Sequences Of System Calls** *S.A.Hofmeyer, S.Forrest, A.Somayaji*

Dziękuję za uwagę