Prezentacja:

Najbardziej popularne metody włamań

Aleksander Grygiel

Plan prezentacji

- Skanery portów
- Ataki przez przepełnienie bufora
- Ataki z wykorzystaniem dowiązań w /tmp
- Ataki odmowy dostępu

Skanowanie portów

- W sieciach TCP/IP każdy komputer udostępnia usługi poprzez „porty” o numerach 0-65535. Typowe usługi to np.: telnet (23), ftp (21), http (80)
- Każda usługa udostępniana przez serwer stanowi potencjalną lukę w bezpieczeństwie systemu
- Skanerów używa się do zbadania, które porty na zdalnym komputerze są otwarte
- Dodatkowo skanery portów potrafią też określić, jaki jest system operacyjny na zdalnej maszynie

Otwieranie połączenia w protokole TCP/IP

„Three way handshake”

- Aby nawiązać komunikację, klient wysyła do serwera pakiet SYN
- Serwer odpowiada pakietem SYN/ACK
- Po jego otrzymaniu klient wysyła pakiet ACK

Wówczas połączenie zostaje nawiązane

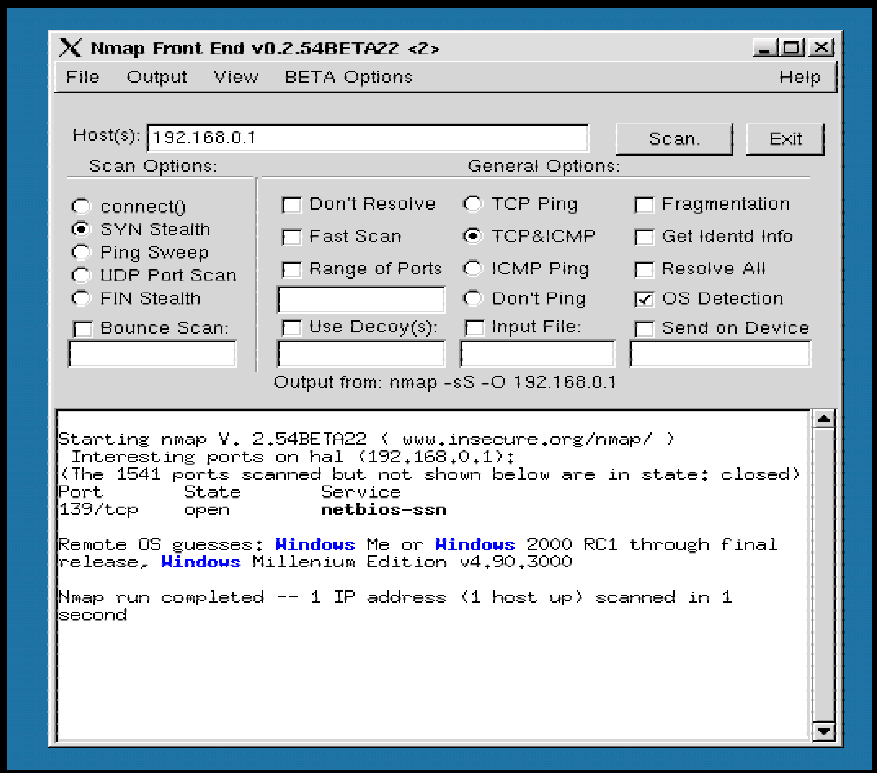
Zasada działania skanerów portów

- Skaner próbuje nawiązać połączenie ze zdalną maszyną na kolejnych portach (wysyła pakiety SYN)
- Gdy otrzyma pozytywną odpowiedź (SYN/ACK), nie kończy procedury otwierania połączenia (nie wysyła potwierdzającego ACK)
- Połączenie nie zostaje otwarte. Często skanowana maszyna nie rejestruje żadnych komunikatów o skanowaniu
- Na podstawie różnic w implementacjach protokołu TCP/IP, skanery potrafią rozpoznać system operacyjny

Zaawansowane metody skanowania

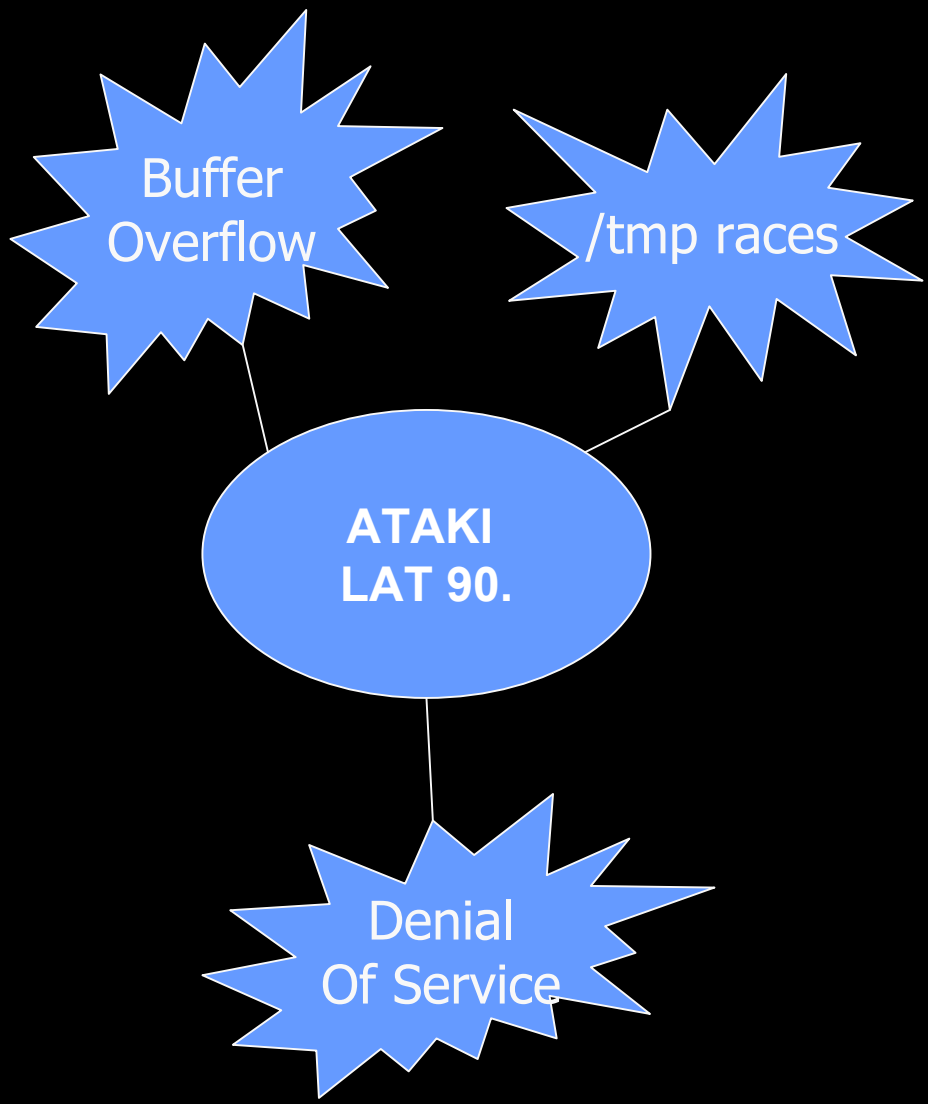
- Za pomocą pakietów FIN (nie działa na maszyny Windows)
- Przy użyciu „fragmentowanych” pakietów
- Przy użyciu pakietów ze sfałszowanym adresem źródłowym (tzw. skanowanie ip.id)
- Rozproszone skanowanie

Popularny skaner portów: nmap



„nmap” to bardzo popularny skaner portów. Jest dostępny w standardych dystrybucjach Linuxa. Posiada nawet graficzny interfejs dla X Windows (nmapfe)

Metody włamań w latach 90.

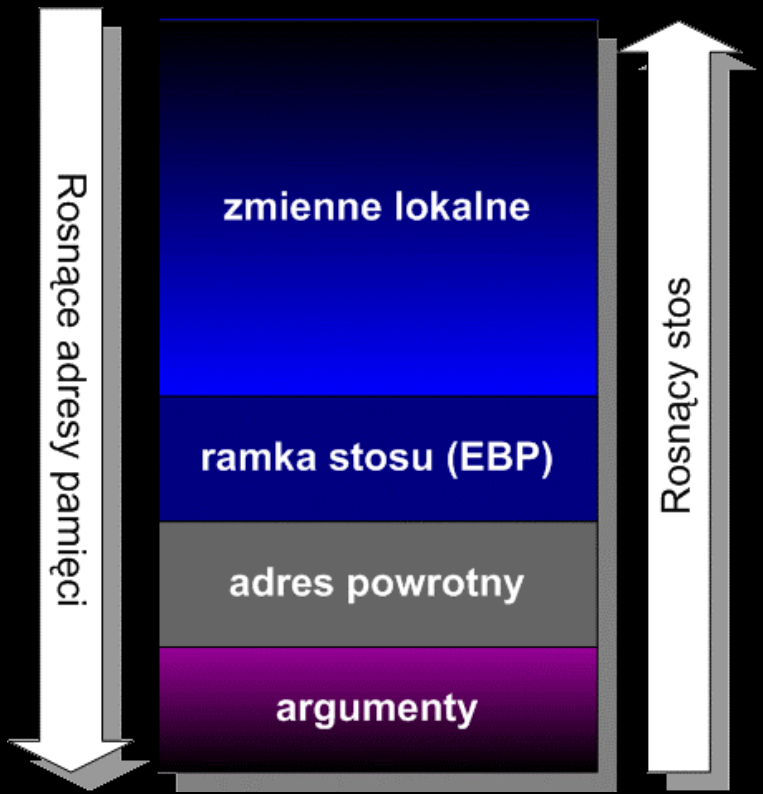


- Przepełnienie bufora (Buffer Overflow)
- Ataki na katalog /tmp
- Ataki „odmowy dostępu” (Denial of Service)

1. Przepełnienie stosu (buffer overflow)

- Podatne programy: źle lub niestarannie napisane programy SUID
- Na co pozwala taki atak: uruchamianie dowolnego kodu z prawami roota
- Metoda spopularyzowana przez artykuł w czasopiśmie Phrack pt. „Smashing The Stack For Fun And Profit”

Kiedy można nadpisać stos




```
int zrób_coś(char* text){
   char buf[100];
   strcpy(buf, text);
   ...
}
```

Po wywołaniu funkcji:
-na stos kładzione są argumenty funkcji
-następnie adres powrotny
-adres ramki stosu (EBP)
-na koniec odkładany jest obszar na zmienne lokalne

Jak to wykorzystać?

Najprostszy przypadek wykorzystania błędu BO:

- Kompilujemy do postaci maszynowej fragment kodu uruchamiający powłokę (shell)
- Generujemy „zabójczy napis” o długości przekraczającej rozmiar bufora w wadliwej funkcji
- Przekazujemy taki napis do funkcji. Adres powrotny, leżący w pamięci za buforem, zostaje nadpisany wskaźnikiem do naszego kodu uruchamiającego shell
- Zostaje uruchomiona powłoka z uprawnieniami administratora (jeżeli program miał atrybut SUID)
- Kod uruchamiający shell można umieścić w samym „zabójczym napisie” albo np. w zmiennej środowiskowej

Popularne funkcje C podatne na przepełnienie

- strcpy
- scanf
- gets (patrz: man gets)
- sprintf

Jak częste są błędy BO ?

realsh.c – Notatnik

Plik Edycja Wyszukaj Pomoc

```
{
  struct varslot *v;
  if ((v = find(name)) == NULL || v->name == NULL)
    return(NULL);
  return(v->val);
}  /*EVget*/


BOOLEAN EVinit ()                /* initialize symbol table from environment*/
/*
        Funkcja tworzy zmienne odpowiadajace zmiennym srodowiska
*/

{
  int i, namelen;
  char name[20];

  for (i=0; environ[i] != NULL; i++) {
    namelen = strcspn(environ[i], "=");
    strncpy(name, environ[i], namelen);
    name[namelen] = '\0';
    if (!EVset(name, &environ[i][namelen+1]) || !EVexport(name))
      return(FALSE);
  }
  return(TRUE);
}  /*EVinit*/


BOOLEAN EVupdate ()              /* build environment from symbol table */

/*
        Funkcja tworzy srodowisko z eksportowanych zmiennych
```

Zadanie zaliczeniowe nr 1: realsh

Gdzie wykrywano błędy BO

- programy pocztowe Microsoftu
- apache
- pine
- oracle
- kerberos
- samba
- xterm i inne aplikacje X Windows
- routery Cisco
- firewalle (Check Point)
- itd...

Ochrona przed atakami typu BO

- Ograniczenie liczby programów z atrybutem SUID do minimum
- Bezpieczne kompilatory C (z kontrolą przepełnienia bufora)
- Zablokowanie wykonywania kodu w segmencie stosu (odpowiednia łata na jądro systemu)

2. Ataki z wykorzystaniem katalogu /tmp

- Napastnik odgaduje nazwę pliku tymczasowego, który będzie stworzony przez proces innego użytkownika (np. roota) w katalogu /tmp
- W /tmp tworzy dowiązanie o takiej właśnie nazwie do któregoś z kluczowych plików systemowych, takich jak: /etc/passwd, rhosts itp.
- Proces roota, pisząc do pliku tymczasowego, faktycznie nadpisuje /etc/passwd

3. Ataki odmowy dostępu (DoS)

- Celem takich ataków jest zablokowanie niektórych usług albo zawieszenie atakowanego serwera
- Wykorzystuje się błędy w implementacji programów dostarczających usługi sieciowe (http, ftp, a także np. serwery Quake2)...
- ...jak również błędy w implementacji samego protokołu TCP/IP

Przykłady ataków DoS

- Stare wersje najpopularniejszego serwera www, Apache 1.2.4, nie sprawdzały długości nazwy pliku w poleceniu pobrania (GET nazwa_pliku).
- Każdy znak '/' w nazwie pliku wydłużał znacząco czas trwania operacji (odszukanie pliku na dysku).
- Prosty atak, powodujący dość znaczne spowolnienie serwera:

  GET ////... 7kb .../////

Znane błędy w implementacjach TCP/IP

- Ping of death
- SYN flood
- teardrop
- land / la terra
- smurf

SYN flood

- Napastnik wysyła bardzo dużo pakietów SYN do ofiary
- Atakowany komputer dla każdego nadchodzącego pakietu SYN próbuje utworzyć połączenie
- Powstają „pół-otwarte” połączenia (half-open): czekają na odpowiedź od napastnika, która jednak nigdy nie nadejdzie
- Powoduje to poważne spowolnienie albo całkowite zablokowanie ofiary

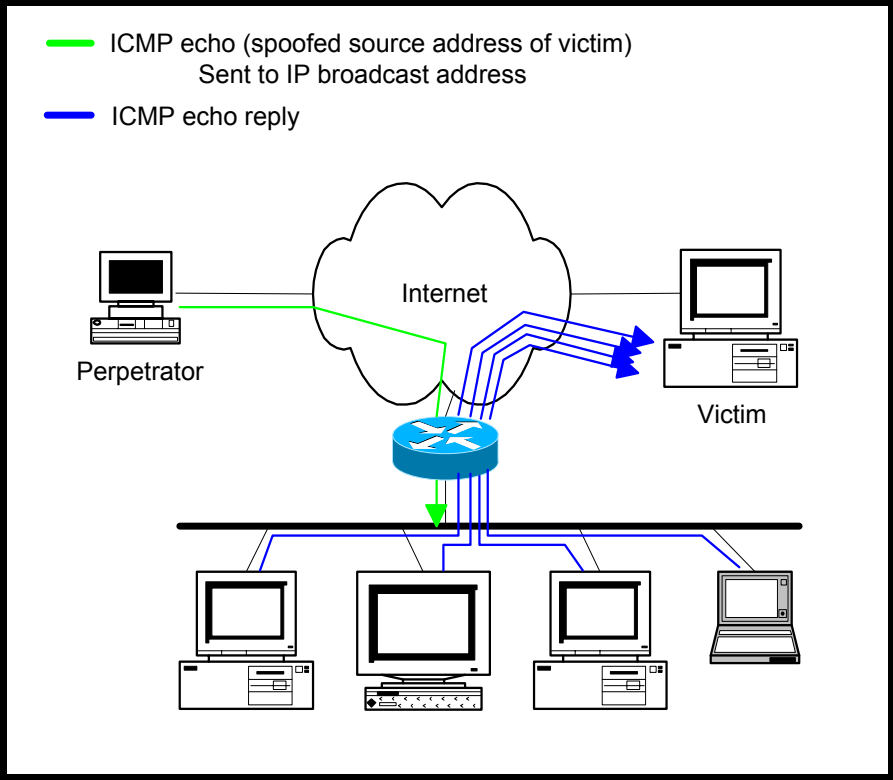
Land

- Napastnik wysyła pakiet ACK zawierający sfałszowany adres źródłowy równy adresowi docelowemu
- Atakowany host usiłuje nawiązać połączenie z samym sobą
- Powoduje to zawieszenie maszyny

Teardrop

- Błąd w implementacji stosu TCP/IP, występował zarówno w systemach Windows, jak i Linux/Unix
- System nie radził sobie z pofragmentowanymi pakietami
- Odpowiednio spreparowane i pofragmentowane pakiety, przy łączeniu w całość, powodowały nadpisanie jądra systemu
- Firewalle bezradne: albo przepuszczały zabójcze pakiety, albo same się wykładały

Smurf



- Napastnik wysyła pakiety ICMP echo/reply typu broadcast do sieci pośredniczącej w ataku
- Fałszuje adres IP, podszywając się pod adres ofiary
- Ofiara zalana odpowiedziami na pinga

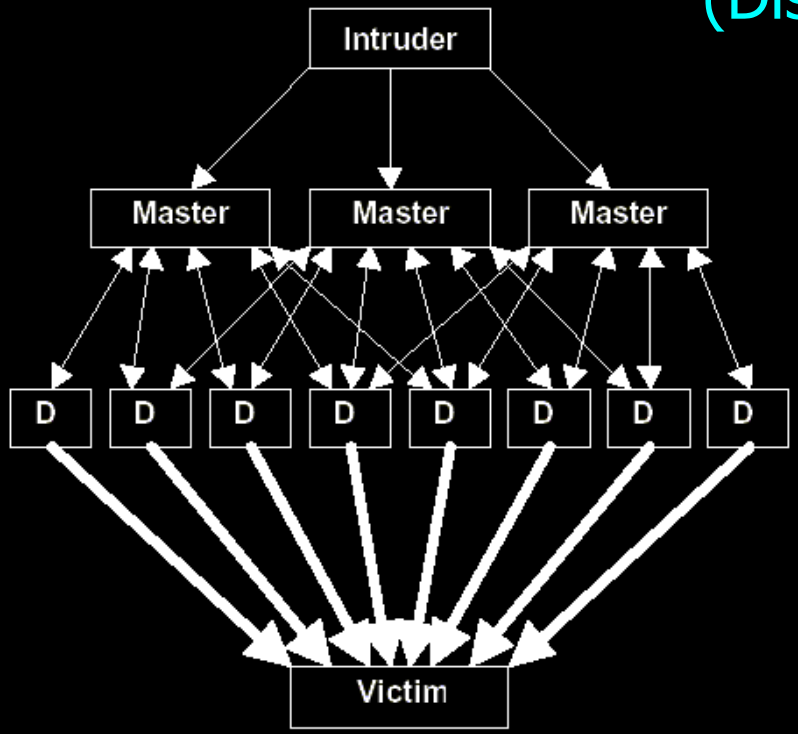
Ochrona przed DoS

- Ochrona przed atakami DoS jest bardzo trudna – każdy z nich jest inny i opiera się na specyficznych błędach w oprogramowaniu.
- Należy ograniczać do niezbędnego minimum usługi sieciowe oferowane przez maszynę.
- Stałe śledzenie listy bugtraq oraz innych list poświęconych bezpieczeństwu i błędom w oprogramowaniu.

  (np.: www.securityfocus.com, www.ussrback.com)

Nowa odmiana tej metody: DDoS

Rozproszony atak odmowy dostępu
(Distributed Denial of Service)



- Siłowy atak, polegający na zasypaniu ofiary lawiną pakietów
- Utrudnia identyfikację napastników
- Bardzo skomplikowany, stosowany przeciwko dużym serwisom internetowym (np. serwery DNS, Yahoo, itp.)