Bezpieczeństwo w komputerze

Dominik Sidorek

12 stycznia 2007

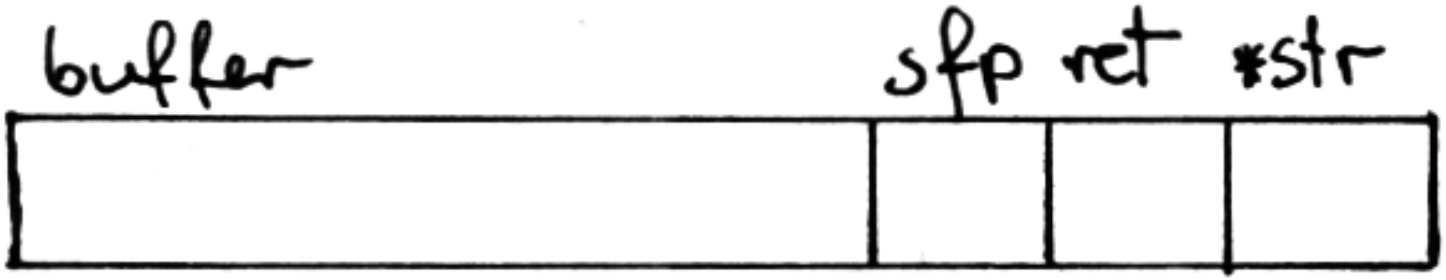
Jak przepełniamy bufor?

Przykład cacka z dziurką:

```
void function(char *str) {
    char buffer[16];

    strcpy(buffer,str);
}
```

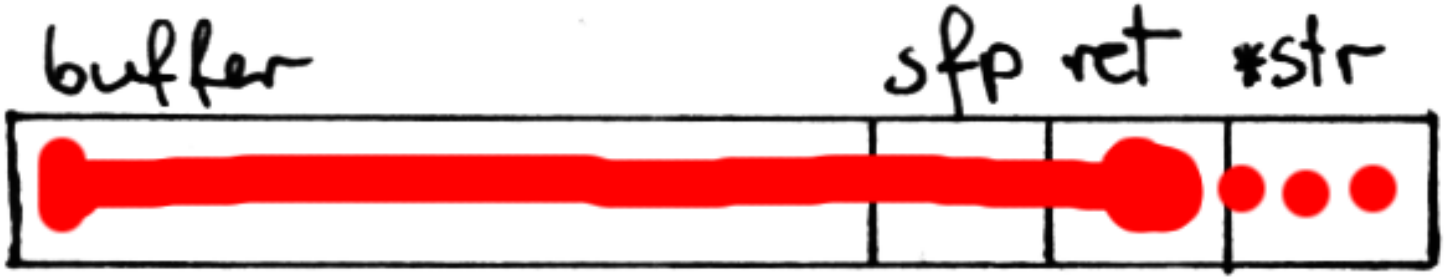
Stack frame:



Przepełniliśmy bufor!

Tak zwany "stack smash"

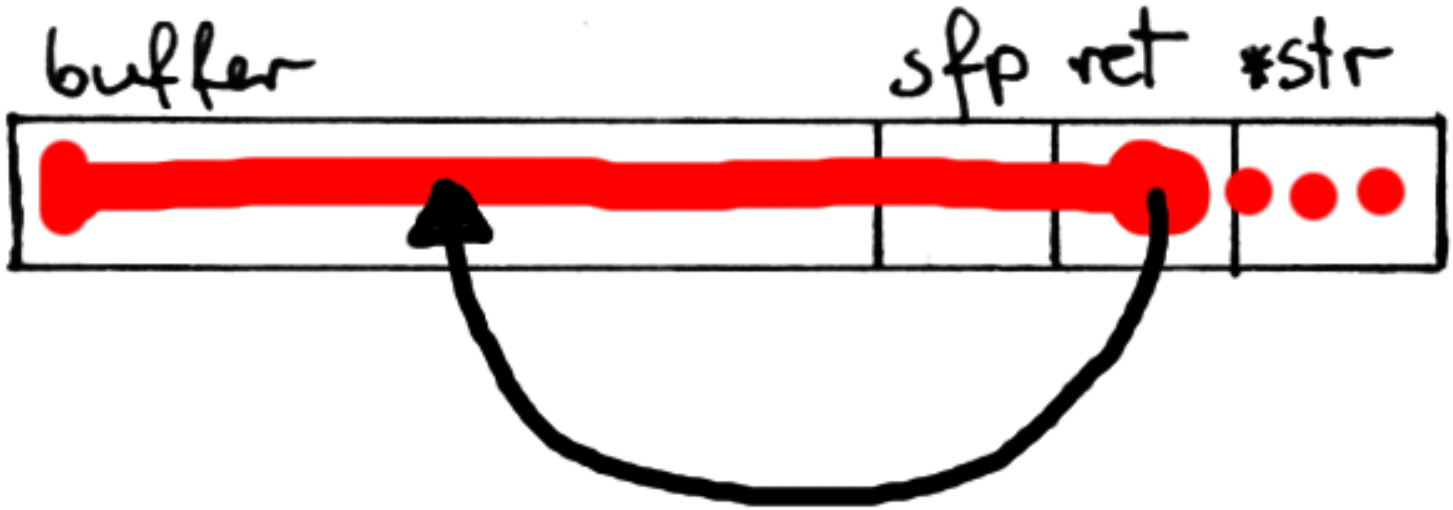
Przepełnienie bufora:



Mamy w ret to co chcemy!
Co dalej?

Atak typu "shellcode"

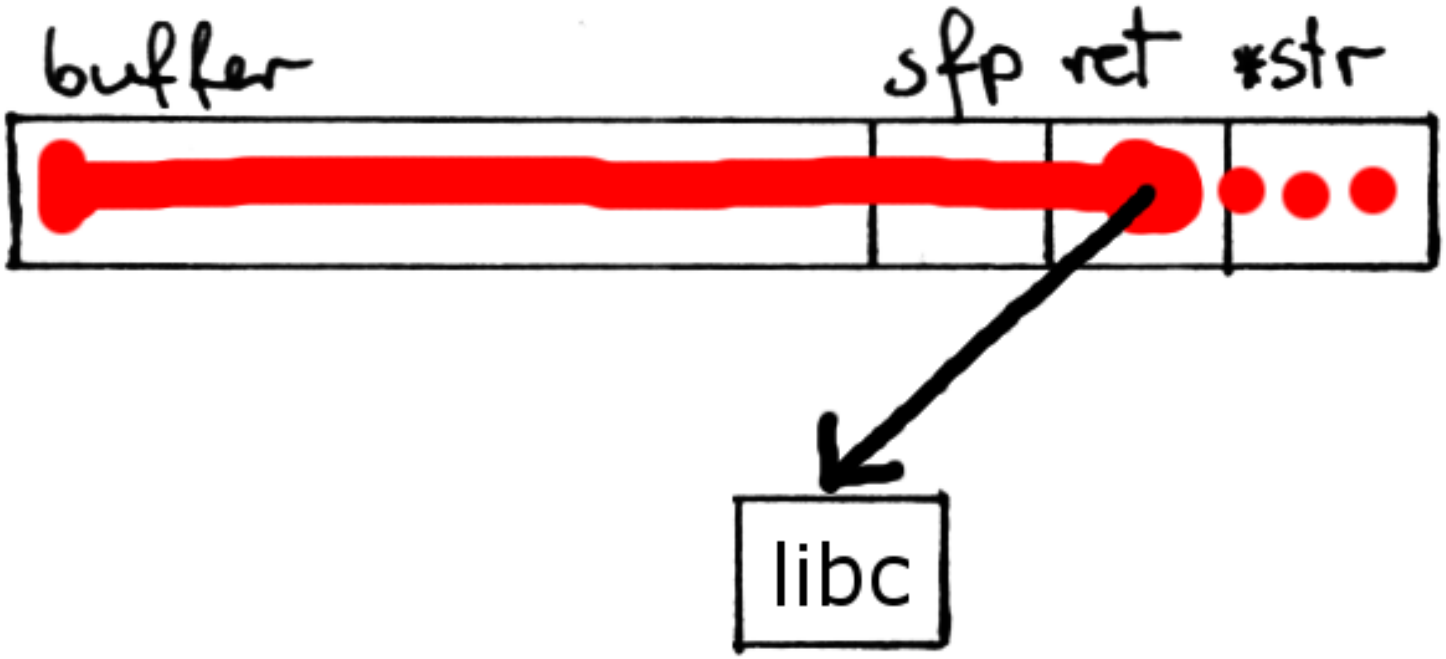
Wykonujemy nasz kod:



Na przykład spałnujemy shella:

```
void main() {
    char *name[2];
    name[0] = "/bin/sh";
    name[1] = NULL;
    execve(name[0], name, NULL);
```

Atak typu "return-to-libc"



Na przykład wywołujemy *system()*

Haczyki na krakera

- Adres powrotu w ret jest bezwzględny.
- Zamazaliśmy sfp.

Generowanie bezpiecznego kodu:

Język / biblioteki

- Uważać :)
  - Zalety: Małe wymagania, zabawne
  - Wady: Nieskuteczne
- Bezpieczny, wysokopoziomowy język
- Biblioteki np. STL
  - Zalety: Automatyczne, pewne
  - Wady: Gotowy kod, wyszkolenie progamistów, wydajność

Generowanie bezpiecznego kodu:

Język / biblioteki

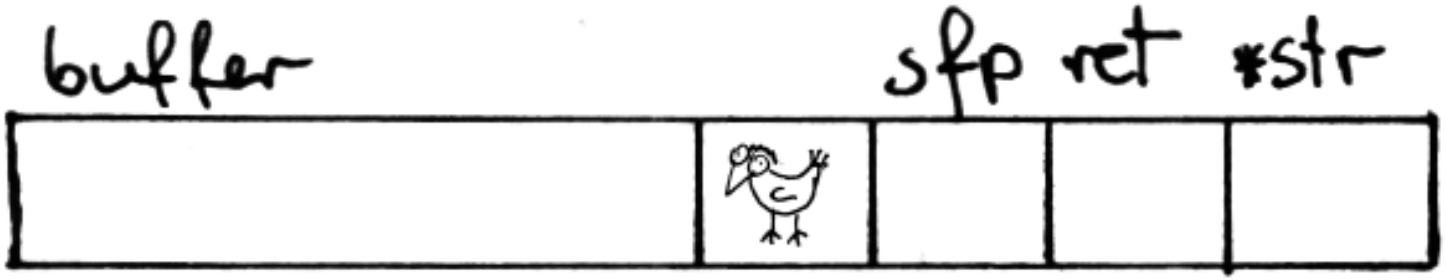
- Uważać :)
  - Zalety: Małe wymagania, zabawne
  - Wady: Nieskuteczne
- Bezpieczny, wysokopoziomowy język
- Biblioteki np. STL
  - Zalety: Automatyczne, pewne
  - Wady: Gotowy kod, wyszkolenie progamistów, wydajność

Praktyka pokazuje, że są to mało praktyczne rozwiązania.

Generowanie bezpiecznego kodu:

Na poziomie kompilacji: Kanarki

A oto kanarek:



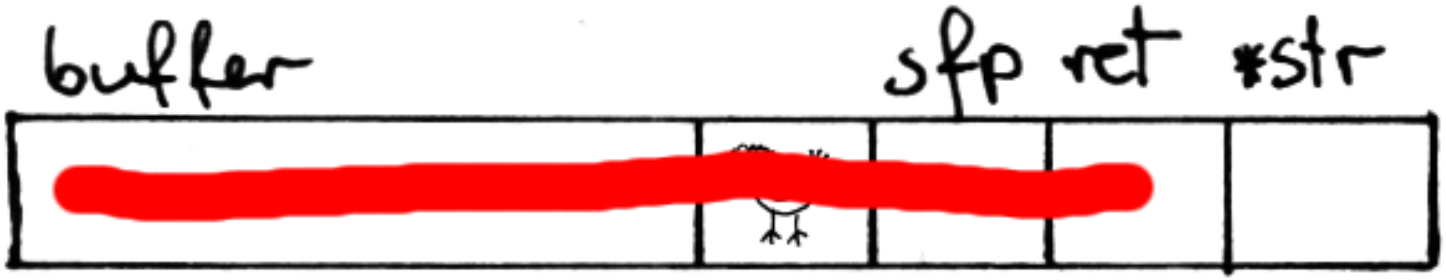
Kanarek jest losowy lub (losowy XOR dane kontrolne)

- Zmniejsza wydajność
- Atak powoduje zakończenie

Generowanie bezpiecznego kodu:

Na poziomie kompilacji: Kanarki

A oto kanarek:



Kanarek jest losowy lub (losowy XOR dane kontrolne)

- Zmniejsza wydajność
- Atak powoduje zakończenie

Zabezpieczenia systemowe

ASLR

ASLR to:

- Space Layout Adress Randomization

Losowo zmieniać można:

- mmap() base
- Stack base
- Heap base

Zabezpieczenia systemowe

ASLR

ASLR to:

- Space Layout Adress Randomization
- Randomization Space Layout Adress

Losowo zmieniać można:

- mmap() base
- Stack base
- Heap base

Zabezpieczenia systemowe

ASLR

ASLR to:

- Space Layout Adress Randomization
- Randomization Space Layout Adress
- **A**dress **S**pace **L**ayout **R**andomization

Losowo zmieniać można:

- mmap() base
- Stack base
- Heap base

Zabezpieczenia systemowe

NX

- Non eXecutable (DX, W xor X)
- Bit oznaczający fragment pamięci jako dane, jest chroniony przed wykonaniem.
- Może być sprzętowy lub emulowany (np. PaX)
- Nie chroni przed return-to-libc

Implementacje

Pacze na gcc:

- ProPolice - kanarki, przestawianie pól w strukturach
- StackGuard - kanarki

Systemy operacyjne:

- PaX. Pacz na linuksa, ASLR, NX (także emuluje), także inne mechanizmy, bardzo rozbudowany
- Linux - od 2.6.12 obsługuje NX, ASLR
- OpenBSD, Solaris, Vista - NX, ASLR

Robaki

- *Robak* jest to program samoreplikujący przez sieć.
- Robak to nie wirus; nie potrzebuje programu nosiciela
- Robak to nie trojan; nie udaje przydatnego programu
- Oczywiście granice bywają płynne

Sposoby mnożenia się

- E-mail i komunikatory
- Sieci P2P
- Samodzielnie przez sieć

Szkody powodowane przez robaki

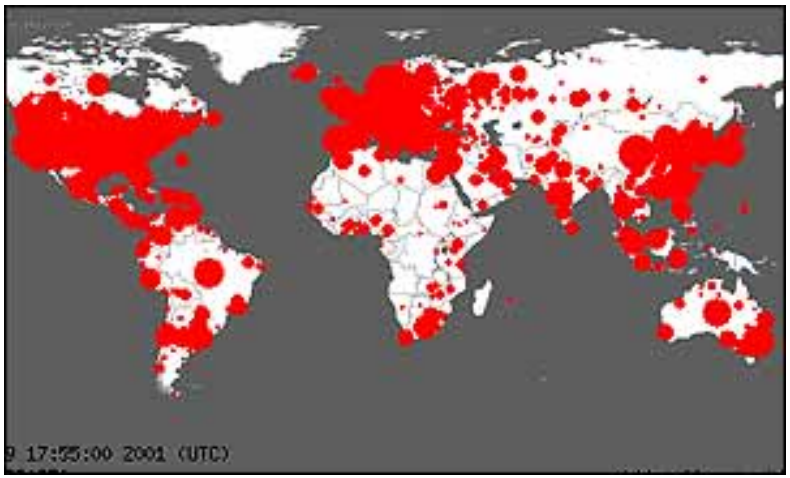
- Przeciążenie sieci i komputerów
- Lokalne szkody - kasowanie, kodowanie plików
- Tworzenie backdoorów
- Botnet - rozsyłanie spamu i ataki DOS

Morris Worm

2 Listopada 1988

- sendmail, finger, rsh, słabe hasła
- VAX / BSD4/SUN3
- 6000 zakażonych, ok. 10% internetu

Code Red



19 Lipca 2001

- buffer overflow w MS ISS
- w szczycie 359k systemów
- Modyfikował strony, próba DOS

SQL Slammer

25 Stycznia 2003

- buffer overflow w MS SQL Server / DE
- 376 bajtów; mieści się w pakiecie UDP
- 75k systemów w ciągu kilku godzin
- Przeciążył wiele routerów

IDS

- IDS - Intrusion Detection System
- NIDS - Network IDS
- PIDS - Protocol Based IDS
- APIDS - Aplication Specific PIDS

Network IDS

- System biernie analizujący ruch w sieci
- Wykrywa skanowanie portów, DOS, podejrzany duży ruch
- Może wykrywać niektóre próby włamań: shellcode, nop
- Aktualizacja czarnych list
- np: Snort

Protocol Based / Application Protocol IDS

- Świadome protokołu np: HTTP, nawet SQL
- Mogą wykryć nieprawidłowości niezauważalne prostą analizą pakietów

IPS

Także nazywane Application Level Firewall

- Proxy
- Analogicznie do typowego firewalla mogą filtrować na podstawie reguł