Bezpieczeństwo serwera WWW w praktyce

Paweł Krawczyk

kravietz@aba.krakow.pl

Fakty

- Polska — podmiana średnio raz na tydzień
  http://www.artur.pl/muzeum.html
- Ofiary — największe firmy polskie i zagraniczne
  NASK (Sylwester 1995/96), TPSA (1998), ENSI, Canal+, Unimil (2x), Wprost (2x), MON, BSI...
- świat — od kwietnia 2000 zmieniono 21 tys. stron, miesięcznie od 1-3 tys.
  http://alldas.de/
- Zastosowane techniki włamań — bardzo różne
- Ilość „cichych” włamań — nieznana

Fakty c.d.

- Systemy operacyjne — 66% Windows, 16% Linux
- Wirus Code Red — ok. 500 tys. zarażonych serwerów na świecie
- Polska — ok. 20 prób infekcji dziennie w małej sieci
- Atak *denial of service* na *whitehouse.gov*

Do czego dążymy?

- do bezpiecznego serwera WWW
- do bezpiecznej strony WWW

Podmiana strony nie musi nastąpić na serwerze!

Kulisy pracy przeglądarki

- człowiek wpisuje w okienku nazwę serwera
- przeglądarka szuka adresu IP w DNS
- przeglądarka łączy się adresem IP
- serwer zwraca odpowiedź (HTML)

Bezpieczeństwo po stronie klienta

- niezależne od nas
- pomyłki wykorzystywane przez cybersquatterów
  - gocities.com, ebsy.com, www.firma.co.pl
  - ochrona prawna (znaki towarowe, nieuczciwa konkurencja)
  - dyskusyjna kwestia „prawa do domeny” (*microsoft.pl*)

Ataki na DNS

- serwer DNS może być poza naszą kontrolą
- serwer WWW może być lepiej chroniony
- prościej **podmienić** adres IP serwera, niż go atakować

Metody

- DNS *cache poisoning*
- *social engineering*, „ataki” na InterNIC i NASK

Bezpieczeństwo DNS

- odporny na włamania serwer DNS
  - DJBDNS (www.djbdns.org)
  - BIND bez praw *roota*, *chroot*
  - odpowiednie prawa do plików domen
- odpowiednie wartości TTL, *expire*
- konflikt pomiędzy kontrolą nad serwerem a zasadą lokalizacji secondary w innej sieci
- kolokacja serwera secondary w zaprzyjaźnionej firmie

Ataki na otoczenie serwera

- ARP *spoofing*
- efekt podobny jak w atakach na DNS

Ataki na serwer

- błędy w samym serwerze
  - serwer działający z uprawnieniami superużytkownika
  - pozostawienie testowych CGI w `/cgi-bin/`
  - ataki typu `../`
- błędy we własnym oprogramowaniu
- błędy w skryptach PHP, ASP
- błędy w CGI
- błędy w konfiguracji
- nieużywanie *suexec*
- zbyt liberalne prawa do plików
- dostęp do poufnych zasobów
- ataki *denial of service*

Ataki ze strony lokalnych użytkowników

- kradzieże skryptów
- wyciąganie haseł ze skryptów

Ataki DoS

- ataki wykorzystujące błędy w oprogramowaniu

  zawieszenie serwera, spowolnienie pracy, zużycie zasobów . . .

- ataki sieciowe

  Smurf, *SYN flood*, *ping flood*

- obrona — w niektórych wypadkach bardzo trudna

Serwery inne niż Apache

- Netscape Enterprise

  niedawne dziury w rodzaju listowania katalogów

- Boa

  bardzo szybki, bezpieczny (www.boa.org)

- Microsoft IIS

Racjonalny wybór serwera

- funkcjonalność
  http://serverwatch.internet.com/webservers.html
- czy na pewno potrzebujemy Apache?
- czy na pewno potrzebujemy PHP?

Może wystarczy nam Boa?

Serwer Apache

- od kilku lat brak poważniejszych dziur
- działa z niskimi uprawnieniami
- rozbudowana kontrola dostępu do zasobów
- szerokie możliwości precyzowania ograniczeń
- program suexec

Bezpieczeństwo Apache

- wyłączenie `/~user`, jeśli na serwerze nie ma innych użytkowników
  `UserDir`
- limity na zasoby
  `RLimitMEM, RLimitCPU, RLimitNproc`
- wykorzystanie suexec, osobni użytkownicy dla każdego serwera wirtualnego
  `User, Group`
- ograniczenie CGI do wyznaczonego katalogu
  `ScriptAlias`

Bezpieczeństwo Apache c.d.

- zablokowanie dostępu do /, zezwolenie tylko do wyznaczonych katalogów
- wyłączenie *Server Side Includes*
- pozostawienie wyłącznie **niezbędnych** modułów
- zablokowanie dostępu do poufnych plików

  ```
  AddHandler ignore .htpasswd
  Action ignore .htpasswd
  ```
- lokalizacja plików z hasłami (`htpasswd`) poza dozwolonymi katalogami

Błędy we własnym oprogramowaniu

- skrypty i programy CGI
- skrypty przykładowe pozostawione po instalacji
  `phf`, `test.cgi`
- języki skryptowe wbudowane w serwer
  PHP, Perl, ASP
- błędy typowe dla oprogramowania
  - przepełnienie bufora
  - SQL injection
  - wykonanie obcego kodu

Typowy błąd

```
#!/bin/sh
...
ping $host 2>&1
...
```

A jeśli `host` to `www.costam.pl; rm -rf ~`?

Unikanie problemów z oprogramowaniem

- języki z natywnym *bounds checking* (*Ada95*)
- specjalne kompilatory
  StackGuard (www.immunix.org)
- inne wynalazki
  http://www.openwall.com/linux/, *libsafe*
- edukacja programistów
- audyty kodu (możliwość automatyzacji, narzędzia)

Ataki ze strony lokalnych użytkowników

- czy użytkownicy potrzebują dostępu do shella?
- prawa do katalogów i plików
- ochrona własności intelektualnej (znakowanie kodu, celowe błędy)

Ochrona integralności serwera

- wykrywanie włamań
  - systemy IDS (www.snort.org)
  - kryptograficzne sumy kontrolne plików (www.tripwire.com)
- sposoby proste, ale skuteczne (*crontab*)
- wykorzystanie mechanizmów systemowych
  atrybut *immutable*, *securelevel*

Serwer ABA-W3S

- zasada KISS (*Keep It Simple, Stupid*)
- minimum funkcjonalności
- wysoka wydajność
- serwer Boa
- częściowy audyt kodu, usunięte niepotrzebne fragmenty
- zmodyfikowany kernel Linuksa
- utrudnione skanowanie
- strony serwowane z CD
- w planach kolejne wersje

http://www.aba.krakow.pl/

Pytania?

Paweł Krawczyk
kravietz@aba.krakow.pl

http://ipsec.pl/