Hacking na żywo

Tomasz Węgrzanowski
CoolKon III

Hacking na żywo

- Bezpieczeńswo komputerowe
- Ważny element rzeczywistości
- Obraz w mediach mocno zafałszowany
- Społeczeństwo wierzy w mity na ten temat
  - Również gracze, pisarze, autorzy filmów
  - Jak też zdecydowana większość informatyków

Hacking na żywo

Wyjątek – The Matrix Reloaded

Hacking na żywo

- Bezpieczeństwo jest subiektywne
  - “To nie jest dziura”
- Polityka bezpieczeństwa
  - Spójne sformalizowane zasady
  - Jedna nieprzewidziana rzecz i ...
  - Zwykle wiele nieprzewidzianego
- Udawanie że problem nie istnieje

Hacking na żywo

The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts.

- Eugene H. Spafford

Hacking na żywo

- Kilka typów problemów dominuje
  - Przepełnienie bufora
  - Wstrzykiwanie SQL
- Jak też ludzkie
  - Stare niełatane oprogramowanie
  - Brak odpowiedzialności użytkowników
  - Nie uczenie się na błędach:
    - IE, sendmail, PHP, C

Hacking na żywo

- Komputery komunikują się wysyłając pakiety
- Na jednym komputerze działa jednocześnie wiele aplikacji sieciowych (przeglądarka, gadu gadu, bittorrent). Każda aplikacja ma wiele połączeń.
- Połączenia są identyfikowane kombinacją:
  - Adres nadawcy, port nadawcy
  - Adres lokalny, port lokalny

Hacking na żywo

- Pakiety TCP/IP:
  - SYN – prośba o zaakceptowanie połączenia
  - SYN+ACK – akceptacja połączenia
  - ACK – potwierdzenie otrzymania pakietu
  - RST – odrzucenie połączenia
  - Bez flag - Dane
  - FIN – zamknięcie połączenia

1	83.27.125.165	3179	209.85.135.99	80	SYN	
10	209.85.135.99	80	83.27.125.165	3179	SYN+ACK1	
2	83.27.125.165	3179	209.85.135.99	80	ACK10	

2	83.27.125.165	3179	209.85.135.99	80		GET / ...
11	209.85.135.99	80	83.27.125.165	3179	ACK2	
12	209.85.135.99	80	83.27.125.165	3179		200 OK ...
3	83.27.125.165	3179	209.85.135.99	80	ACK12	
13	209.85.135.99	80	83.27.125.165	3179		<a href=...
4	83.27.125.165	3179	209.85.135.99	80	ACK13	

14	209.85.135.99	80	83.27.125.165	3179	FIN	
5	83.27.125.165	3179	209.85.135.99	80	ACK14	
6	83.27.125.165	3179	209.85.135.99	80	FIN	
15	209.85.135.99	80	83.27.125.165	3179	ACK6	

Hacking na żywo

- tcpdump
- Podsłuchuje pakiety TCP/IP (i inne)
- Standardowe narzędzie każdego administratora
- Żeby podsłuchać ciekawe dane, trzeba kontrolować miejsce przez które płyną
  - Przejąć kontrolę nad takim miejscem
  - Przekonać serwery do zmiany trasy przesyłu

Hacking na żywo

- Wiele narzędzi podsłuchuje pakiety
- tcpflow
  - Zachowuje przesłane dane
- ettercap
  - Wszechstronne narzędzie podsłuchowe
  - Zbiera hasła ze stron internetowych, email, MSN, nawet do serwerów Quake 3 i Half-Life

Hacking na żywo

- Wiedza o systemie ułatwia atak i obronę
- Skanowanie portów za pomocą nmap
  - Próbuje połączyć się z wszystkimi portami po kolei
  - Tryb naiwny:
    - Jeśli połączenie zaakceptowane, natychmiast zamknij
  - Tryb półotwarty (SYN scan):
    - Wysyłamy SYN
    - Jeśli otrzymamy SYN+ACK, wysyłamy RST, port otwarty
    - Jeśli otrzymamy RST, port zamknięty
    - Wykrywamy bez otwierania pełnego połączenia

Hacking na żywo

- Komunikujące się komputery nie są bezpośrednio podłączone
- Gdy komputer otrzyma pakiet nie do siebie:
  - przekazuje go dalej
  - lub odrzuca
  - być może przy okazji podsłuchując
- Każdy pakiet może podążać inną drogą
  - przeważnie droga się nie zmienia
- traceroute

Hacking na żywo

- Gdy komputer otrzyma pakiet nie do siebie:
  - przekazuje go dalej
  - lub odrzuca
  - być może przy okazji podsłuchując
  - może też modyfikować komunikację
- Modyfikacja pakietów umożliwia dzielenie numeru IP (NAT)
- często stanowi część większego ataku

Hacking na żywo

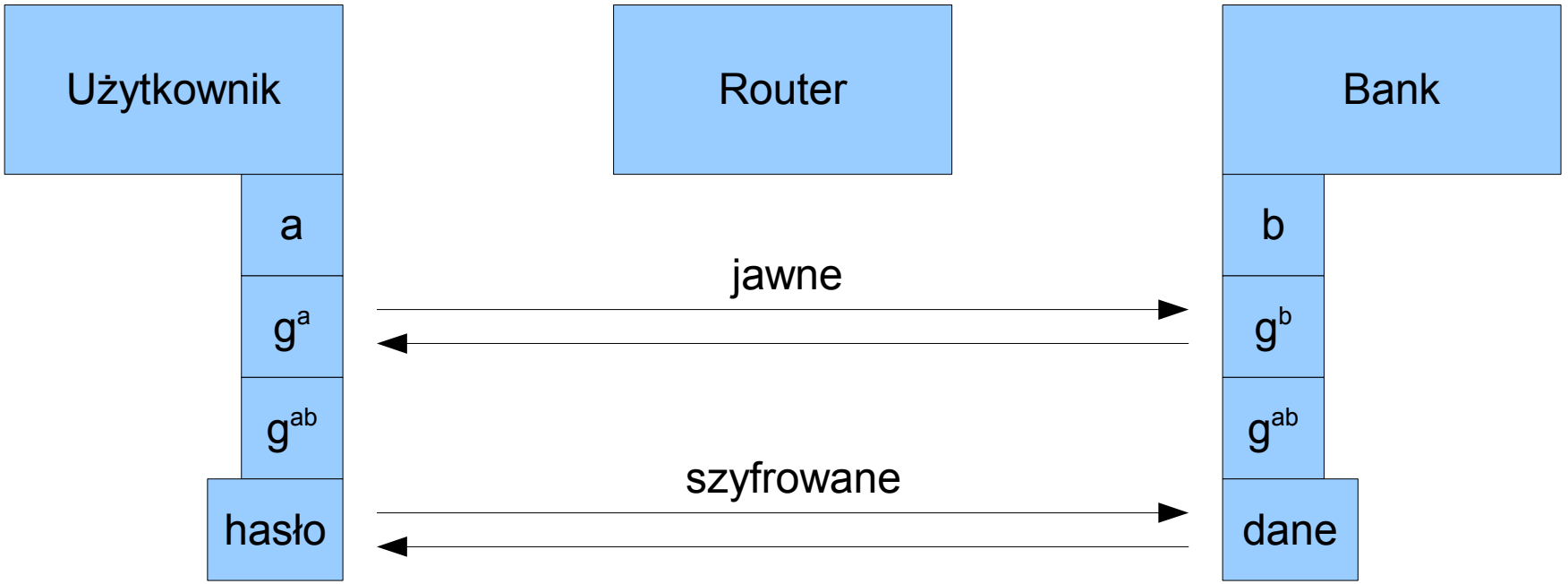
- Jeżeli używa się prawidłowo kryptografii, połączenia szyfrowane są praktycznie nie do złamania
- Atakowane są inne elementy systemu
- Kryptografii używa się przeważnie nieprawidłowo
- Jeśli strony ustaliły wspólny klucz, bez znajomości klucza nie można podsłuchać ani zmodyfikować komunikacji
- Wspólne klucze można łatwo generować

Hacking na żywo

- Generacja klucza
  - *g*, *p* – publicznie znane liczby
  - *a*, *b* – prywatne liczby, losowe co połączenie
  - x, y – wysyłane jawnie
  - żeby znać *k* trzeba znać *a* i *b*, *a* i *y*, lub *b* i *x*
  - obliczenie *a* z *x*, *b* z *y*, lub *k* z *x* i *y* zbyt trudne

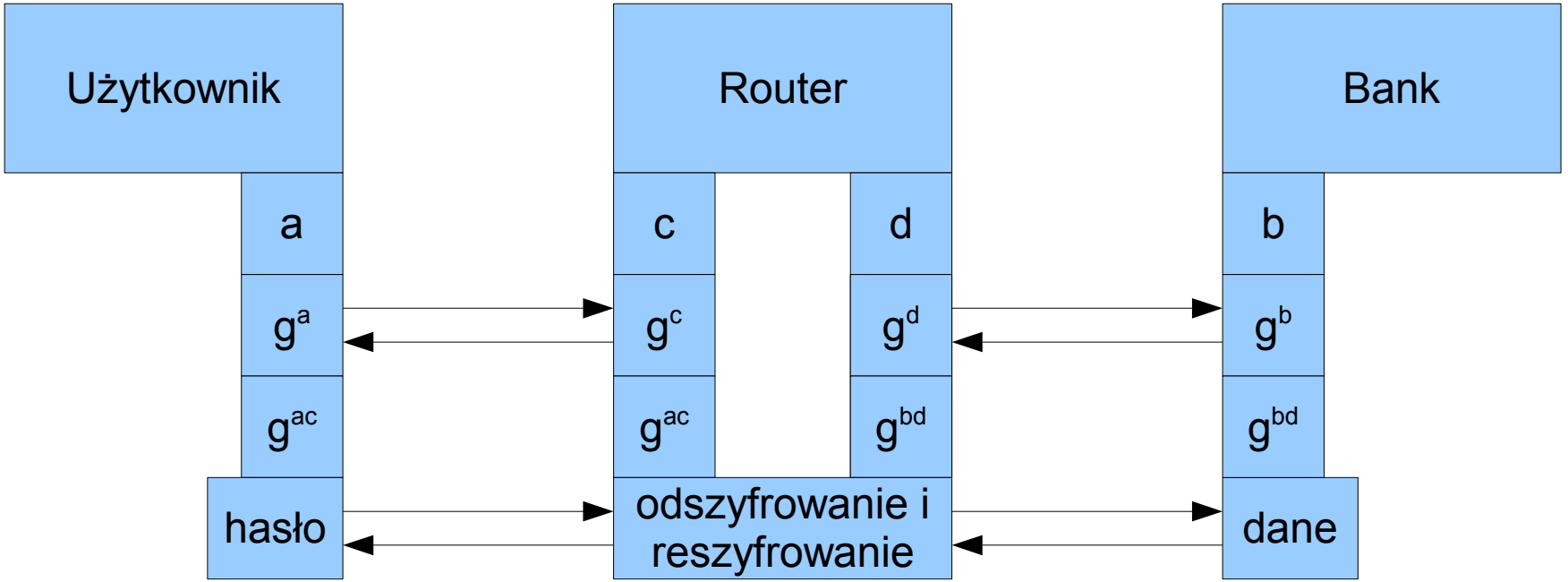
Hacking na żywo

- Chcemy zdobyć hasło użytkownika do banku
- Niestety połącznie jest szyfrowane
  - Klucz ustalany przy połączeniu
  - Samo podsłuchiwanie nie wystarcza



Hacking na żywo

- Oryginalne liczb możemy podmienić własnymi
- Mamy dostęp do haseł i danych bankowych
- Użytkownik i bank są nieświadomi



Hacking na żywo

- Żeby zapobiec man-in-the-middle banki i inne bezpieczne strony internetowe mają certyfikaty
- Sfałszowanie certyfikatu niemożliwe
- Włamanie na serwery banku trudne
- Włamanie na komputer użytkownika łatwiejsze
- Na szczęście dla atakującego:
  - Przedstawiamy błędny certyfikat
  - Przeglądarka otwiera okienko z ostrzeżeniem
  - Użytkownik i tak prawie zawsze naciśnie kontynuuj
  - Użytkownik nie ma certyfikatu więc bank nic nie wie

Hacking na żywo

- Bezpieczeństwa nie można zignorować
- Co robić jeśli odkryje się dziurę ?
- Jest tylko jedna słuszna procedura

Należy ją natychmiast publicznie ujawnić

Hacking na żywo

- Jeśli odkryło się dziurę, można śmiało założyć, że ci źli również o niej wiedzą i z niej korzystają
- Milczenie to moralna współodpowiedzialność za ataki, którym mogliśmy zapobiec
- Praktyka wykazuje, że informowanie producenta i liczenie na szybką reakcję nie działa
  - Poprawki są wydawane po miesiącach lub wcale
  - Użytkownicy nie wiedzą o problemie
    - i dalej używają Internet Explorera

Hacking na żywo

- Razem ze szczegółami na temat dziury publikowany jest zwykle exploit
- Exploity są używane do testowania błędu i są pomocne przy jego łataniu
- Exploity mogą zostać użyte do ataku
- Bazy danych zawierają exploity na stare wersję prawie wszystkich popularnych programów

Hacking na żywo

- SQL – język komunikacji z bazami danych
- SELECT * FROM users WHERE login='admin' AND password='hb376trdvb'
- Zapytania program zwykle generuje przez wklejanie do szablonu
- SELECT * FROM users WHERE login='**?**' AND password='**?**'

Hacking na żywo

- SELECT * FROM users WHERE login='**?**' AND password='**?**'
- Gdy zalogujemy się z:
  - login: admin
  - hasło: ' OR 'x'='x
- SELECT * FROM users WHERE login='**admin**' AND password='**" OR 'x'='x**'
- Czasem można w ten sposób skasować nawet bazę danych

Hacking na żywo

- SQL można traktować jako “zwykły tekst”
  - przy takim traktowaniu łatwo o dziury
- Problem – znaki specjalne takie jak '
- Nie działa 1 – ręczne escape'owanie
- Nie działa 2 – ręczne weryfikowanie
- Nie działa 3 – ręczne filtrowanie
- Nie działa 4 – automatyczna modyfikacja danych pochodzących od użytkownika

Hacking na żywo

- Rozwiązanie – SQL to nie jest zwykły tekst
- Funkcja budująca zapytania powinna dostać szablon i listę elementów do wstawienia
  - automatycznie zajmie się znakami specjalnymi
  - $query = $sql->prepare("SELECT * FROM users WHERE login=? AND password=?");
  - $query->execute($user_name, $password);
- Można też zapomnieć o SQL
  - Use.find_by_login_and_password(login, passwd)
- PHP samo w sobie jest problemem

Hacking na żywo

- Największą grupą dziur są przepełnienia bufora
- Programy używają wielu tablic
  - Tekst to tablica znaków (1 bajt na znak)
- Tablicy dostaje fragment pamięci, w której przechowuje dane
  - 60200 do 60299 – 100-elementowa tablica bajtów
  - 60300 do 60303 – inny obiekt
  - i-ty element ma adres 60200+i
    - elementy liczone oczywiście od zera

Hacking na żywo

- W większości języków programowania dostęp do elementów tablicy jest weryfikowany
  - Próba dostępu do elementu 122 tablicy 100-elementowej zgłosi wyjątek
- C nie zawraca sobie takimi problemami głowy
  - Element 0 to 60200
  - Element 74 to 60274
  - Element 102 to 60302 – tam już jest inny obiekt !
- Z samego tylko powodu przepełnień bufora żaden program w C nie jest bezpieczny