Session Management

Sławomir Rozbicki

slawek@rozbicki.eu

28-07-2011

OWASP TOP 10

A1: Injection

A2: Cross-Site Scripting (XSS)

A3: Broken Authentication and Session Management

A4: Insecure Direct Object References

A5: Cross-Site Request Forgery (CSRF)

...

Uwierzytelnienie i ustanowienie sesji

Zapytanie

```
POST /login.php HTTP/1.1
(...)

username=test&password=test&submit=Submit
```

Odpowiedź

```
HTTP/1.1 302 Found
Cache-Control: private, no-cache, no-store, must-revalidate
(...)
Set-Cookie: user=10001434038; expires=Sat, 06-Aug-2011 12:15:24 GMT;
    path=/; domain=.example.com; secure
Set-Cookie: csm=1; expires=Sat, 06-Aug-2011 12:15:24 GMT; path=/;
    domain=.example.com; httponly
Set-Cookie: sessid=ekmTICYN2aizF26zXqy; expires=Sat, 06-Jul-2013 12:15:24
    GMT; path=/; domain=.example.com; httponly
```

Uwierzytelnienie i ustanowienie sesji

Zapytanie

```
POST /login.php HTTP/1.1
(...)

username=test&password=test&captcha=fS2XG29x&submit=Submit
```

- Implementacja captcha zapobiega atakom na mechanizm uwierzytelnienia
- Identyfikator sesji podatny jest na ten sam rodzaj ataków

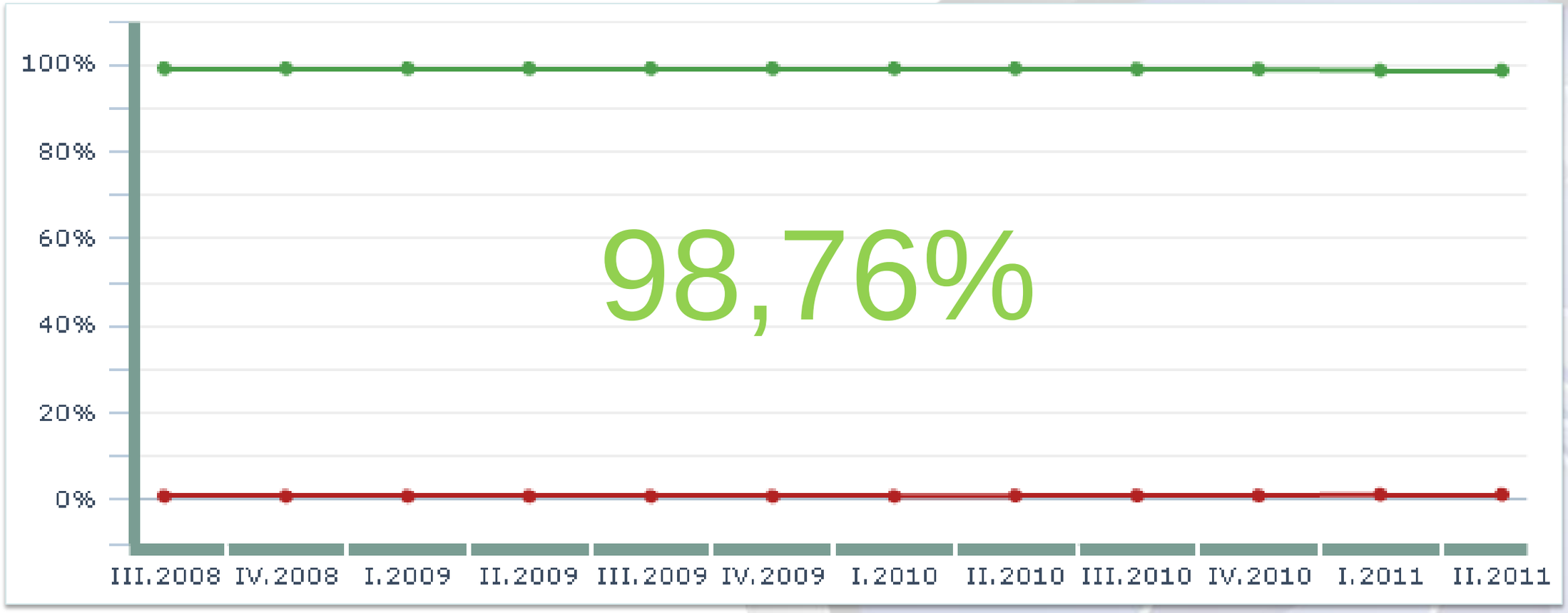
Czy taki atak zostanie wykryty?

Alternatywa? Identyfikator w URL

```
GET /index.php?PHPSESSID=f6a85f35bf457b066706a8153f844b92&action=post
    (...)
```

- nie wymaga obsługi cookies
- pozostawia ślady w logach oraz w „historii”
- identyfikator widoczny w przeglądarce
- łatwy do przypadkowego udostępnienia
- łatwość manipulacji wartością (session fixation)

Poziom akceptacji cookies



źródło: ranking.pl

Cechy bezpiecznego identyfikatora

- losowość
- złożoność (duża przestrzeń znaków)
- czas życia identyfikatora:
  - 5, 10, 20 minut – w zależności od wagi informacji dostępnych użytkownikowi
  - określona ilość żądań
- tokeny uzależnione od adresu IP
- bezpieczna (odseparowana) przestrzeń na serwerze, w której składowane są informacje o tokenach

Cechy bezpiecznego identyfikatora

- walidacja tokenów wysyłanych przez klientów
- zmiana identyfikatora w przypadku żądania, które zmienia poziom uprawnień zalogowanego użytkownika
- rejestracja zdarzeń nietypowych (prób ataku)
- szyfrowany kanał komunikacji
- identyfikator nie jest powiązany z informacjami o uzytkowniku (login, hasło, id, numer telefonu)
- funkcja „remember me” – stosowana z głową

Wniosek?

- Wykorzystanie funkcji zarządzania sesją wbudowanych w framework
- Hardening konfiguracji frameworku w zależności od wymaganego poziomu bezpieczeństwa

Cookie – atrybuty i flagi

- Expires – data wygaśnięcia
- Max-Age – okres ważności
- Domain - nazwa hosta, adres ip
- Secure – flaga lub jej brak
- HttpOnly – flaga lub jej brak
- Path(!) – ścieżka

Ataki na sesje

Wrażliwe informacje zawarte w ciastkach mogą zostać:

- odgadnięte
- przechwycone
- nadpisane

Rezultat: atakujący wchodzi w ich posiadanie.

Odgadywanie tokenów

Ryzyko odgadnięcia rośnie wraz z ilością zalogowanych użytkowników. Ewentualny wzrost popularności aplikacji musi być przewidziany już na etapie projektowania.

Identyfikacja niepoprawnych implementacji:

- Wykorzystanie czasu/daty
- Wykorzystanie funkcji rand(), java.util.Random
- Wykorzystanie losowo wygenerowanego niezmiennego łańcucha znaków
- Identyfikatory o długości 32 bitów lub mniej

Odgadywanie tokenów

Poprawne implementacje:

- Funkcje: java.security.SecureRandom, System.Security.Cryptography.RNGCryptoServiceProvider (.NET)
- /dev/urandom
- OpenSSL RAND

Przechwytywanie tokenów

- Sniffing – ataki sieciowe

Współdzielone medium, arp spoofing w sieciach przełączanych, dns poisoning, dhcp spoofing, protokoły routingu dynamicznego, SSLSTRIP

- Cross-site scripting

document.cookie w źródle HTML (np. IMG SRC)

Nadpisywanie tokenów

Session fixation – atak możliwy na skutek wad projektowych aplikacji. Takich jak:

- Brak zmiany wartości tokenu po zalogowaniu lub po zwiększeniu uprawnień użytkownika
- Współdzielenie identyfikatora w sesji HTTP i HTTPS
- Honorowanie identyfikatora wygenerowanego przez klienta

Ciekawe przykłady

aplikacja.domena.pl przechowuje token w ciastku z flagą SECURE, i parametrem *Domain=„.domena.pl”*.

jeżeli w źródle strony pojawi się odniesienie do dowolnej subdomeny **.domena.pl*, na przykład:

```
<img src=„https://oszust.domena.pl/plik.jpg”>
```

to aplikacja otrzyma pliki cookie.

w korporacjach otrzymanie certyfikatu SSL podpisanego przez firmowy CA nie jest trudne.

Ciekawe przykłady

aplikacja.domena.pl ustanawia plik cookie z parametrem domain=„aplikacja.domena.pl”, który zawiera token sesyjny.

Cookie nie jest oznaczony flagą SECURE, jednak serwer aplikacja.domena.pl filtruje wszystkie porty oprócz 443/TCP, na którym nasłuchuje serwer HTTPS.

Użytkownik zostaje zwabiony na poniższy adres:

```
http://aplikacja.domena.pl:443/plik.jpg
```

Rezultat: dane sesyjne przesyłane są w formie nieszyfrowanej.

Q&A

Kto nie ma żadnych pytań?

Więcej informacji

- https://www.owasp.org/index.php/Session_Management_Cheat_Sheet#Secure_Attribute
- https://www.isecpartners.com/files/web-session-management.pdf
- http://www.sans.org/reading_room/whitepapers/webservers/secure-session-management-preventing-security-voids-web-applications_1594