



security

programistyczne
Wojciech Makurót



the grugq

@thegrugq



Follow

New rule: if you are hacked via OWASP Top 10, you're not allowed to call it "advanced" or "sophisticated."

1:58 PM - 27 Oct 2015



1,407



1,119

Zagrożenia + Dziury =
Ataki

The Open Web Application Security Project

- Non-profit popularyzujący wiedzę o bezpieczeństwie
- Publikuje listę 10 najbardziej **krytycznych** dziur

Co decyduje o miejscu w rankingu?

- Powszechność (*prevalence*)
- Wykrywalność (*detectability*)
- Możliwość wykorzystania (*exploitability*)
- Siła rażenia (*impact*)

Web Application Security Consortium

- Non-profit popularyzujący wiedzę o bezpieczeństwie
- Baza danych upubliczonych incydentów

OWASP Top10 - repeta

- A1-Injection
- A2-Broken Authentication and Session Management
- A3-Cross-Site Scripting (XSS)
- A4-Broken Access Control
- A5-Security Misconfiguration
- A6-Sensitive Data Exposure
- A7-Insufficient Attack Protection
- A8-Cross-Site Request Forgery (CSRF)
- A9-Using Components with Known Vulnerabilities
- A10-Underprotected APIs

OWASP Top10 - repeta

- **A1-Injection**
- **A2-Broken Authentication and Session Management**
- **A3-Cross-Site Scripting (XSS)**
- A4-Broken Access Control
- A5-Security Misconfiguration
- **A6-Sensitive Data Exposure**
- A7-Insufficient Attack Protection
- **A8-Cross-Site Request Forgery (CSRF)**
- A9-Using Components with Known Vulnerabilities
- A10-Underprotected APIs

materialy/security

WebGoat

Jak to działa?

- Dane wejściowe przesyłane **bezpośrednio** do interpretera
- Można “oszukać” interpreter i spowodować, że serwer **wykona kod** dostarczony przez atakującego
- SQL, NoSQL, LDAP, XPath... nawet kod server-side

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Średnia	Średnia	Łatwe	Duża

Wykrywanie

- Inspekcja kodu
- Wszystkie dane wejściowe
 - Zapytania
 - Interpretery
- Skanery i fuzzery

Przeciwdziałanie

- Bindowanie parametrów zapytań
- *Escapowanie* i filtrowanie danych wejściowych
- Niskie uprawnienia

DEMO

Krajowe Biuro Wyborcze Giełda Papierów Wartościowych

Jak to działa?

- Brak ochrony loginów i haseł
- Brak ochrony ID sesji
- Błędy w obsłudze ID sesji
 - Przykład: wiecznie żywe sesje

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Duża	Średnia	Średnia	Duża

Wykrywanie

- Loginy, hasła, identyfikatory sesji
 - URL
 - Formularze
 - Ciasteczka
- Przypomiananie i reset hasła

Przeciwdziałanie

- Ujednolicony system uwierzytelniania
 - Prostota użycia
- Resetowanie haseł
 - Nigdy e-mail!

DEMO

Jak to działa?

- Dane wejściowe renderowane **bezpośrednio** w przeglądarce
- Atakujący może spowodować wykonanie kodu w przeglądarce **innego użytkownika**
- Różne rodzaje: *stored, reflected, DOM-based*

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Bardzo duża	Łatwa	Średnia	Średnia

Wykrywanie

- Inspekcje kodu
 - Łatwe!
- Wstrzykujemy kod w pola wejściowe
 - HTML i JavaScript
- Skanery

Przeciwdziałanie

- *Escapowanie* danych wejściowych
- Filtrowanie danych wejściowych
- Ciasteczka HttpOnly
- Content Security Policy

DEMO

Jak to działa?

- **Bezpośredni dostęp do zabezpieczonego zasobu**
- Atakujący może manipulować identyfikatorem zasobu
- Każde odniesienie do zasobu musi weryfikować prawa użytkownika

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Średnia	Łatwa	Łatwe	Średnia

Wykrywanie

- Inspekcje kodu
 - Odniesienia za pomocą klucza
- Brak wsparcia narzędzi automatycznych

Przeciwdziałanie

- Eliminacja dostępu bezpośrednio za pomocą klucza
- Weryfikacja dostępu

Jak to działa?

- Konfiguracja i wersja każdego elementu stosu powinna być aktualna
- System operacyjny, serwer WWW, baza danych, interpreter, biblioteki, kod aplikacji
- Zbędne oprogramowanie, usługi, otwarte porty

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Średnia	Łatwa	Łatwe	Średnia

Wykrywanie

- Skanery!
- Proces

Przeciwdziałanie

- Hardening
- Regularne patchowanie
- Audyty
- Automatyzacja

GeekedIn

Jak to działa?

- E-maile, loginy, hasła, karty kredytowe, dane osobowe
- Dane przesyłane otwartym tekstem
- Dane przechowywane w postaci źródłowej w bazie danych
 - Backupy!

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Mała	Średnia	Mała	Duża

Wykrywanie

- Downgrade
szyfrowanego
połączenia
 - Ataki na TLS
- “Łamacze haseł”

Przeciwdziałanie

- Brak danych!
- Szyfrowanie
- Hashowanie haseł
- SSL / TLS
 - HSTS + HPKP

Brak kontroli dostępu na poziomie funkcji

Jak to działa?

- Ukrywanie zastrzeżonych funkcji **jedynie** na poziomie interfejsu użytkownika
- Dynamiczny wybór akcji przez użytkownika
 - Parametr w URLu lub formularzu
- Single Page Applications i REST API

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Średnia	Średnia	Łatwe	Średnia

Wykrywanie

- Nagrywamy
 - Użytkownik uprzywilejowany
- Odtwarzamy
 - Użytkownik o niższych uprawnieniach

Przeciwdziałanie

- Kontrola dostępu w middleware
- Centralne zarządzanie i audyt
- Defense in depth

Jak to działa?

- Żądania HTTP inicjowane przez prawdziwego użytkownika i atakującego są z reguły **nieroróżnialne**
- Atakujący może **oszukać** użytkownika i zmusić go do wysłania żądania HTTP przygotowanego przez **atakującego** korzystając z **uwierzytelnionej sesji użytkownika**

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Średnia	Łatwa	Średnia	Średnia

Wykrywanie

- Linki i formularze modyfikujące dane
- Wsparcie narzędzi

Przeciwdziałanie

- Tokeny
- Podwójne uwierzytelnienie
- Chronimy przed XSS

DEMO

YouTube

Jak to działa?

- Stos typowej aplikacji Webowej zawiera wiele komponentów
- **Każdy komponent zawiera dziury**
 - Łatwe do automatycznego skanowania
- Skutki aż do **przejęcia kontroli nad serwerem**

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Duża	Trudna	Średnia	Średnia

Wykrywanie

- Zarządzanie konfiguracją
- Skanery
- Proces

Przeciwdziałanie

- Katalog komponentów
 - Wersje
- Monitoring dziur
- Patchowanie
 - Automatyzacja

Zabezpieczenie serwera

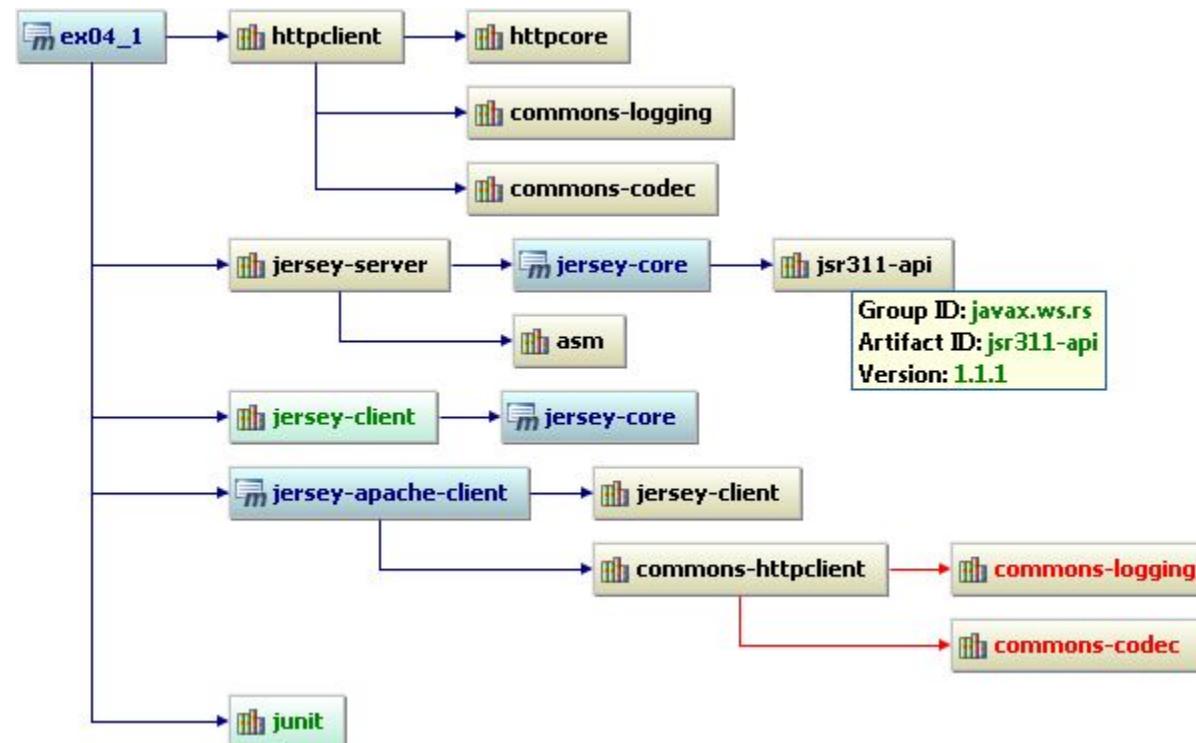
- Przykładowe błędy:
 - dziura w PHP pozwalająca na wykonanie skryptu po stronie serwera
 - dziura w OS pozwalająca wykonać polecenie po stronie serwera - tzw. „shellshock”
 - dziura w SSL pozwalająca podglądać szyfrowany ruch „heartbleed”
[\(<https://www.youtube.com/watch?v=bhJmVBJ-F-4>\)](https://www.youtube.com/watch?v=bhJmVBJ-F-4)

Zabezpieczenie serwera

- wersje używanych bibliotek
 - mvn dependency:tree
 - mvn dependency:tree -Dverbose
 - mvn dependency:tree -Dverbose -Dincludes=commons-collections

Zabezpieczenie serwera

- wersje używanych bibliotek



Jak to działa?

- Niektóre aplikacje umożliwiają przekierowanie użytkownika w inne miejsce aplikacji (np. po uwierzytelnieniu)
- Często adres przekierowania jest przekazywany w URLu
- Atakujący wykorzystuje zaufanie użytkownika do *phishingu*

Powszechność	Wykrywalność	Możliwość wykorzystania	Siła rażenia
Mała	Łatwa	Średnia	Średnia

Wykrywanie

- Inspekcja kodu
- Crawlery
 - Szukamy odpowiedzi HTTP 30x

Przeciwdziałanie

- Brak przekierowań
- Przekierowania bez parametrów
- Mapowanie parametrów

OWASP Top10 - repeta

- **A1-Injection**
- **A2-Broken Authentication and Session Management**
- **A3-Cross-Site Scripting (XSS)**
- A4-Broken Access Control
- A5-Security Misconfiguration
- **A6-Sensitive Data Exposure**
- A7-Insufficient Attack Protection
- **A8-Cross-Site Request Forgery (CSRF)**
- A9-Using Components with Known Vulnerabilities
- A10-Underprotected APIs

- Troy Hunt
 - Blog: <https://www.troyhunt.com/>
 - Pluralsight: <https://www.pluralsight.com/authors/troy-hunt>
- Scott Helme
<https://scotthelme.co.uk/>
- The Basics of Web Application Security
<http://martinfowler.com/articles/web-security-basics.html>
- OWASP Testing Project
https://www.owasp.org/index.php/Category:OWASP_Testing_Project
- OWASP Code Review Project
https://www.owasp.org/index.php/Category:OWASP_Code_Review_Project
- sqlmap: Automatic SQL injection and database takeover tool
<http://sqlmap.org/>

- OWASP Application Security Verification Standard
https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project
- OWASP Dependency-Check
 - https://www.owasp.org/index.php/OWASP_Dependency_Check
- OWASP AppSensor
https://www.owasp.org/index.php/OWASP_AppSensor_Project
- XSS game (Google)
<https://xss-game.appspot.com/>
- NodeGoat
<https://github.com/OWASP/NodeGoat>
- OWASP ZAP
https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project

- SQL Injection Attacks by Example
<http://www.unixwiz.net/techtips/sql-injection.html>
- Stored procedures and ORMs won't save you from SQL injection
<https://www.troyhunt.com/stored-procedures-and-orms-wont-save>
- Unraveling some of the Mysteries around DOM-based XSS
https://www.owasp.org/images/c/c5/Unraveling_some_Mysteries_around_DOM-based_XSS.pdf
- Cross-site scripting
<https://www.google.com/about/appsecurity/learning/xss/>
- Open-source software security
https://en.wikipedia.org/wiki/Open-source_software_security
- Open redirect URLs: Is your site being abused?
<http://bit.ly/25e1GKc>