security

konfiguracyjne

Wòjcech Makùrôt

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

OWASP Top10 - repeta

- A2-Broken Authentication and Session Management
- A4-Broken Access Control
- A5-Security Misconfiguration
- A9-Using Components with Known Vulnerabilities
- DoS/DDoS
- monitoring
- SSO

Rodzaje ataków

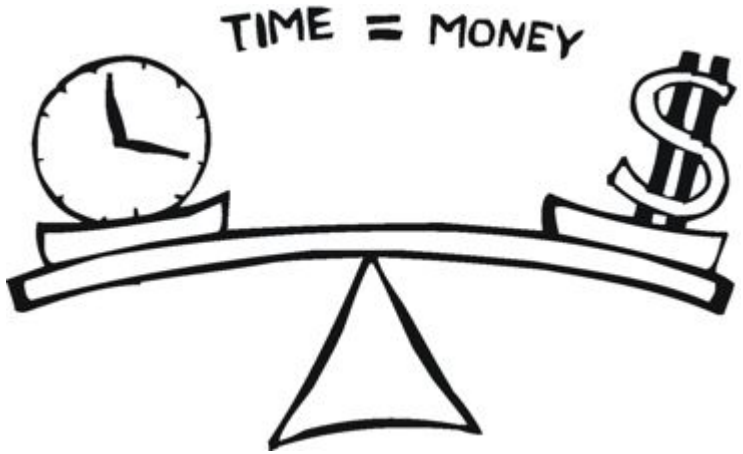
- cel ataku
- warstwa ISO/OSI, której dotyczy
- wewnętrzy vs zewnętrzny
- z dostępem fizycznym i bez
- ...

Rodzaje ataków

- jakie znacie ataki zw. z WWW?

Skutki ataków

- Efektem ataków jest zawsze strata finansowa lub czasowa



Dostęp do serwerów produkcyjnych

- Dostęp do środowiska produkcyjnego, powinien być jak najlepiej zabezpieczony i jak najtrudniejszy do uzyskania dla osób nieupoważnionych.

Dostęp do serwerów produkcyjnych

- Można to osiągnąć poprzez:
  - Odpowiednie polityki dostępu
    - dostęp ma tylko te osoby które powinny mieć
    - dostęp jest tylko do wybranych serwerów
    - dostęp jest tylko do wybranych zasobów na serwerze (odpowiednie uprawnienia)
    - istnieją do tego gotowe narzędzia - np. freeipa
  - Dostęp do serwera produkcyjnego jedynie poprzez VPN

Dostęp do serwerów produkcyjnych

- Dostęp do serwera produkcyjnego jedynie poprzez „entrance”
  - serwer wystawiony na niestandardowym porcie SSH
  - logiki logowania i banowania nieudanych prób - np. iptables
  - logowanie przy użyciu certyfikatów zamiast haseł
  - dostęp tylko z określonych IP (white list)

Zabezpieczenie serwera

- Korzystamy zawsze z najnowszych wersji pakietów (obserwujemy strony i listy mailingowe w kontekście nowych błędów).
- Dotyczy to zarówno serwera web, jak i wszystkich dodatkowych modułów oraz samego OS.

Zabezpieczenie serwera

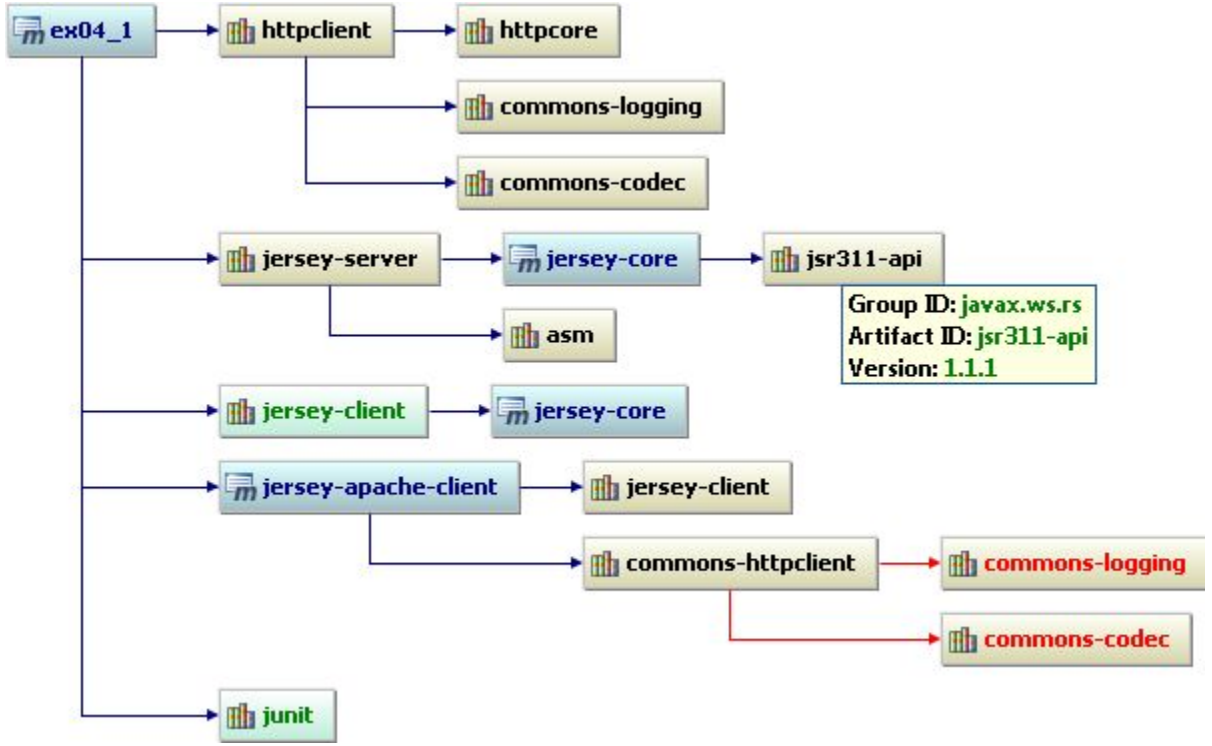
- Przykładowe błędy:
  - dziura w PHP pozwalająca na wykonanie skryptu po stronie serwera
  - dziura w OS pozwalająca wykonać polecenie po stronie serwera - tzw. „shellshock”
  - dziura w SSL pozwalająca podglądać szyfrowany ruch „heartbleed” (https://www.youtube.com/watch?v=bhJmVBJ-F-4)

Zabezpieczenie serwera

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

Zabezpieczenie serwera

- wersje używanych bibliotek



Zabezpieczenie serwera - domyślne porty

- management-http:9990
- management-https:9993
- ajp:8009
- http:8080
- https:8443
- txn-recovery-environment:4712
- txn-status-manager:4713

Zabezpieczenie serwera - domyślne porty

- Wildfly
  - jboss.socket.binding.port-offset

Zabezpieczenie serwera - domyślne porty

- Maven
  - <plugin>

```
<groupId>org.wildfly.plugins</groupId>
<artifactId>wildfly-maven-plugin</artifactId>
<configuration>
  <serverArgs>-Djboss.socket.binding.port-offset=100</serverArgs>
</configuration>
</plugin>
```

Zabezpieczenie serwera - domyślne porty

- Docker
  - -p 5653:8080

banowanie

- udostępnianie tylko wybranego portu
  - iptables -P FORWARD DROP
  - iptables -P INPUT DROP
  - iptables -A INPUT --protocol tcp --destination-port 80 -j ACCEPT

banowanie

- blokowanie adresu IP
  - iptables -A INPUT -s 192.168.0.11 -j DROP
  - iptables -A FORWARD -s 192.168.0.11 -j DROP
  - iptables -A INPUT -d 192.168.0.11 -j DROP
  - iptables -A FORWARD -d 192.168.0.11 -j DROP

http://iptables.pl/

dostęp do serwera

- hasła vs certyfikat
- IP white lists
  - iptables

banowanie

- udostępnianie tylko wybranego portu
  - iptables -P FORWARD DROP
  - iptables -P INPUT DROP
  - iptables -A INPUT --protocol tcp --destination-port 80 -j ACCEPT

banowanie

- IP white lists
  - iptables

Zabezpieczenie serwera

- wyłączenie niewykorzystywanych modułów serwera
  - standalone.xml/domain.xml

Zabezpieczenie serwera

- usuwanie niewykorzystywanych zależności maven
  - standalone.xml/domain.xml
  - mvn dependency:analyze
  - IDE

Zabezpieczenie serwera

- wyłączenie indeksacji zawartości serwera

Zabezpieczenie serwera

- Ukrywanie tożsamości
  - HTTP Server
  - HTTP X-Powered-By

Zabezpieczenie serwera

- Ukrywanie tożsamości
  - welcome content

Zabezpieczenie serwera

- Ukrywanie tożsamości
  - JSESSIONID
  - favicon

Zabezpieczenie serwera

- Ukrywanie .jsp
  - <servlet>
    <servlet-name>index</servlet-name>
    <jsp-file>/index.jsp</jsp-file>
    </servlet>
    <servlet-mapping>
    <servlet-name>index</servlet-name>
    <url-pattern>/index</url-pattern>
    </servlet-mapping>

Zabezpieczenie serwera

- Uruchamianie serwera z odpowiednimi uprawnieniami
  - # groupadd -r wildfly
  - # useradd wildfly -r -g wildfly -d /var/www -s /sbin/nologin

Zabezpieczenie serwera

- Ograniczanie podstrony tylko do wybranych IP
  - <filter-ref name="ipAccess"/>
  - <expression-filter name="ipAccess" expression="path-prefix[/admin] -> ip-access-control[default-allow=false, acl={'127.0.0.1 allow'}]"/>

Zabezpieczenie serwera

- Ograniczenie metod HTTP tylko do tych, z których korzystamy
  - <context-param>
    <param-name>resteasy.role.based.security</param-name>
    <param-value>true</param-value>
    </context-param>

https://github.com/Holdo/wildfly-10-web-app/blob/master/ui/src/main/webapp/WEB-INF/web.xml

Zabezpieczenie serwera

- Ograniczenie metod HTTP tylko do tych, z których korzystamy
  - <login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
    <form-login-page>/login.html</form-login-page>
    <form-error-page>/fail_login.html</form-error-page>
    </form-login-config>
    </login-config>
    <security-role>
    <role-name>admin</role-name>
    </security-role>

Zabezpieczenie serwera

- Ograniczenie metod HTTP tylko do tych, z których korzystamy
  - <security-constraint>
    <web-resource-collection>
    <url-pattern>/rest/demo/*</url-pattern>
    <http-method>DELETE</http-method>
    </web-resource-collection>
    <auth-constraint>
    <role-name>admin</role-name>
    </auth-constraint>
    </security-constraint>

Monitoring

- Obserwacja logów
  - foo.example.com - - [12/Jul/2002:01:59:13 +0200] "GET /.htpasswd HTTP/1.1"

DoS/DDoS

- timeout
  - <http-listener read-timeout="120000"/>
- max parameters
- max-buffered-request-size
- max-parameters
- max-connections
- max-header-size
- max-headers
- max-processing-time

Autentykacja EJB

```
@Stateless
@RolesAllowed("friend")
public class EndpointEJB implements EndpointInterface {
  /* (...) */
}
```

https://docs.jboss.org/author/display/WFLY10/Authentication

HTTPS

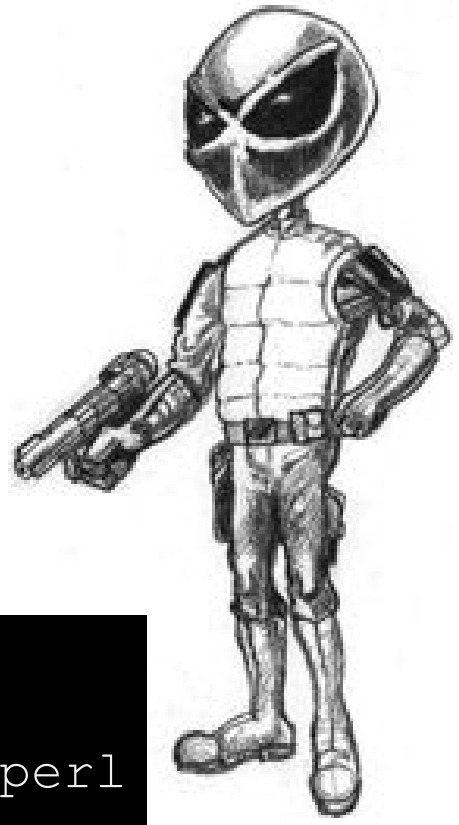
```
@Stateless
@RolesAllowed("friend")
public class EndpointEJB implements EndpointInterface {
  /* (...) */
}
```

https://docs.jboss.org/author/display/WFLY10/Authentication

Zabezpieczenie serwera

https://cirt.net/Nikto2

https://github.com/sullo/nikto



```
git clone https://github.com/sullo/nikto.git
apt-get install -y libnet-ssleay-perl libcrypt-ssleayperl
cd nikto/program
./nikto.pl -host <HOST>
```