

Сигурност на уеб приложенията

Потенциални атаки и защита от тях



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Разработка на софтуер

Съдържание

1. Основи на сигурността
2. Най-често срещаните атаки:
 1. SQL Injection
 2. XSS
 3. CSRF
 4. Parameter Tampering





Основи на Сигурността

Концепциите за уеб сигурност

Основи на Сигурността

- Уеб Сигурността включва мерки за подобряване на сигурността на приложението
 - Често се прави чрез поправяне и предотвратяване на уязвимости в сигурността
- Уязвимостите се развиват непрекъснато, но най-вече - до голяма степен последователни

"Едно нещо се счита за сигурно, когато разходите за пробив струват повече от стойността, получена по този начин"

Основи на Сигурността

- Нарушенията на сигурността често се случват спонтанно
 - Уязвимостта може да бъде напълно непреднамерена
- Нарушенията на сигурността са резултат от злонамерени атаки
 - Тези атаки може да имат много мотиви, които ги подкрепят
 - Предизвикателство, любопитство, вандализиране, кражба
- Нарушенията на сигурността могат да бъдат напълно дискретни
 - Силно опитни нападатели няма да оставят следа
 - Най-вероятно ще разберете, че сте били нападнати доста по-късно

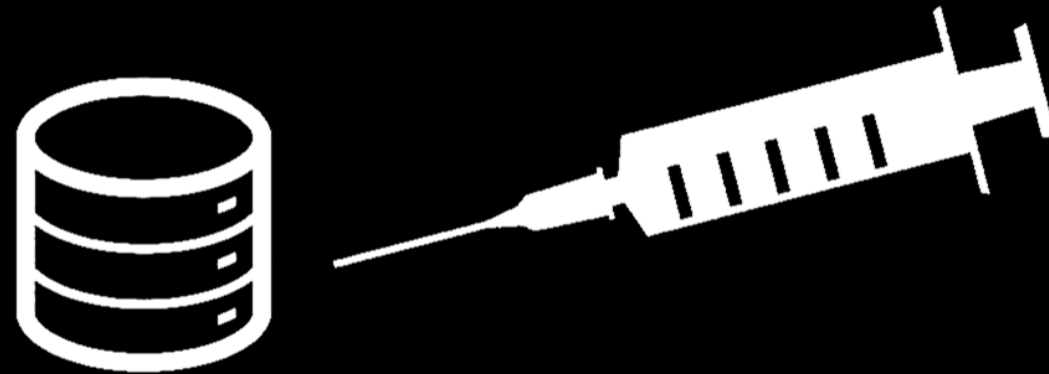
Основи на Сигурността

- Съществува широк спектър от известни видове заплахи и

Категория	Атаки
Валидиране на входа	Преливане на буфер, скриптове, SQL инжекция, канонизация
Подправяне на параметри	Манипулиране на низове за заявки, манипулация на полето на формуляра, манипулиране на бисквитки, манипулация на HTTP хедъри
Управление на сесиите	Открадване на сесия, session replay, man-in-the-middle
Криптография	Лошо генериране на ключове или управление на ключове, слабо или персонализирано криптиране
Чувствителна информация	Достъп до чувствителен код или данни в хранилището, подслушване на мрежата, подправяне на код / данни, администраторска парола
Управление на изключенията	Разкриване на информация, отказ на услугата

Основи на Сигурността

- Някои от най-добрите действия, които един програмист може да предприеме, за подsigуряване на приложението:
 - Максимална простота
 - Подsigуряване на най-слабата връзка
 - Ограничаване на публично достъпните ресурси
 - Неправилно, докато не се докаже правилно
 - Принципът "Weakest Privilege"
 - Сигурност при грешки
 - Осигуряване на постоянна защита



SQL Injection

SQL Injection

- Следните SQL команди се изпълняват:
 - Обичайно търсене(без SQL инжектиране):

```
SELECT * FROM Messages WHERE MessageText LIKE '%Nikolay.IT%'
```

- Търсене с SQL инжектиране(съвпада с всички записи):

```
SELECT * FROM Messages WHERE MessageText LIKE '%%%'
```

```
SELECT * FROM Messages WHERE MessageText LIKE '%' or 1=1 --%'
```

- Команда за вмъкване със SQL инжектиране:

```
SELECT * FROM Messages WHERE MessageText  
LIKE '%'; INSERT INTO Messages(MessageText, MessageDate) VALUES  
( 'Hacked!!!', '1.1.1980' ) --%'
```

SQL Injection

- Оригинално SQL заявка:

```
string sqlQuery = "SELECT * FROM user WHERE name = ' " +  
username + "' AND pass=' " + password + "'";
```

- Задаване на потребителско име на John & парола на ' OR '1'='1

```
string sqlQuery = "SELECT * FROM user WHERE name = 'Admin' AND  
pass=' ' OR '1'='1'";
```

- Резултатът:

- Потребителят с потребителско име – "Admin" ще влезе БЕЗ парола
- Заявката за преминаване ще се превърне в bool израз, който винаги е верен



SQL Injection

Демонстрация на Живо



XSS

Cross Site Scripting (XSS)

XSS

- Cross-site scripting (XSS) е често срещана уязвимост в веб приложенията
- Веб приложенията показват JavaScript код
 - Изпълнява се в браузъра на клиента
 - Хакерите могат да поемат контрол над сесиите, бисквитките, паролите и т.н.
- Как да се предпазим от XSS?
 - Проверете потребителския вход (вградено в ASP.NET Core)
 - Изпълнявайте HTML escaping при показване на текстови данни

XSS

- Cross-site scripting атака:
 - Кражба на бисквитки
 - Кражба на акаунт
 - Промяна на съдържанието
 - Променете потребителските настройки
 - Изтеглете зловреден софтуер
 - Изпращане на CSRF атаки
 - Подказване на парола





XSS

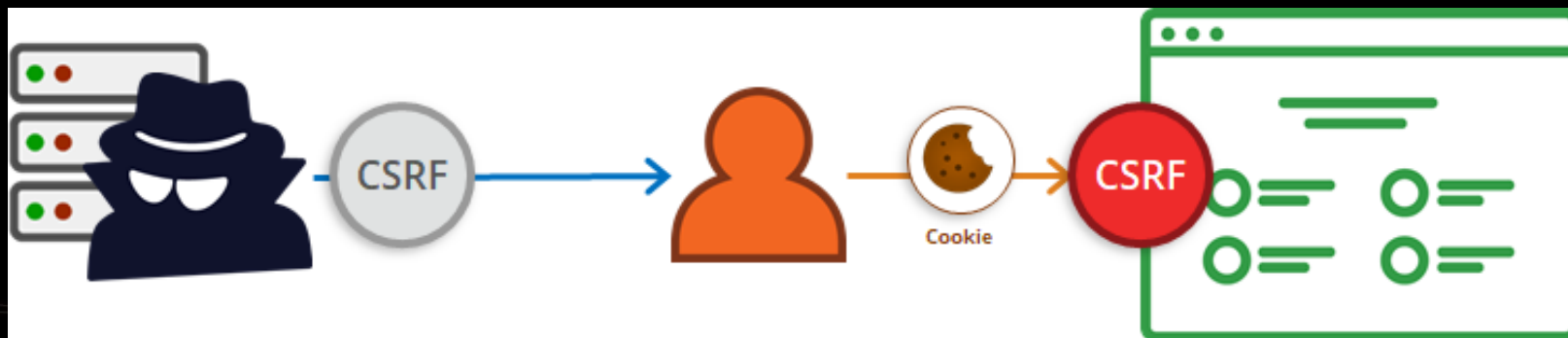
Демонстрация на Живо

CSRF

Cross-Site Request Forgery

Cross-Site Request Forgery

- Това е атака на уеб сигурност над HTTP протокола
 - Позволява изпълнението на неоторизирани команди от името на някой потребител
 - Потребителят има валидни разрешения за изпълнение на заявената команда
 - Нападателят използва тези разрешения злонамерено, непознато за потребителя



Cross-Site Request Forgery

- Процесът не е толкова сложен за разбиране:
 - Потребителят има валидна бисквитка за автентикация до victim.org
 - Съхранява се в браузъра
 - Нападателят моли потребителя да посети <http://evilsite.com>
 - Нападателят взема съхранената бисквитка
 - Злият сайт изпраща HTTP Заявка до victim.org чрез бисквитката
 - victim.org извършва действия от името на потребителя
 - Действията се извършват с данните на потребителя

Cross-Site Request Forgery

- Как Cross-Site Request Forgery изглежда най-често:

CONGRATULATIONS
YOU ARE OUR 42069 WINNER

A1IN A12947198247109247881029398135812
CHANGE!!>!?!?!1/11//

**DONT MISS YOUR
CHANCE**



[CLICK HERE TO COLLECT YOUR
XXX-SUPER-DUPER-MEGA-ULTRA-TURBO-DIEZEL-4X4-XXX-PRIZE](#)

Cross-Site Request Forgery

- Какво Cross-Site Request Forgery всъщност е:

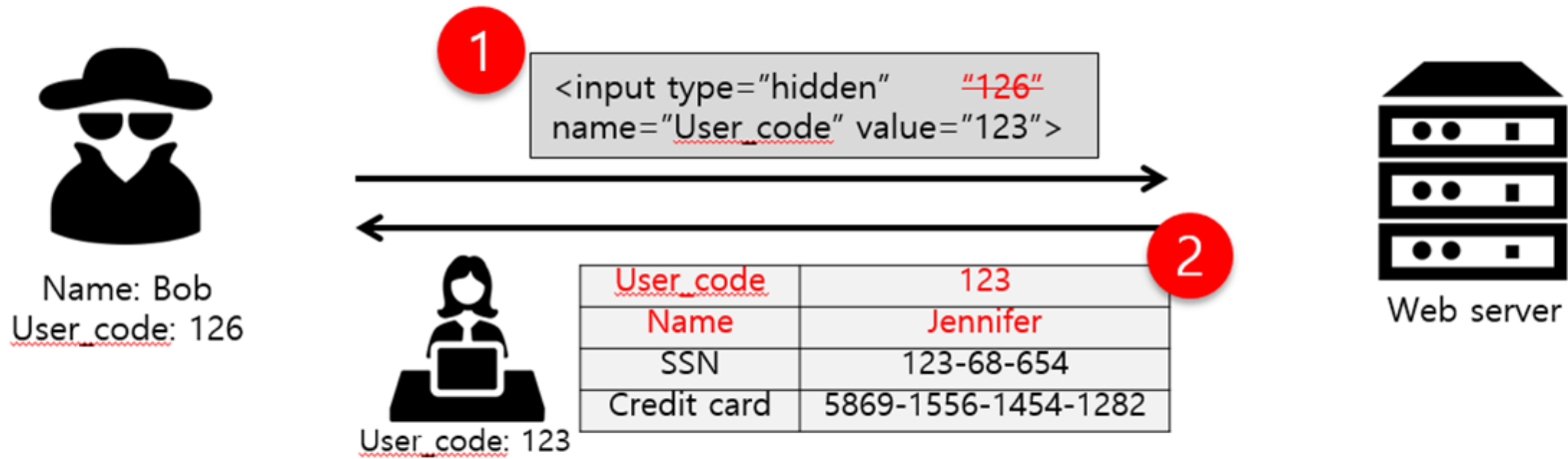
```
<!-- SOME MULTI-COLOR USELESS CLICKBAIT CONTENT -->  
  
<form action="http://good-banking-site.com/api/account" method="post">  
  <input type="hidden" name="Transaction" value="withdraw">  
  <input type="hidden" name="Amount" value="1000000">  
  <input type="submit" value="Click to collect your prize!">  
</form>
```

- Потребителят дори може грешно да кликне бутона
 - Това ще активира атаката
 - Сигурността срещу подобни атаки е необходима
 - Защищава както вашето приложение, така и вашите клиенти



Cross-Site Request Forgery

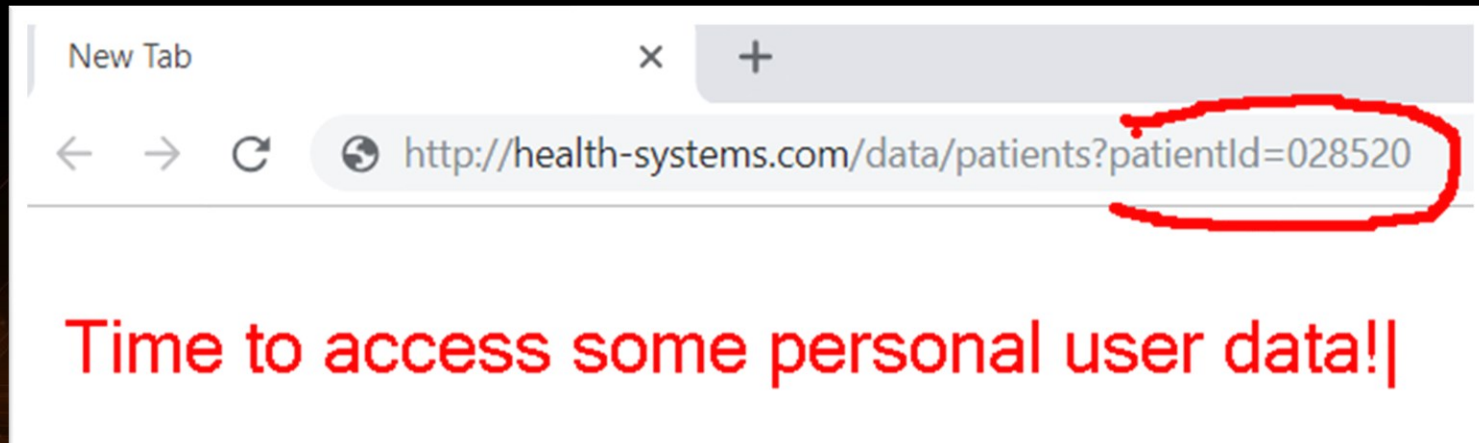
Демонстрация на Живо



Parameter Tampering

Parameter Tampering

- Parameter Tampering е манипулиране на параметри, обменяни между клиент и сървър
 - Променени низове за запитвания, тяло на заявка, бисквитки
 - Пропуснати валидации на данните, инжектирани допълнителни параметри





Parameter Tampering

Демонстрация на Живо

Други заплахи за сигурността

- Семантични URL/HTTP атаки (URL/HTTP манипулация)
 - Винаги проверявайте данните от страна на сървъра
- Man in the Middle (Винаги ползвайте SSL)
- Недостатъчен контрол на достъпа
- Други видове инжектиране на данни (Винаги saniрайте данните)
- DoS и DDoS и Brute Force attacks (CAPTCHA и Firewall)
- Phishing и Social Engineering (Образовайте потребителите си)
- Пропуски в сигурността на други софтуери (Използвайте последните версии)

Сигурност на уеб приложенията



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

