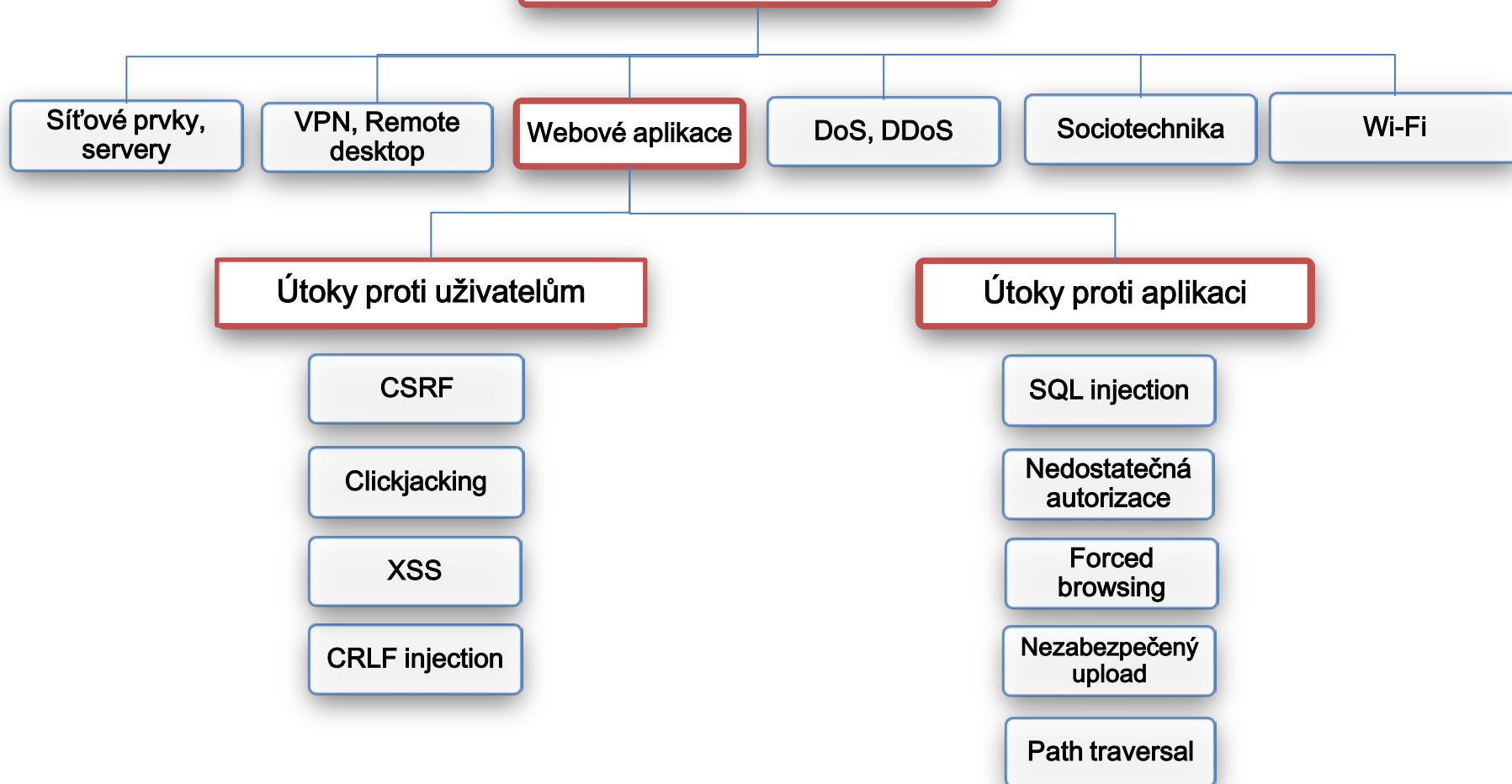


Testování webových aplikací

Seznam.cz

Roman Kümmel

Bezpečnostní hrozby



Cross-Site Request Forgery (CSRF)

- Zneužití důvěry serveru v uživatele
- Nic netušící uživatel odešle serveru GET/POST požadavek
- Skript na serveru rozpozná uživatele podle cookie a požadavek pod identitou uživatele vykoná
- Může jít také o XMLHttpRequesty (AJAX požadavky)

Cross-Site Request Forgery (CSRF)

- Útočník odešle uživateli odkaz pro odeslání nebezpečného požadavku cílové aplikaci

```
http://www.webmailcz/remail?email=utocnik@seznam.cz
```

- Možnost zneužití HTML prvků načítajících externí obsah

```
<img src=http://aplikace.cz/remail?email=utocnik@seznam.cz>
```

Cross-Site Request Forgery (CSRF)

- Útočník vytvoří webovou stránku, která sama odešle nebezpečné požadavky cílové aplikaci ve chvíli, kdy ji (přihlášený) uživatel navštíví
- Pro utajení akce se formulář vloží do skrytého rámu

```
<form name="fake" method="post" action="http://aplikace.cz/remail">  
  <input type="hidden" name="email" value="utocnik@seznam.cz">  
</form>  
  
<script>document.fake.submit();</script>
```

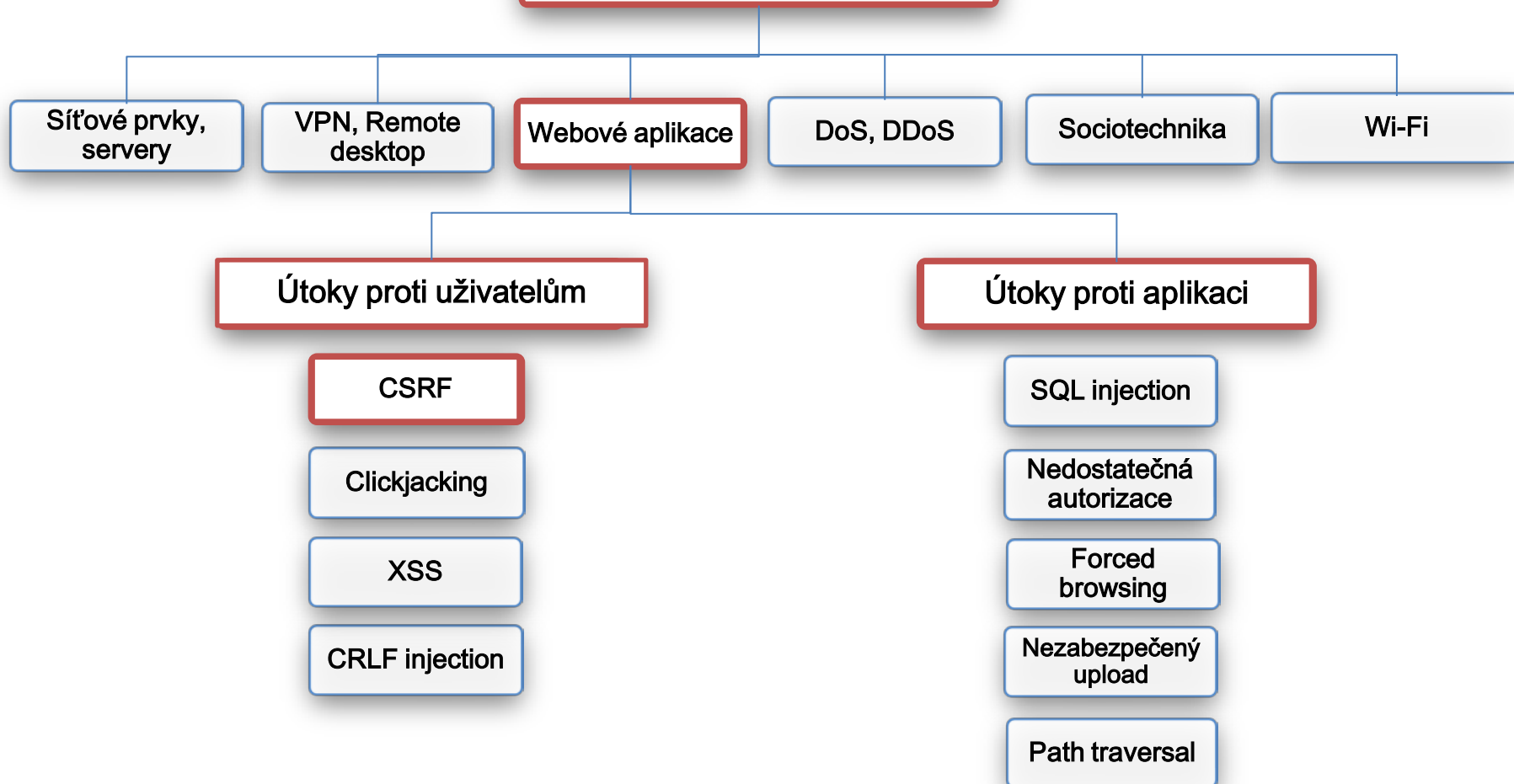
Cross-Site Request Forgery (CSRF)

OBRANA

- Kontrola HTTP hlavičky REFERER
 - Nedoporučuji, možnost odeslání požadavků bez této hlavičky
- Kontrola hlavičky ORIGIN u XMLHttpRequestů
 - Podobná situace jako u hlavičky Referer
- Přidání autorizačního tokenu ke všem požadavkům
 - Útočník nemůže připravit útočný požadavek bez jeho znalosti
 - Ideální je platnost tokenu časově omezit

<http://aplikace.cz/remail?email=utocnik@seznam.cz&ticket=ba5e5aa2f472>

Bezpečnostní hrozby



Clickjacking

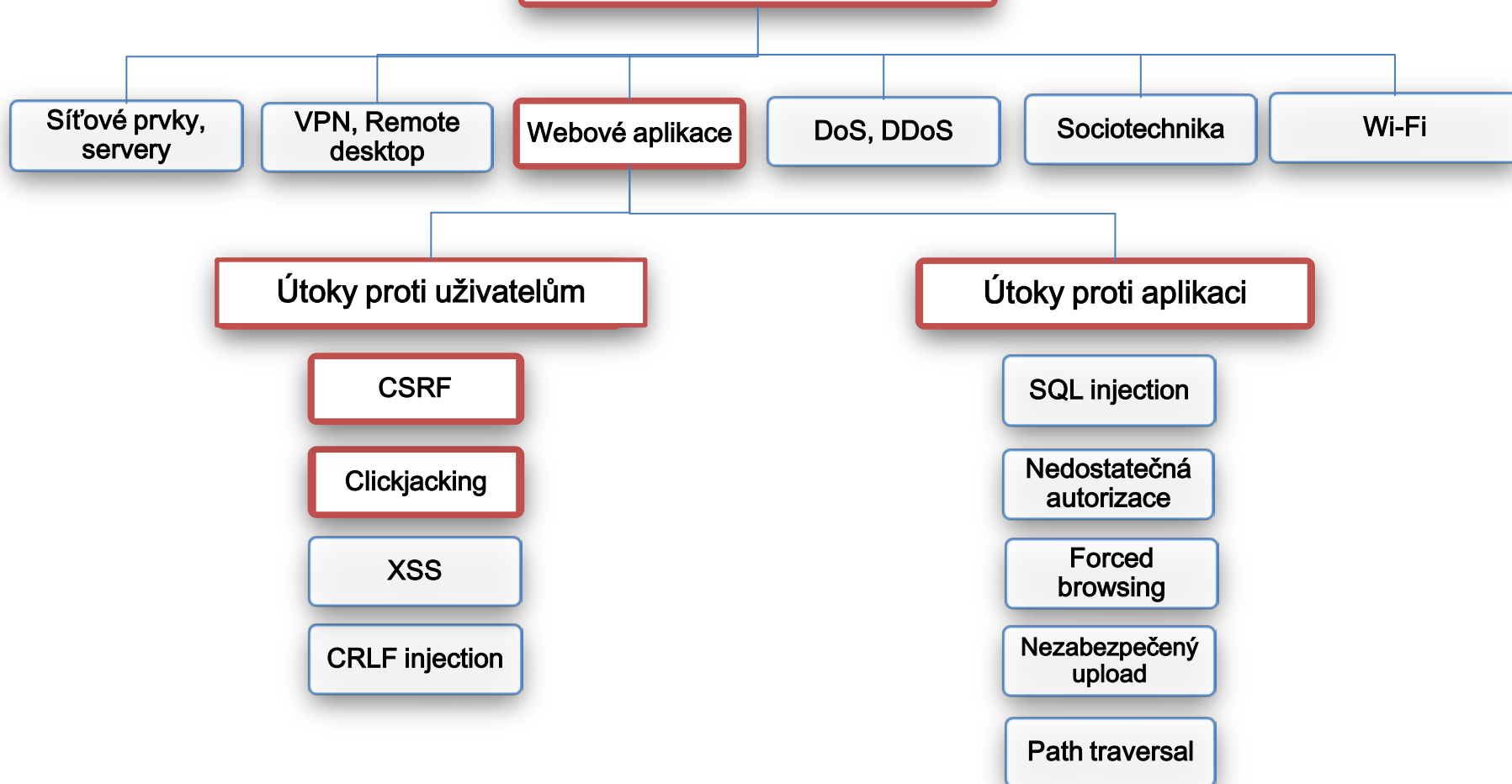
- Je-li nasazena ochrana před útoky CSRF
- Nic netušící uživatel sám klikne na prvek nebo vyplní a odešle formulář, bez toho, aby věděl, co vlastně dělá.
- Útok založen na možnosti načíst web. stránku do rámu
 - Průhlednost rámu
 - Překrytí nechtěných prvků
- Vkládání údajů do aplikace, nebo krádež dat z aplikace.

Clickjacking

OBRANA

- JavaScript **FrameKiller**
 - `if (top.location != self.location)`
 - Nedoporučuji
 - možnost načtení zdrojáku `view-source:`
 - Lze vyřadit zneužitím XSS filtru
- HTTP Response Hlavička **X-Frame Options**
 - DENY
 - SAMEORIGIN
 - ALLOW-FROM
- Content Security Policy

Bezpečnostní hrozby



Cross-Site Scripting (XSS)

- Spuštění JavaScriptu v uživatelově prohlížeči
 - Same Origin Policy brání přístupu k jiným doménám
 - Injekce skriptu do webové aplikace umožňuje přístup ke všem jejím objektům (čtení/zápis)
 - Únos sezení
 - Změna nebo čtení uložených dat (podvržení přihlašovacích formulářů, atd.)
 - XSS proxy
- Typy zranitelnosti XSS
 - Perzistentní
 - Non-Perzistentní
 - DOM based (např. u AJAX aplikací)

Cross-Site Scripting (XSS)

```
<div>Uživatelský příspěvek: <script>alert(1)</script></div>
```

```
<form>  
  <input type="text" value=""><script>alert(1)</script></div>  
</form>
```

```
<input type="text" value="" onclick="alert(1)">
```

```
<script type="text/javascript">  
  DOT.cfg({service: 'firmy', query: ''});alert(1);//'});  
</script>
```

```
<a href="javascript:alert(1)">odkaz</a>
```

Cross-Site Scripting (XSS)

AJAX

```
var json = eval("(" + xhr.responseText + ")");
```

```
{"a":10, "b":20, "c":alert(1)}  
{"a":10, "b":20, "c":"alert(1)"}
```

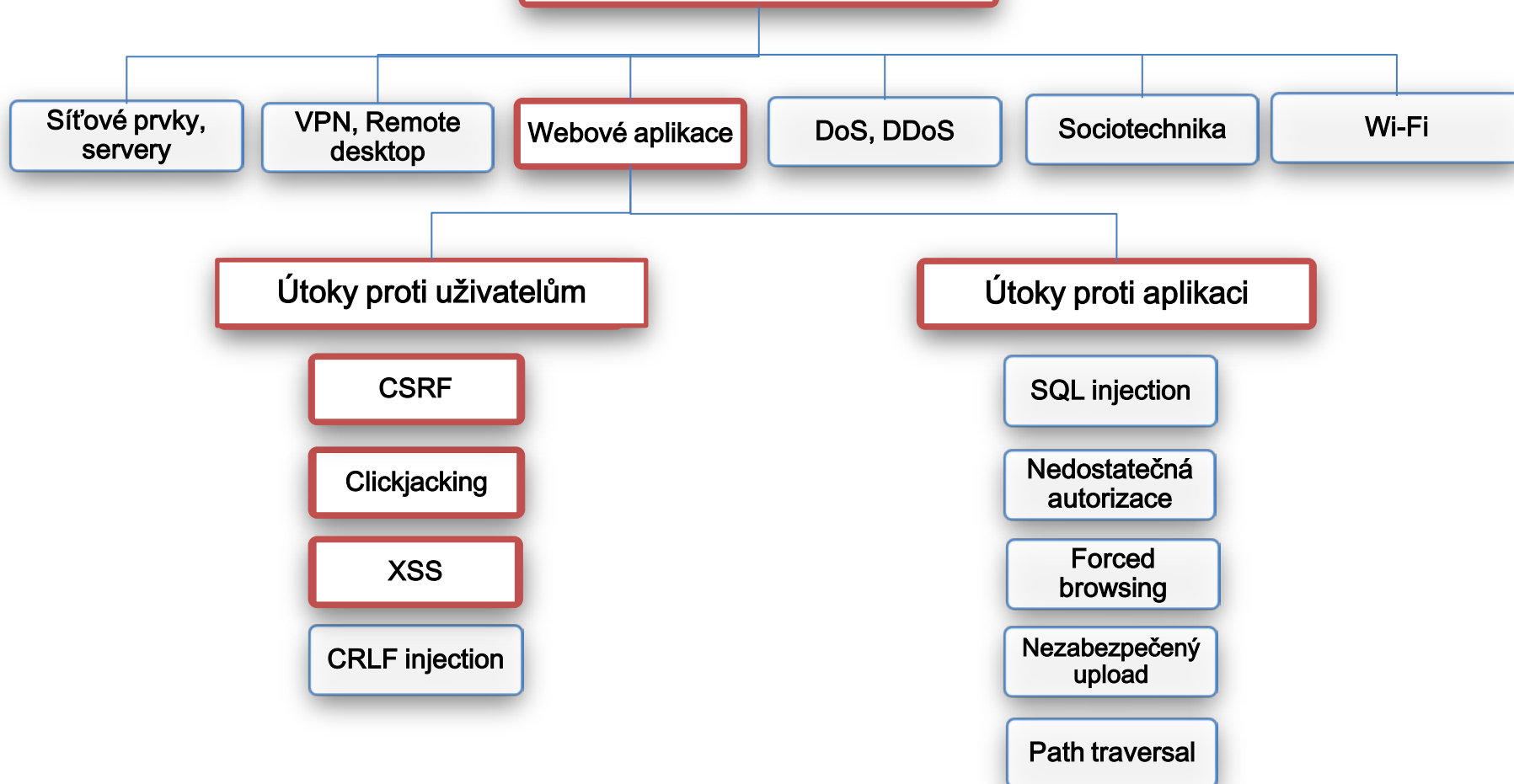
```
" + alert("XSS") + "  
{"a": "" + alert("XSS") + ""}  
{"a": "\ " + alert("\XSS\ ") + \ ""}
```

Cross-Site Scripting (XSS)

OBRANA

- Náhrada metaznaků “ ’ & < > za HTML entity <
- Escapování metaznaků v řetězcích JavaScriptu \'

Bezpečnostní hrozby



CRLF injection

Injekce HTTP hlaviček vložením bílých znaků CR+LF

Request:

seznam.cz?foo=test%0D%0Aheader:value

Response:

location: www.seznam.cz?foo=test

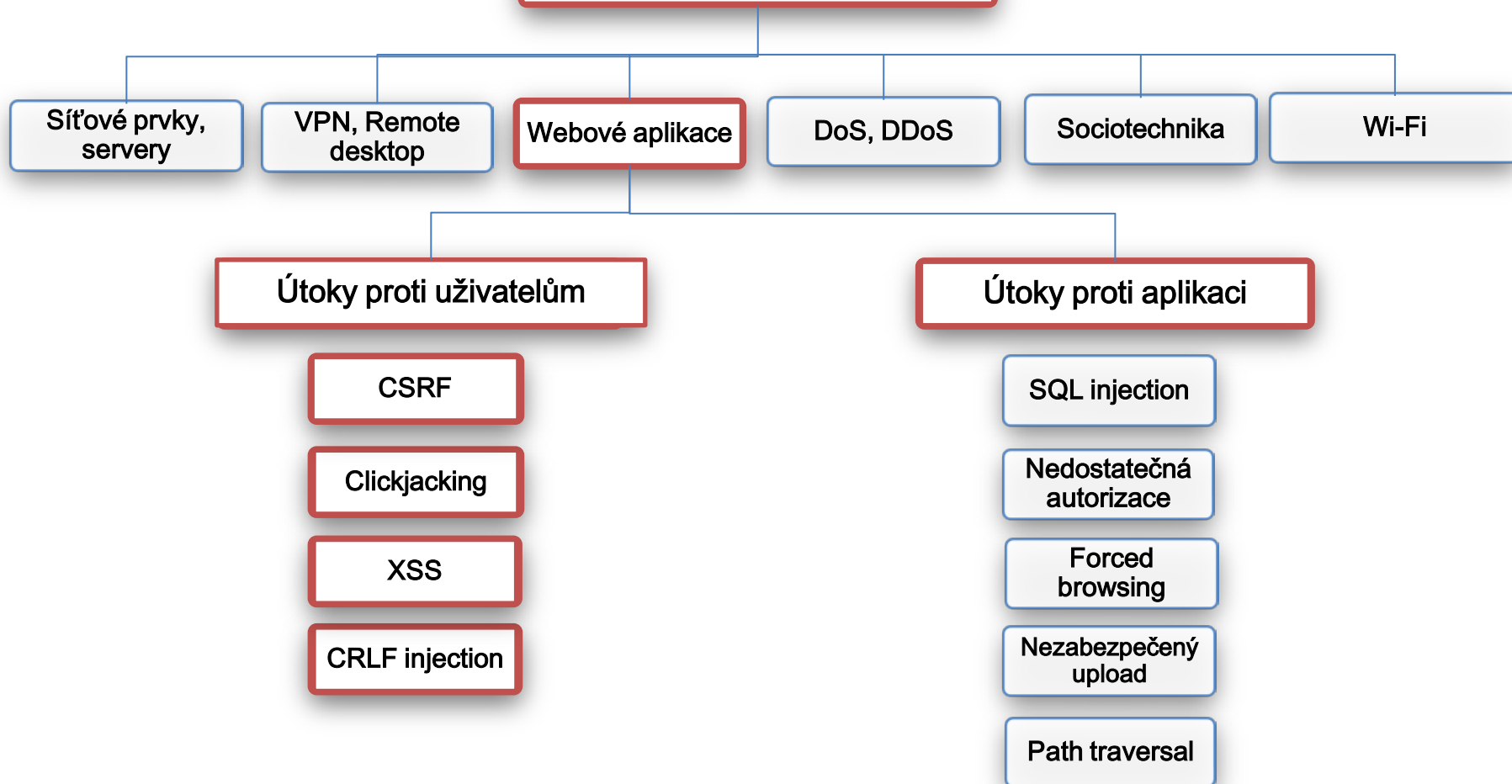
header: value

CRLF injection

OBRANA

- Ošetřit bílé znaky před vložením hodnoty do hlavičky
 - například URL encoding

Bezpečnostní hrozby



SQL injection

```
$name = $_GET['name']
```

```
$query = "SELECT * FROM database WHERE name='$name'"
```

```
www.webmail.cz?name=' OR 'a'='a
```

```
SELECT * FROM database WHERE name="" OR 'a'='a'
```

```
SELECT * FROM database WHERE name=""; DROP database--'
```

SQL injection

OBRANA

Escapovat metaznaky (apostrofy)

```
SELECT * FROM database WHERE name='\' OR \'A\'=\'A\'
```

Direktiva **magic_quotes_gpc**

SQL injection

`$id = $_GET['id']`

`$query = "SELECT * FROM database WHERE id=$id"`

`www.webmail.cz?id=1 or 1=1`

`SELECT * FROM database WHERE id=1 OR 1=1`

SQL injection

OBRANA

Přetypováním:

```
$id = (int)$_GET['id']
```

Uzavírání číselných hodnot do apostrofů + jejich ošetření

```
www.webmail.cz?id=1 or 1=1
```

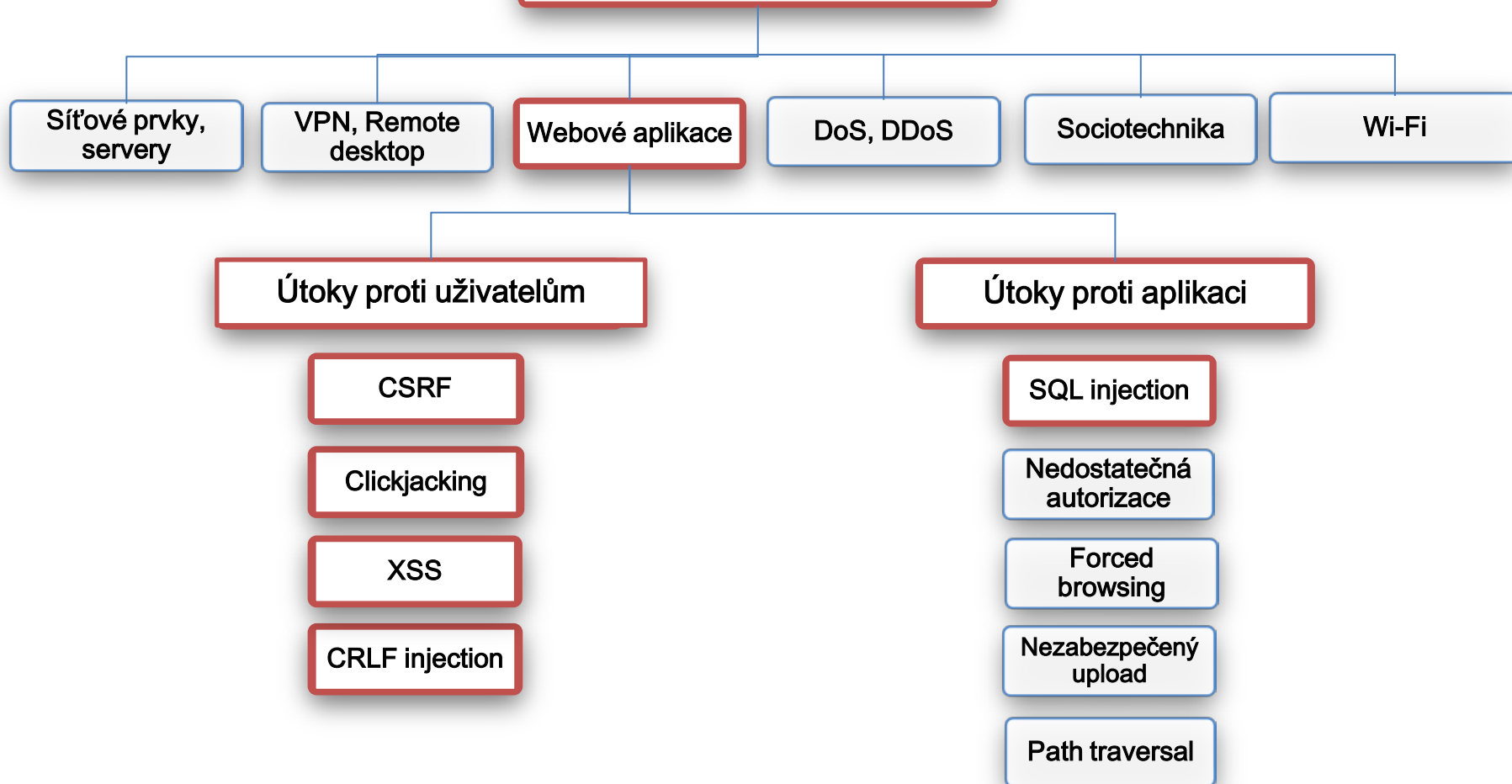
```
$query = "SELECT * FROM database WHERE id='$id'"
```

```
SELECT * FROM database WHERE id='1 OR 1=1'
```

```
www.webmail.cz?id=1' or '1'='1
```

```
SELECT * FROM database WHERE id='1\' or \'1\'=\'1'
```

Bezpečnostní hrozby



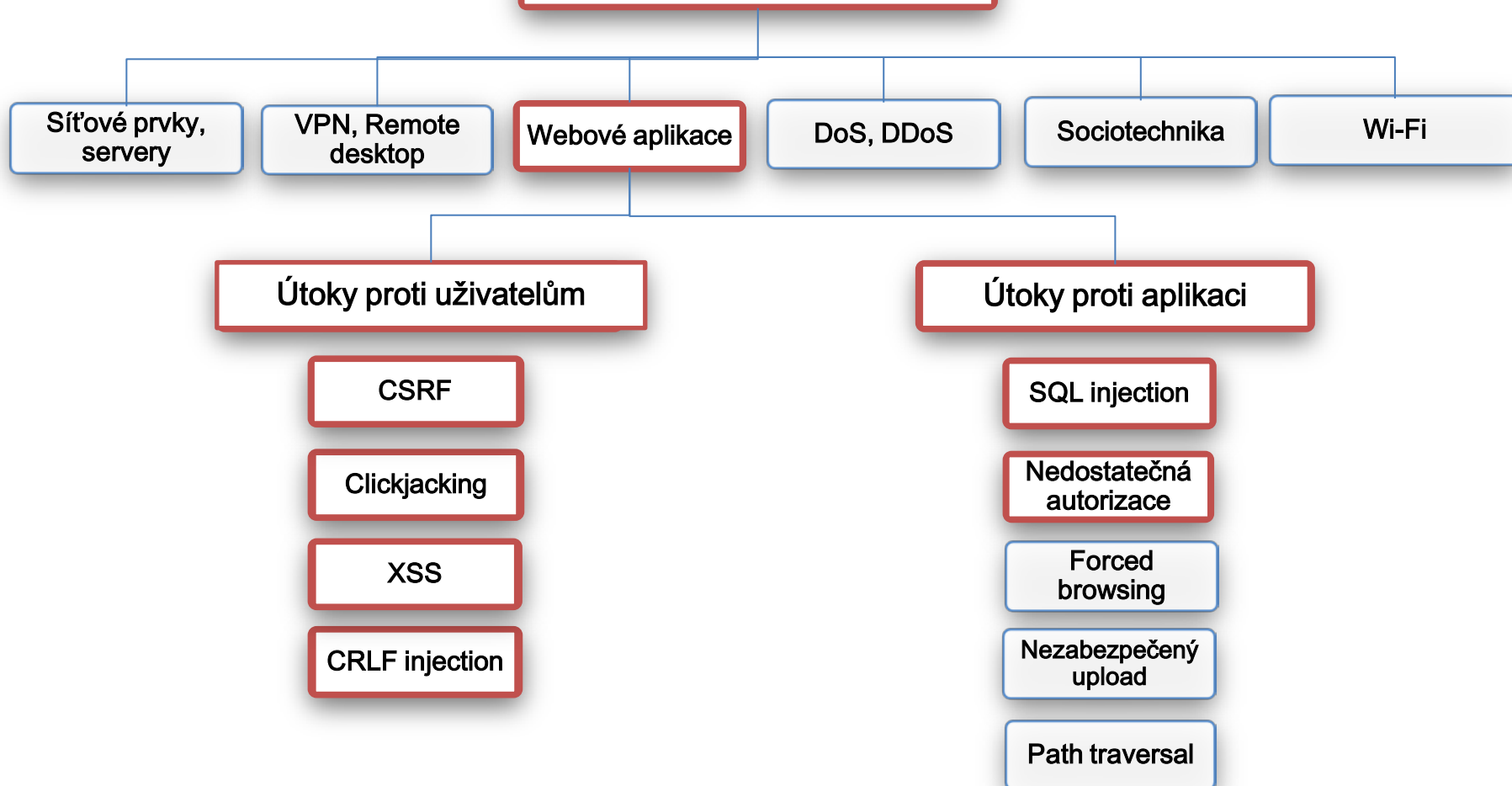
Nedostatečná autorizace

- Kontrola zda je uživatel přihlášen, ne zda má dostatečná práva
- Není kontrolováno, zda uživatel skutečně nakládá se svými daty
- Nejčastější útoky změnou id v požadavku

OBRANA

- Při každé akci provést všechny možné kontroly
 - Je uživatel přihlášen?
 - Nakládá s daty, ke kterým má povolen přístup?
 - Má uživatel povoleny akce jež se snaží vykonat?

Bezpečnostní hrozby



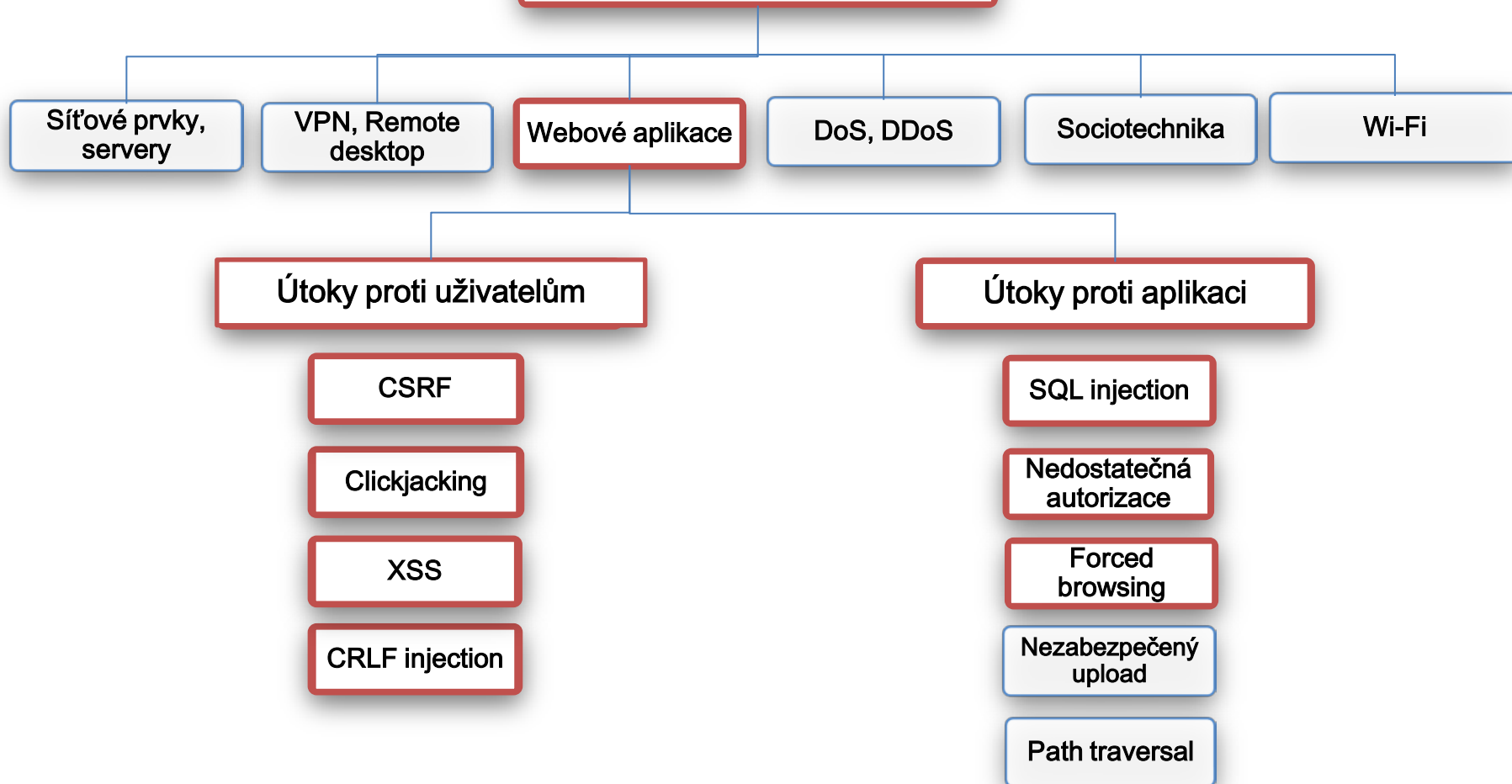
Forced browsing

- Procházení stránek změnou id v požadavku
- Možnost získání celých databází

OBRANA

- Nepoužívat id v požadavcích
- Používat jedinečný identifikátor

Bezpečnostní hrozby



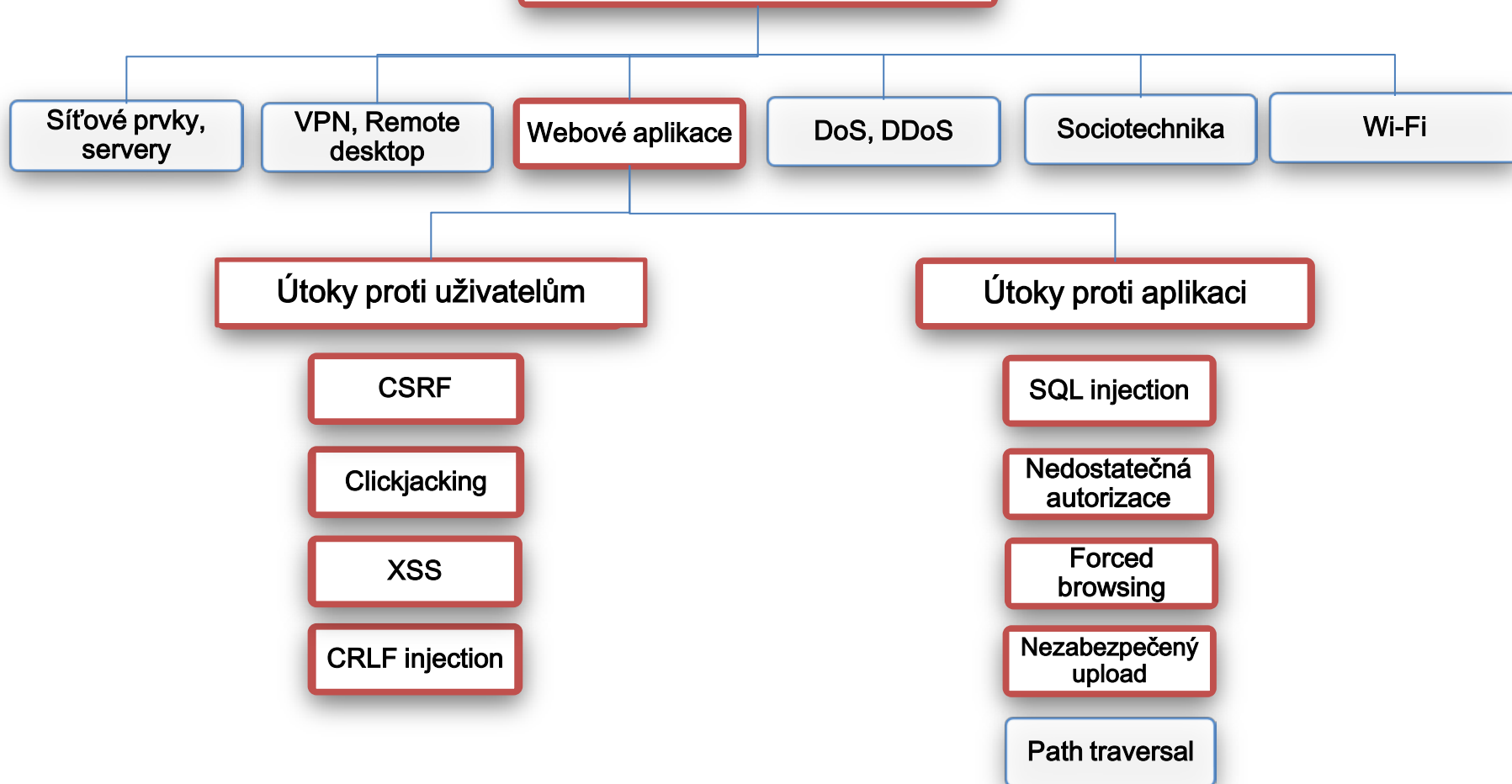
Nezabezpečený upload

- Upload serverového skriptu a jeho spuštění může vést k ovládnutí serveru
- Zneužití důvěryhodné domény k uploadu souborů, které slouží k infikování počítačů konečných uživatelů

OBRANA

- Kontrolovat typ a strukturu uploadovaných dat
- Konvertovat obrázky
- Ošetřit bílé znaky a znaky pro procházení adresářů v názvu uploadovaného souboru (použít nový název)
- Zakázat spuštění skriptů v adresářích pro upload

Bezpečnostní hrozby



Local File Inclusion (disclosure)

- Možnost zobrazení nebo spuštění serverového skriptu
- Script lze na server dostat:
 - Uploadem souboru
 - Uploadem nakaženého obrázku
 - Přes cookies a session proměnné
 - Enviroment (proc/self/environ)
 - Zneužitím logů (var/www/logs/access_log, apd.)

Local File Inclusion (disclosure)

GET: webmail.cz?action=view.php

```
$page = $_GET["action"];  
Include($page);
```

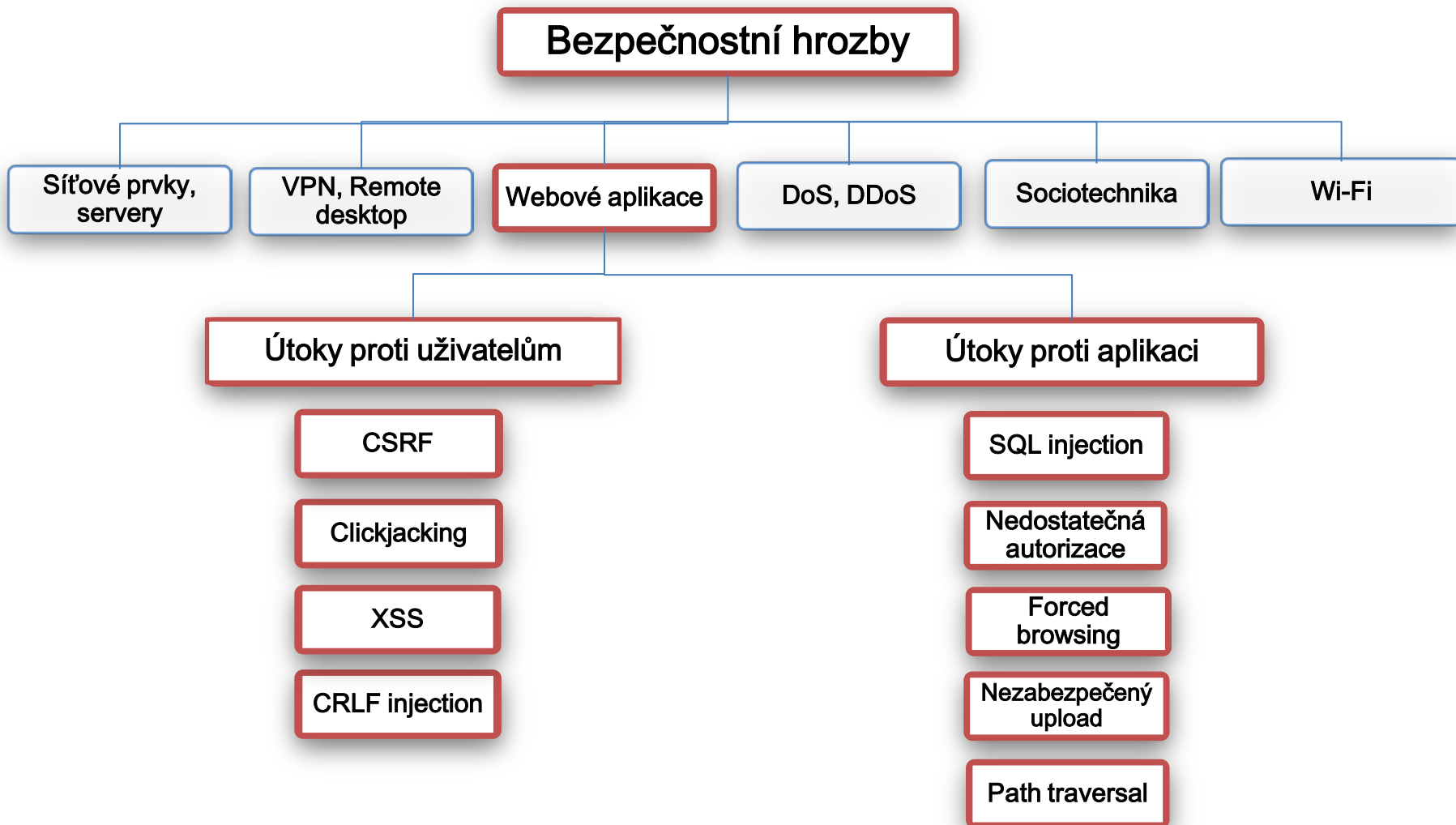
webmail.cz?action=../../../../etc/passwd

webmail.cz?action=../images/foto.jpg

Local File Inclusion (disclosure)

OBRANA

- Vhodně nastavený webový server
- Neincludovat soubory na základě proměnné obdržené od uživatele
- Ošetřit výskyt sekvence `../`



Webmail použitý v ukázkách je dostupný na adrese

<http://webmail.hackingvpraxi.cz>



Penetrační testování

Automatické

vs.

manuální testování

Automatické testování

- Klady

- Rychlost
- Odstraňují opakující se úkony (../ ../.. / ../..../ ../..../.. /)
- Hledání konfiguračních souborů
- Fuzzing
- apd.

- Zápory

- Nelze se na ně plně spolehnout
- Nedokáží odhalit reakce projevující se na jiných místech
- Při důkladných testech se nelze manuálním testům vyhnout

Manuální testování

- Klady

- Důkladná analýza výstupu umožní lépe přizpůsobit následné testy
- Je možné odhalit i reakce projevující se na jiných místech aplikace
- Na důkladně provedené testy se lze spolehnout

- Zápory

- Pentester je jen člověk a může něco přehlédnout
- Opakující se činnosti je časově náročné a vyčerpávající

Jak zajišťujeme bezpečnost aplikací v Seznam.cz

Jak zajišťujeme bezpečnost aplikací v Seznam.cz

- Vzděláváním vývojářů
- Penetračním testováním betaverzí
- Průběžnou kontrolou ostrých verzí
- Komunikací s white hat hackery
(spolupráce se SOOM.cz)



Dotazy

Děkuji za pozornost