|  |
| --- |
| ***File Include*** |
| Kaan Efe Öğüt  *ADLİ BİLİŞİM MÜHENDİSLİĞİ* |
| File Include nedir ve bWAPP& Web for Pentester laboratuvarları üzerinde File Include zafiyeti nasıl uygulanır ve bunun hakkında bilgi vereceğim. |

**29.07.2021**

***FİLE INCLUDE***

-Türkçesine baktığımızda dosya dahil etme olarak basitleştirebiliz.Hedef sunucuda bulunan veya sunucuda bulunan dosyalarn hedef sunucuda çalışmasına izin veren bir web uygulama açığıdır.

-LFI(Local File İnclude) : Sitenin bulunduğu sunucudaki dosyalar çalıştırılabilir.

-RFI(Remote File İnclude) : Uzak sunucuda bulunan dosyalar hedef sunucudaymış gibi çalıştırılıp hedef sunucuyu ele geçirme olarak tanımlayabiliriz.

Örnek vermek gerekirse;

-/etc/passwd dosyası ile sistemde bulunan kullanıcı ve yetkileri görüntülüyebiir

-Veritabanı bağlantı bilgilerini barındıran dosyayı mevcut sayfaya dahil ederek veritabanı şifresi öğrenilebilir.

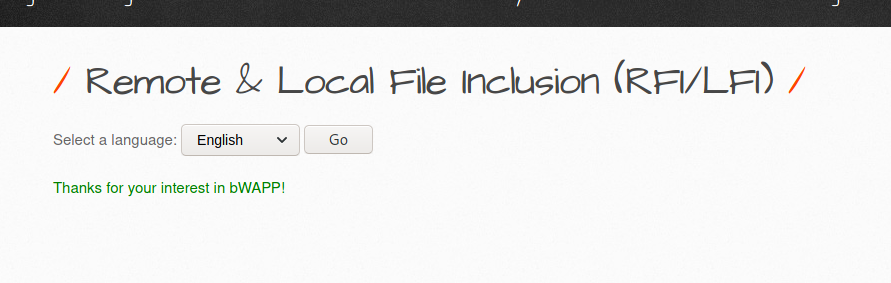
-Saldıran yetkisi olmayan dizinleri gezinerek hassas bilgilere ulaşabilir.

-Zararlı bir kod içeren bir dosya dahil ederek işlenmesi sağlanır.

Teorik olarak bilgi verdikten sonra uygulamaya geçersek;

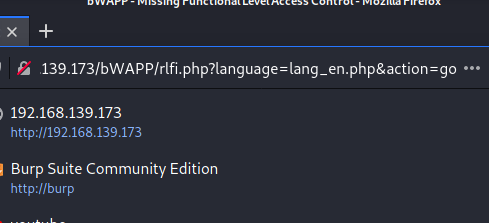
-bWAPP üzerinden 1 WFP üzerinden 2 adet olmak üzere toplamda 3 adet uygulama gerçekleştireceğiz.

**bWAPP Uygulama 1**



-Bwapp üzerinden zaafiyetimi başlatıyorum.

-Açılan sayfayı incelediğimde basic olarak dil seçip çalıştırdığım bir sayfa.



-İlk olarak linke göz atıyorum.İçerisinde bir .php dosyası olduğunu görüntülüyorum.Bu php dosyasını değiştirip bir zafiyet elde edebilirim.

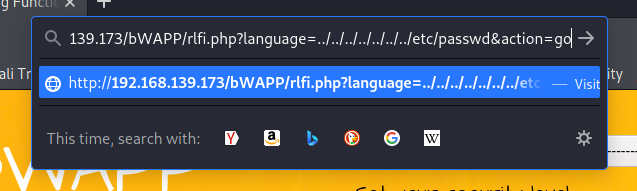


- Açık kaynak kodunu görüntülüyorum.Seçilen dile göre güvenlik paketlerine dağılıyor.

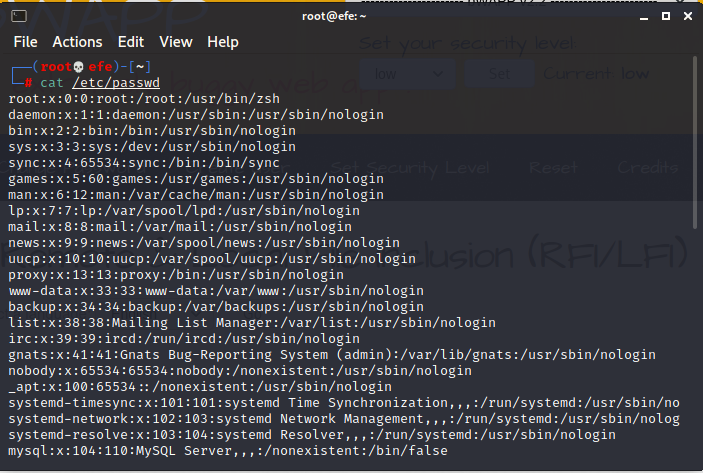
-Öncelikle dil seçiyoruz ve seçtiğimiz paketin get metodu ile gönderimi sağlanıyor.



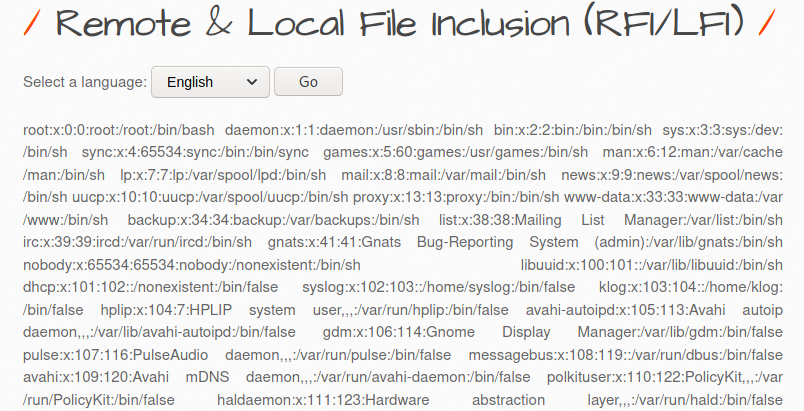
-Security\_Level 0 olduğu için else kısmına giriyor ve direk include ediliyor.



-URL sonuna “../../../../../../../etc/passwd” kodunu ekliyorum ve zafiyete uygun hale getiriyorum.

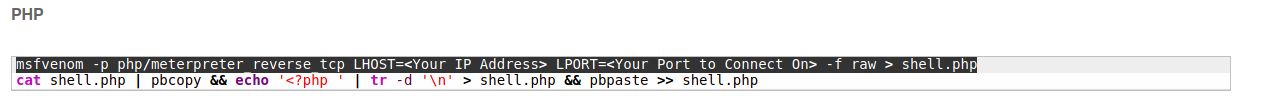


-Döndürebileceği değerleri öğrenmek için ilk önce kendi sanal makinem üzerinde “cat /etc/passwd” komutu ile döndürülen değerleri inceliyorum.

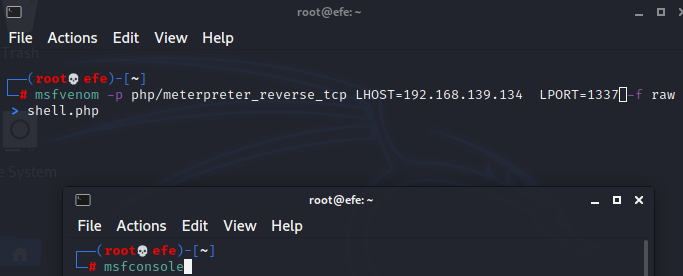


-URL üstünden oynadığım zaafiyet sonucunda passwd ekranında ki paketlerin hepsi önüme geldi.

.



-Tarayıcı üzerinden Msfvenom ile dinleme işlemi için Reverse Shell oluşturuyorum ve PHP dosya kısmını kopyalıyorum.



-Msfvenom üzerinden kopyaladığım komutu uçbirime yapıştırıyorum.IP numaram ve kendi belirlediğim port adresini ekliyorum.

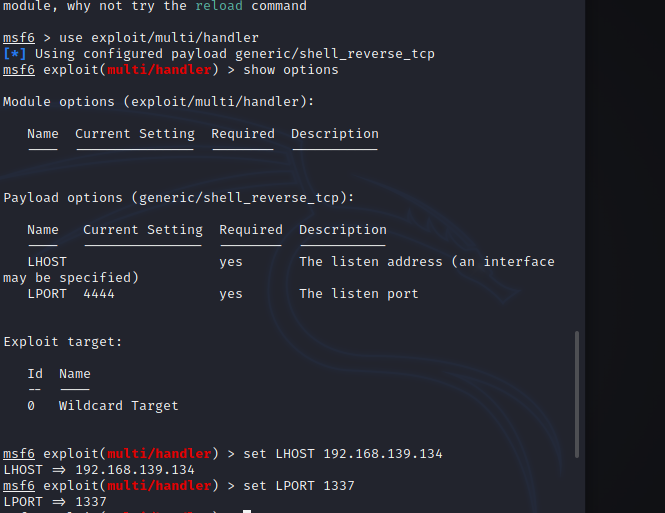
-Msfconsole ile msf üzerinden işlemlerime başlayacağım.



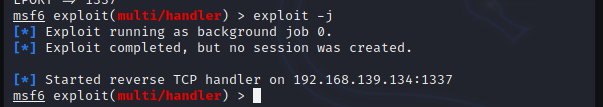
-Bağlantı başarılı bir şekilde sağlandı fakat .php şeklinde işlem yaptığımda sunucu bunu okuyamayacak bu yüzden txt ye çeviriyorum.

- .Txt uzantısına dönüştürmek için “mv shell.php Shell.txt” komutunu çalıştırıyorum.

.



-Msfvenom aracı üzerinden metasploit multihandler ile IP adresi ve Port’u set ediyorum.

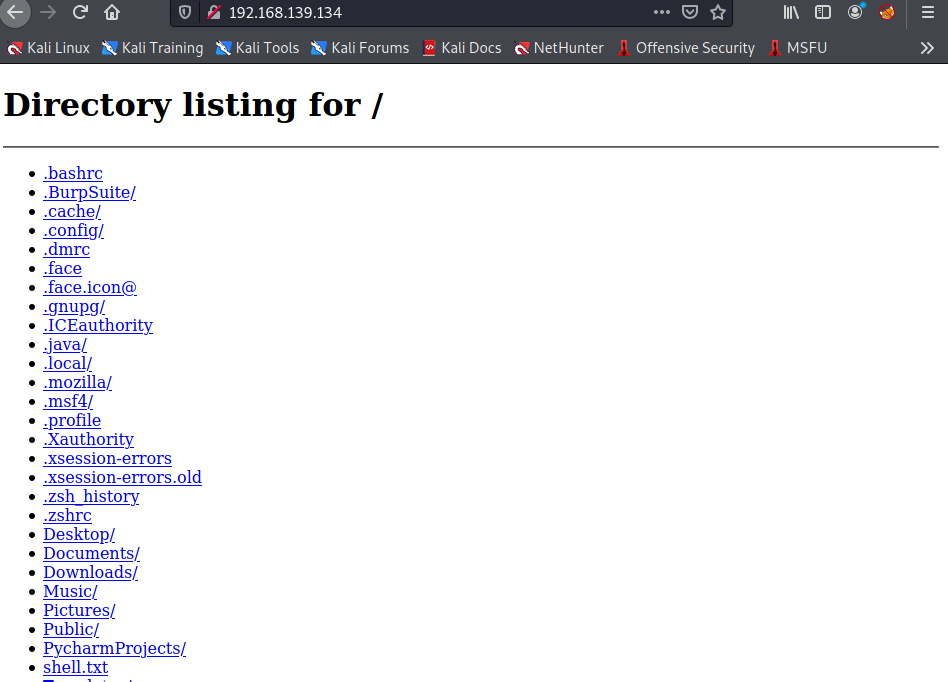


-Metasploit işlemimi başlatıyorum ve ardından shell.txt'mi internete açıyorum.

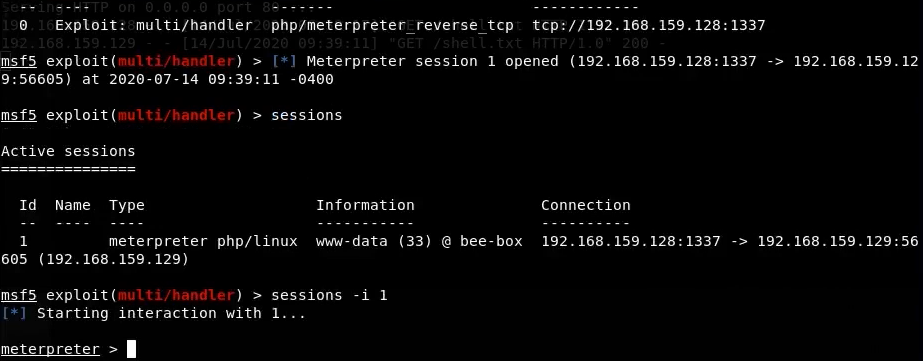


-Python üzerinden simpleHTTPserver’ı başlatıyorum.

-Shell dosyası üzerinde gerekli işlemleri sağladıktan sonra şimdi sırada sunucu üzerine ekleyip Shell.php’yi çalıştırma işleminde;



-Reverse Shell aracını başarılı bir şekilde sunucuya ekleyebildim.



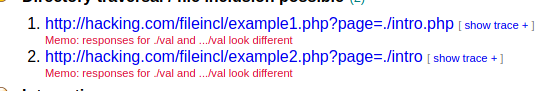
-İşlem yapıldığı zaman meterpreter üzerinden yapılan tüm işlemler kayıt altına alınacaktır.

**WFP Uygulama 1 :** LFI

-Diğer zafiyetlerde olduğu gibi önce zaproxy ve skipfish araçları ile zafiyet testi yapıyorum



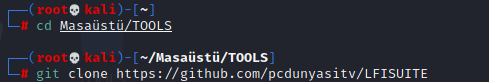
-Zaproxy aracı zafiyeti tespit etti fakat farklı bir kategoride bu zafiyeti sınıflandırdı.



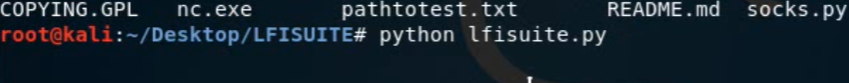
-Skipfish aracı da zafiyeti başarılı bir şekilde tespit etti.

-Zaproxy ve Skipfish araçlarını inceledim fakat bana yeterli bir bilgi döndürmedi.Bu sebeple farklı bir araç üzerinden işlem gerçekleştireceğim.

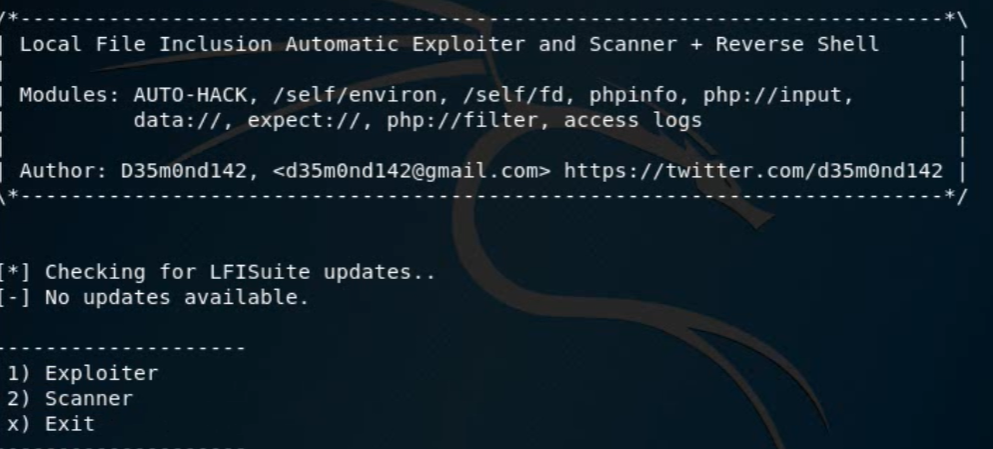
-LFISUITE araci FileInclude zafiyetlerinde açık bulmamıza yardımcı oluyor.



-Git clone üzerinden indirme işlemini başlatıyorum.



-İndirme işleminin ardından python komutu ile uygulamayı başlatıyorum.

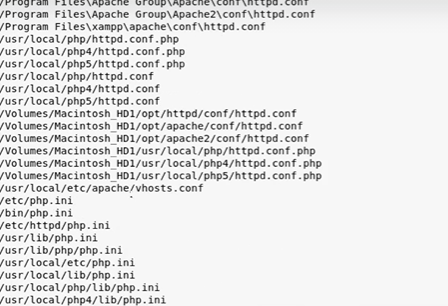


-Çalıştırma işleminden sonra karşımıza çıkan menüde bizi 2 adet seçenek karşılıyor.

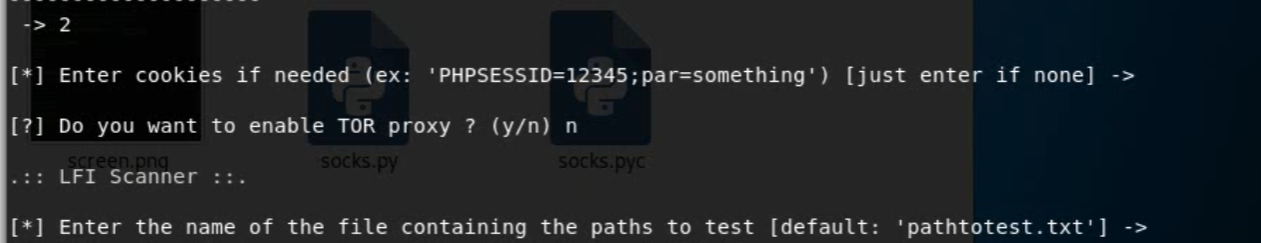
1-)Exploiter açığı tespit ettikten sonra sızın işlemini gerçekleştirecek araç

2-)Scanner ise açığı tespit etmemizi sağlayacak olan araçtır.

-SQL ve XSS zafiyetlerinde olduğu gibi bu zafiyet içinde payload kullanımı gerçekleştiricez.Bu zafiyete de tarayıcı üzerinden erişim sağlayabiliriz.



-Payload’ın içeriğini az da olsa göstermek istedim.



-Payload’ın yerini belirtiyorum.

-Daha sonrasında zafiyet bulunan sitenin URL’ini istiyor.

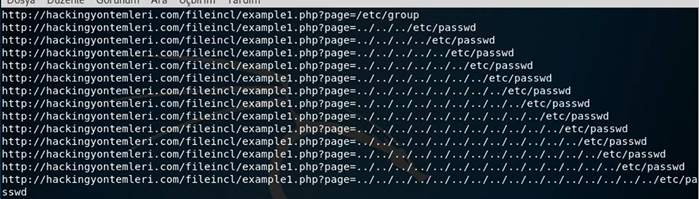


-Web for Pentester üzerindeki Example1 linkini kopyalıyorum.

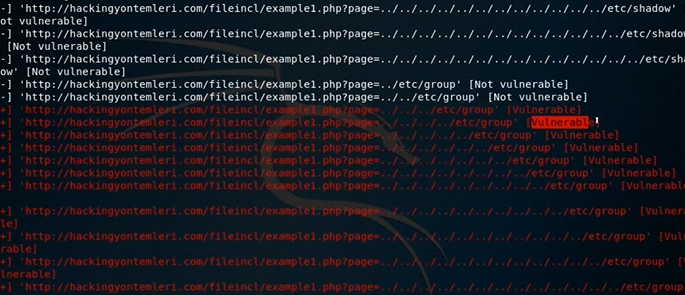


-Linki araç üzerine yapıştırıp işlemi başlatıyorum.Programımız kendi .txt dosyasında bulunan payloadların hepsini tek tek deneme işlemine başlayacaktır.

-Yaklaşık bir 10-15 dakika sonra zafiyet testim sonuçlandı.



-27 tane zaafiyet buldu ve linklerini tarafıma aktardı.Ancak 1 taneside benim işimi görüyor.



-Ayrıca bulamadığı kısımlarda Not vulnerable ve Vulnerable komutları yer alıyor.

-Döndürdüğü linkler üzerinden de işlem yapabiliriz fakat ben program üzerinden devam etmek istiyorum.

-LFISUITE uygulamasını tekrar çalıştırıyorum.



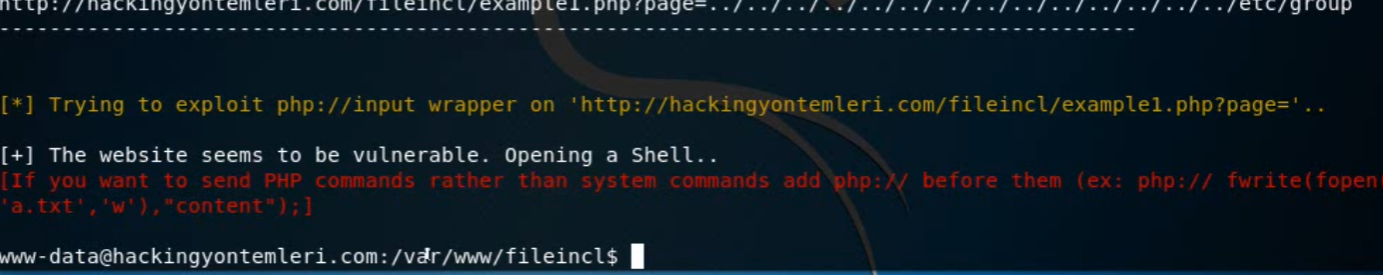
-Programı çalıştırdım fakat bu sefer 1 ile devam ediyorum ve cookie olmadığı için orayı yine boş geçiyorum.



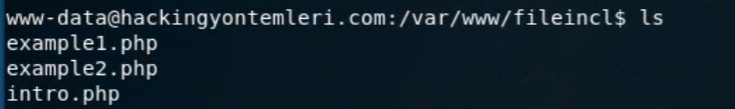
-Karşımıza uygulanabilir enjeksiyon türleri geliyor.Biz auto-hack seçeneği ile devam ediyoruz.Autohack seçeneği hepsini tek tek deneyecektir.



-Az önce scanner üzerinden aldığımız link üzerinden devam edebileceğimiz gibi WFP kısmındaki link üzerinden de işleme devam edebiliriz.



-Yapmış olduğumuz işlemler sonucunda hacking adresine başarılı bir şekilde sızma işlemini gerçekleştirdik

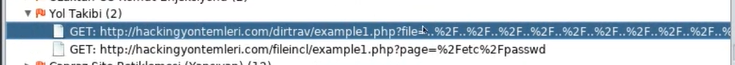


-Basit bir şekilde ls komutunu çalıştırırsak zafiyetten yararlandığımızı görüntüleyebiliriz.

**WFP UYGULAMA 2 :** RFI

-RFI açığını kullanarak başka sunucuya shell üzerinden bağlantı kurmaya çalışacağım.

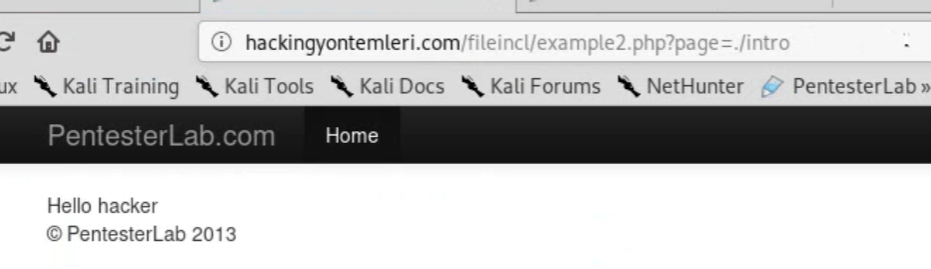
-Diğer zafiyetler üzerinde de yaptığım gibi öncelikte zafiyet testlerime bakıyorum.



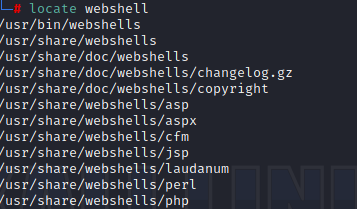
-Zaproxy aracını incelediğimde zafiyeti tespit edemedi.



-Skipfish üzerinde zafiyet tespiti başarılı bir şekilde yapıldı.

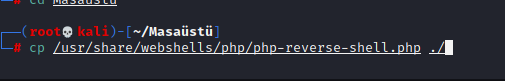


-Site URL’sini incelediğimizde zaafiyetle diğer zaafiyetin farkı linkin sonunda .php olmamasıdır.



-“locate webshell” komutu ile Linux üzerinden kullanabileceğim shell toollarına erişim sağlıyorum.

-Php-reverse-shell.php ile işlem yapacağım için onu kopyalıyorum.Bu Shell benim uzakta ki sunucuya erişim sağlamamı sağlıyor.



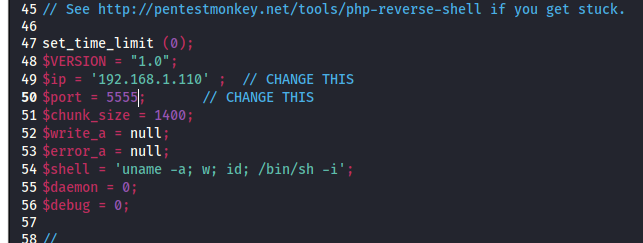
-Shell dosyasını masaüstüne kopyalıyorum.



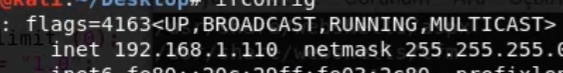
-İsmini “shell.php” olarak değiştiriyorum.



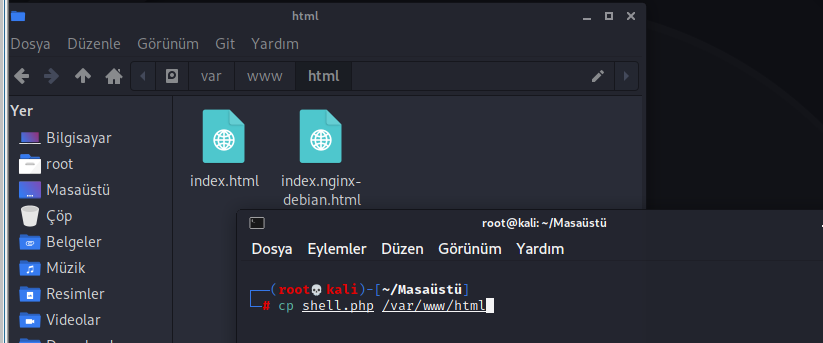
-“gedit” komutu ile .php dosyasını açıp düzenleme işlemine başlıyorum.



-Shell.php üzerinde kendi ip adresimi ekliyorum ve port adresini rastgele olarak 5555 olarak veriyorum.



-“ifconfig” komutu ile Ip adresimi öğrenebilirim.



-Kali linux üzerinde kurulu olarak gelen Apache server üzerinde işlem yapmak için Apache server’ın kullandığı dizine kopyalıyorum.

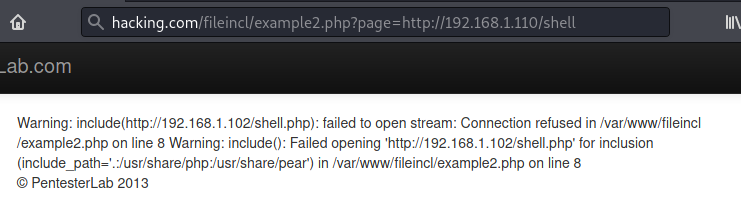
**Not :** Daha önceden Apache Server işlemi yapmadıysan bu servisi başlatmanız gerekmektedir.



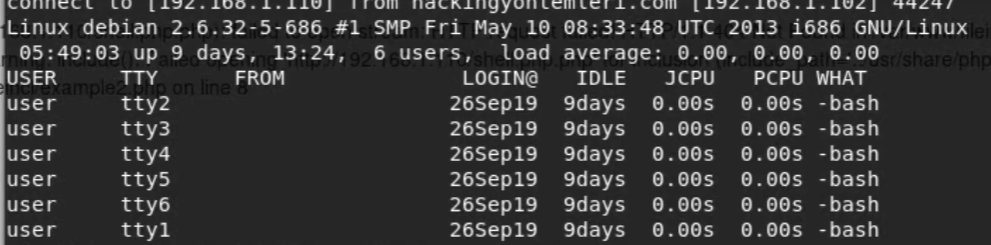
-Apache serverımız uzak sunucu oldu.Bu dosyanın gideceği portu netcat üzerinden dinliyorum ki shell dosyasını aldığında bilgi alabileyim.Port olarak 5555 verdiğim için o portu dinliyorum.



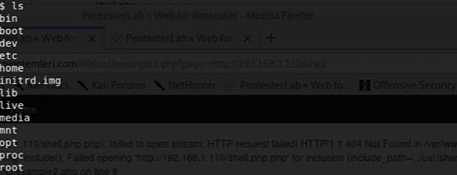
-Linkimi ip adresim ve dosya ismime göre ayarlayıp enter'ı tuşluyorum.



-Sayfanın son görünümü bu şekilde fakat asıl işlem netcat üzerinde gerçekleşicek.



-Netcat üzerinde başarılı bir şekilde bağlantı kurabildim.Şİmdi ise bir komut çalıştırarak bağlantıyı test edelim.



-“LS “komutunu çalıştırdığımda başarılı bir şekilde işlem yapabiliyorum.

-Zafiyet başarıyla gerçekleşti artık istediğimiz Linux komutunu uygulayabiliriz.

***Korunma Yöntemleri***

-Kullanıcıdan alınan girdiler ve ekrana bastırılacak çıktılar kontrol edilmelidir.

-Kullanıcının yüklediği dosyaların uzantıları kontrol edilmeli, gerekirse uzantılar değiştikten sonra sunucuda saklanmalıdır.

-Kullanıcıya gösterebilecek veya kullanıcı tarafından çalıştırılabilecek dosyaların beyaz listesi oluşturulabilir.

-Olası saldırı simülasyonları ile test yapılmalıdır.

-IDS/IPS veya WAF Kullanılmalıdır.