



techcareer.net

Cyber Security Bootcamp Bitirme Ödevi

Sonuç Raporu

Rapor Detayları

Rapor Başlığı	Cyber Security Bootcamp Bitirme Ödevi Sonuç Raporu
Versiyon	1.0.0
Yazan	Ömer Mert GÜLSEVEN
Kontrol Eden	Ammar KARABULUT
Onaylayan	Ammar KARABULUT

Rapor İçeriği

- Kapsam
- Tespit Edilen Güvenlik Zafiyetlerinin Özet Tablosu
- Risk Seviyelerine Göre Zafiyetlerin Dağılımı Grafiği
-

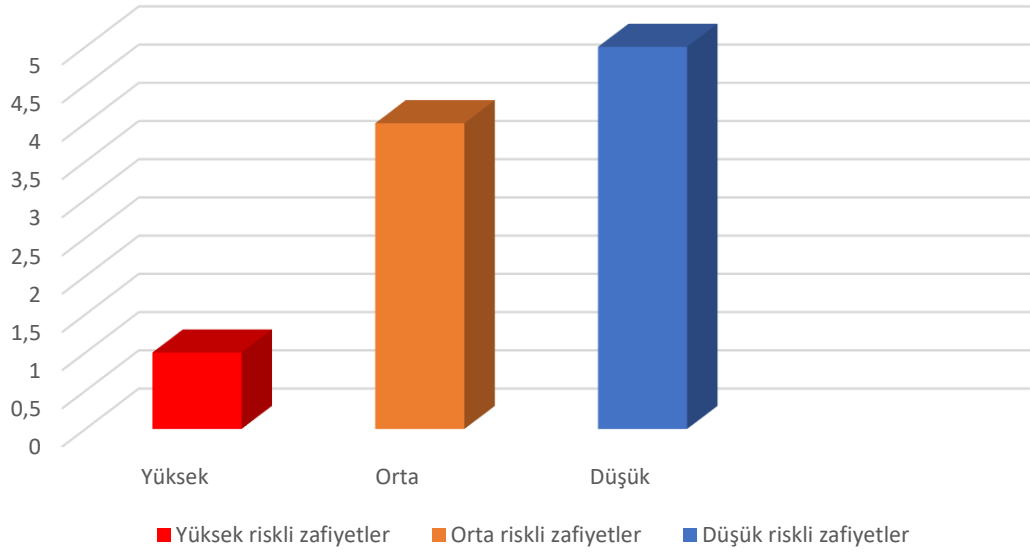
KAPSAM

Verilen bu ödev kapsamında ilgili sitede (<http://php.testsparker.com/process.php?file=Generics/index.nsp>) çeşitli zafiyet araştırmaları yapılmış ve sitede tespit edilen zafiyetler listelenmiştir. Zafiyet taramasına ek olarak port taraması da yapılmış ve açık portlar hakkında bilgi toplanmıştır.

Tespit Edilen Güvenlik Zafiyetlerinin Özet Tablosu

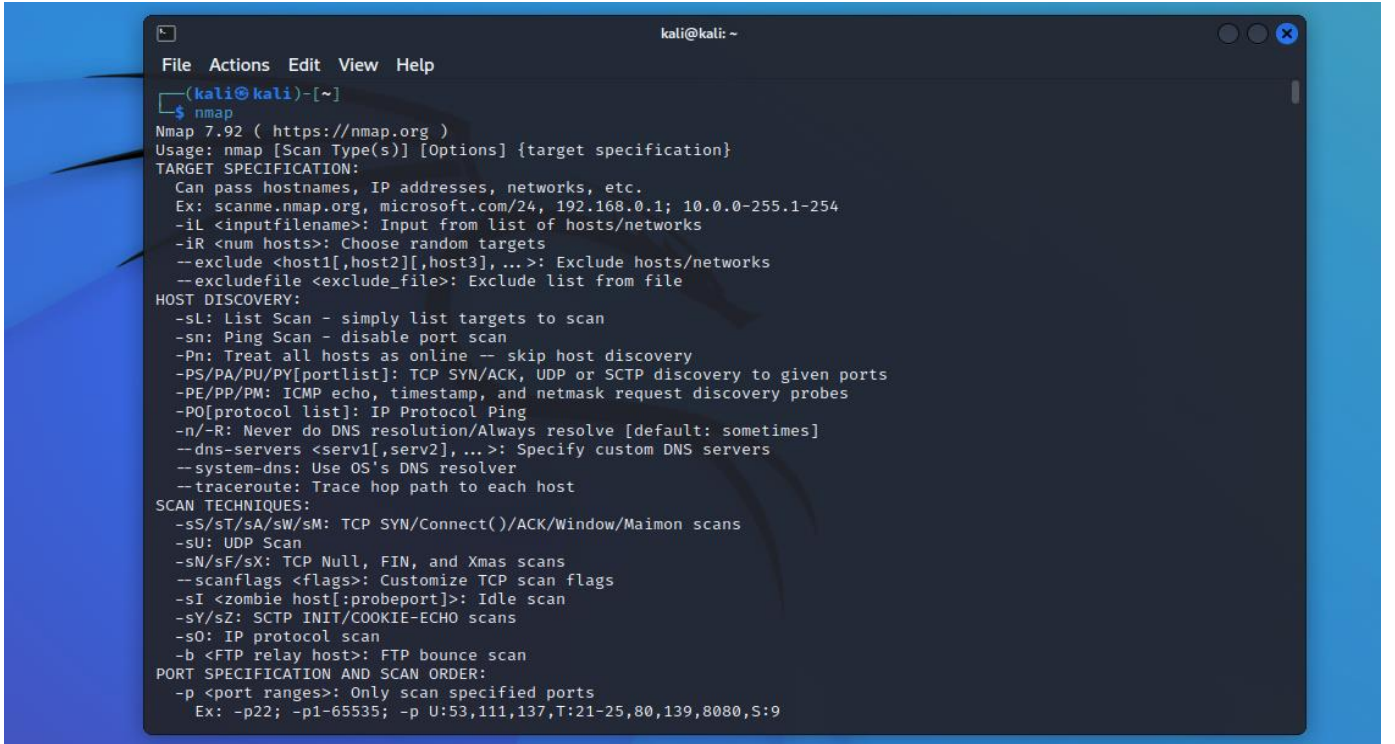
Bulgu Adı	Önem Derecesi	Bulgu Kategorisi
Siteler arası komut dosyası çalıştırma saldırısı (Dom XSS)	Yüksek	Web
Siteler Arası İstek Sahtekarlığı (CSRF - Cross Site Request Forgery) Absence of Anti CSRF Tokens	Orta	Web
Application Error Disclosure (Uygulama Hata Bildirimi)	Orta	Web
İçerik Güvenliği Politikası (CSP-Content Security Policy)	Orta	Web
Missing Anti-clickjacking header	Orta	Web
Cookie No HttpOnly Flag	Düşük	Web
Cookie only SameSite attribute	Düşük	Web
Information Disclosure- Bilgi ifşası	Düşük	Web
Private IP Disclosure- Özel IP Adresi Açıklaması	Düşük	Web
Server Leaks Information on via "X Powered By" Http Response Header Fields	Düşük	Web

Risk Seviyelerine Göre Zafiyetlerin Dağılımı



Port Taraması Hakkında

Port taraması yapabilmek için bir Windows makine üzerinde veya bir sanal makine içerisinde kurulu olan Kali Linux işletim sistemi içerisinde Nmap isimli araç kurulu olmalıdır. Bu işlem için Nmap isimli araç kullanılacaktır.

A screenshot of a Kali Linux terminal window. The window title is 'kali@kali: ~'. The terminal shows the command 'nmap' being entered, followed by the Nmap 7.92 help text. The help text includes usage instructions, target specification options, host discovery options, scan techniques, and port specification options.

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap  
Nmap 7.92 ( https://nmap.org )  
Usage: nmap [Scan Type(s)] [Options] {target specification}  
TARGET SPECIFICATION:  
  Can pass hostnames, IP addresses, networks, etc.  
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254  
  -iL <inputfilename>: Input from list of hosts/networks  
  -iR <num hosts>: Choose random targets  
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks  
  --excludefile <exclude_file>: Exclude list from file  
HOST DISCOVERY:  
  -sL: List Scan - simply list targets to scan  
  -sn: Ping Scan - disable port scan  
  -Pn: Treat all hosts as online -- skip host discovery  
  -PS/PA/PY/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports  
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes  
  -PO[protocol list]: IP Protocol Ping  
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]  
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers  
  --system-dns: Use OS's DNS resolver  
  --traceroute: Trace hop path to each host  
SCAN TECHNIQUES:  
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans  
  -sU: UDP Scan  
  -sN/sF/sX: TCP Null, FIN, and Xmas scans  
  --scanflags <flags>: Customize TCP scan flags  
  -sI <zombie host[:probeport]>: Idle scan  
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans  
  -sO: IP protocol scan  
  -b <FTP relay host>: FTP bounce scan  
PORT SPECIFICATION AND SCAN ORDER:  
  -p <port ranges>: Only scan specified ports  
  Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
```

Yukarıda yer alan görüntüde Kali Linux üzerinde Nmap çalıştırılıyor. Ardından karşımıza uzunca bir menü çıkıyor ve buradan yararlanarak yapmak istediğimiz işlemi seçerek devam ediliyor.

```
kali@kali: ~  
File Actions Edit View Help  
~(kali@kali)-[~]  
$ nmap --packet-trace php.testsparker.com  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-10 10:59 EDT  
CONN (0.1999s) TCP localhost > 107.20.213.223:80 => Operation now in progress  
CONN (0.2000s) TCP localhost > 107.20.213.223:443 => Operation now in progress  
s  
CONN (0.4281s) TCP localhost > 107.20.213.223:80 => Connected  
NSOCK INFO [0.4280s] nsock_ioc_new2(): nsock_ioc_new (IOD #1)  
NSOCK INFO [0.4280s] nsock_connect_udp(): UDP connection requested to 192.168  
.112.2:53 (IOD #1) EID 8  
NSOCK INFO [0.4280s] nsock_read(): Read request from IOD #1 [192.168.112.2:53  
(timeout: -1ms) EID 18  
NSOCK INFO [0.4280s] nsock_write(): Write request for 45 bytes to IOD #1 EID  
27 [192.168.112.2:53]  
NSOCK INFO [0.4280s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS  
S for EID 8 [192.168.112.2:53]  
NSOCK INFO [0.4280s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS  
for EID 27 [192.168.112.2:53]  
NSOCK INFO [0.5140s] nsock_trace_handler_callback(): Callback: READ SUCCESS f  
or EID 18 [192.168.112.2:53] (101 bytes)  
NSOCK INFO [0.5140s] nsock_read(): Read request from IOD #1 [192.168.112.2:53  
(timeout: -1ms) EID 34  
NSOCK INFO [0.5140s] nsock_ioc_delete(): nsock_ioc_delete (IOD #1)  
NSOCK INFO [0.5140s] nsock_delete(): nsock_delete on event #34 (type READ)  
CONN (0.5149s) TCP localhost > 107.20.213.223:554 => Operation now in progress  
s  
CONN (0.5150s) TCP localhost > 107.20.213.223:1723 => Operation now in progress  
ss  
CONN (0.5150s) TCP localhost > 107.20.213.223:5900 => Operation now in progress  
ss  
CONN (0.5151s) TCP localhost > 107.20.213.223:139 => Operation now in progress  
s  
CONN (0.5152s) TCP localhost > 107.20.213.223:25 => Operation now in progress  
CONN (0.5153s) TCP localhost > 107.20.213.223:993 => Operation now in progress  
s  
CONN (0.5153s) TCP localhost > 107.20.213.223:8080 => Operation now in progress  
ss  
CONN (0.5154s) TCP localhost > 107.20.213.223:995 => Operation now in progress  
s  
CONN (0.5154s) TCP localhost > 107.20.213.223:113 => Operation now in progress  
s  
CONN (0.5155s) TCP localhost > 107.20.213.223:1025 => Operation now in progress  
ss  
CONN (2.6576s) TCP localhost > 107.20.213.223:1025 => Operation now in progress
```

Yukarıda yer alan görüntüdeki gibi `nmap --packet-trace <hedef site>` şeklinde bir komut girerek detaylı olarak bir port taraması işlemi başlatılıyor.

```
kali@kali: ~  
File Actions Edit View Help  
CONN (118.8878s) TCP localhost > 107.20.213.223:80 => Operation now in progress  
CONN (118.8978s) TCP localhost > 107.20.213.223:3517 => Operation now in progress  
CONN (119.2825s) TCP localhost > 107.20.213.223:9998 => Operation now in progress  
CONN (119.6666s) TCP localhost > 107.20.213.223:1105 => Operation now in progress  
CONN (120.0519s) TCP localhost > 107.20.213.223:1105 => Operation now in progress  
CONN (120.4367s) TCP localhost > 107.20.213.223:32771 => Operation now in progress  
CONN (120.8210s) TCP localhost > 107.20.213.223:32771 => Operation now in progress  
CONN (121.2061s) TCP localhost > 107.20.213.223:3052 => Operation now in progress  
CONN (121.5903s) TCP localhost > 107.20.213.223:3052 => Operation now in progress  
CONN (121.9752s) TCP localhost > 107.20.213.223:10616 => Operation now in progress  
CONN (122.3599s) TCP localhost > 107.20.213.223:10616 => Operation now in progress  
CONN (122.7447s) TCP localhost > 107.20.213.223:80 => Operation now in progress  
CONN (123.1300s) TCP localhost > 107.20.213.223:5989 => Operation now in progress  
CONN (123.5142s) TCP localhost > 107.20.213.223:5989 => Operation now in progress  
CONN (123.8984s) TCP localhost > 107.20.213.223:2000 => Operation now in progress  
CONN (124.2830s) TCP localhost > 107.20.213.223:2000 => Operation now in progress  
CONN (124.6678s) TCP localhost > 107.20.213.223:1138 => Operation now in progress  
CONN (125.0526s) TCP localhost > 107.20.213.223:1138 => Operation now in progress  
CONN (125.4368s) TCP localhost > 107.20.213.223:9090 => Operation now in progress  
CONN (125.8209s) TCP localhost > 107.20.213.223:9090 => Operation now in progress  
CONN (126.2057s) TCP localhost > 107.20.213.223:8222 => Operation now in progress  
CONN (126.5910s) TCP localhost > 107.20.213.223:8222 => Operation now in progress  
CONN (126.9750s) TCP localhost > 107.20.213.223:80 => Operation now in progress  
CONN (127.3591s) TCP localhost > 107.20.213.223:42 => Operation now in progress  
CONN (127.7432s) TCP localhost > 107.20.213.223:42 => Operation now in progress  
CONN (128.1282s) TCP localhost > 107.20.213.223:3322 => Operation now in progress  
CONN (128.5128s) TCP localhost > 107.20.213.223:3322 => Operation now in progress  
CONN (128.8970s) TCP localhost > 107.20.213.223:27355 => Operation now in progress  
CONN (129.2815s) TCP localhost > 107.20.213.223:27355 => Operation now in progress  
CONN (129.6659s) TCP localhost > 107.20.213.223:1185 => Operation now in progress  
CONN (130.0505s) TCP localhost > 107.20.213.223:1185 => Operation now in progress  
CONN (130.4352s) TCP localhost > 107.20.213.223:1914 => Operation now in progress  
CONN (130.8197s) TCP localhost > 107.20.213.223:1914 => Operation now in progress  
CONN (131.2045s) TCP localhost > 107.20.213.223:80 => Operation now in progress  
Nmap scan report for php.testspark.com (107.20.213.223)  
Host is up (0.16s latency).  
rDNS record for 107.20.213.223: ec2-107-20-213-223.compute-1.amazonaws.com  
Not shown: 997 filtered tcp ports (no-response), 1 filtered tcp ports (host-unreach)  
PORT      STATE SERVICE  
80/tcp    open  http  
443/tcp   open  https  
Nmap done: 1 IP address (1 host up) scanned in 131.59 seconds
```

Tarama işlemi başladıktan sonra bir süre beklemenin ardından sonuçlar görüntüleniyor. Yapılan tarama sonucunda hedef sitede 80 ve 443 portlarının açık olduğu tespit edilmiştir.

Port taramasının yanı sıra hedef makineye veya siteye versiyon taraması da yapılabilir. Bu tarama sayesinde yine açık portlar tespit edilebilir.

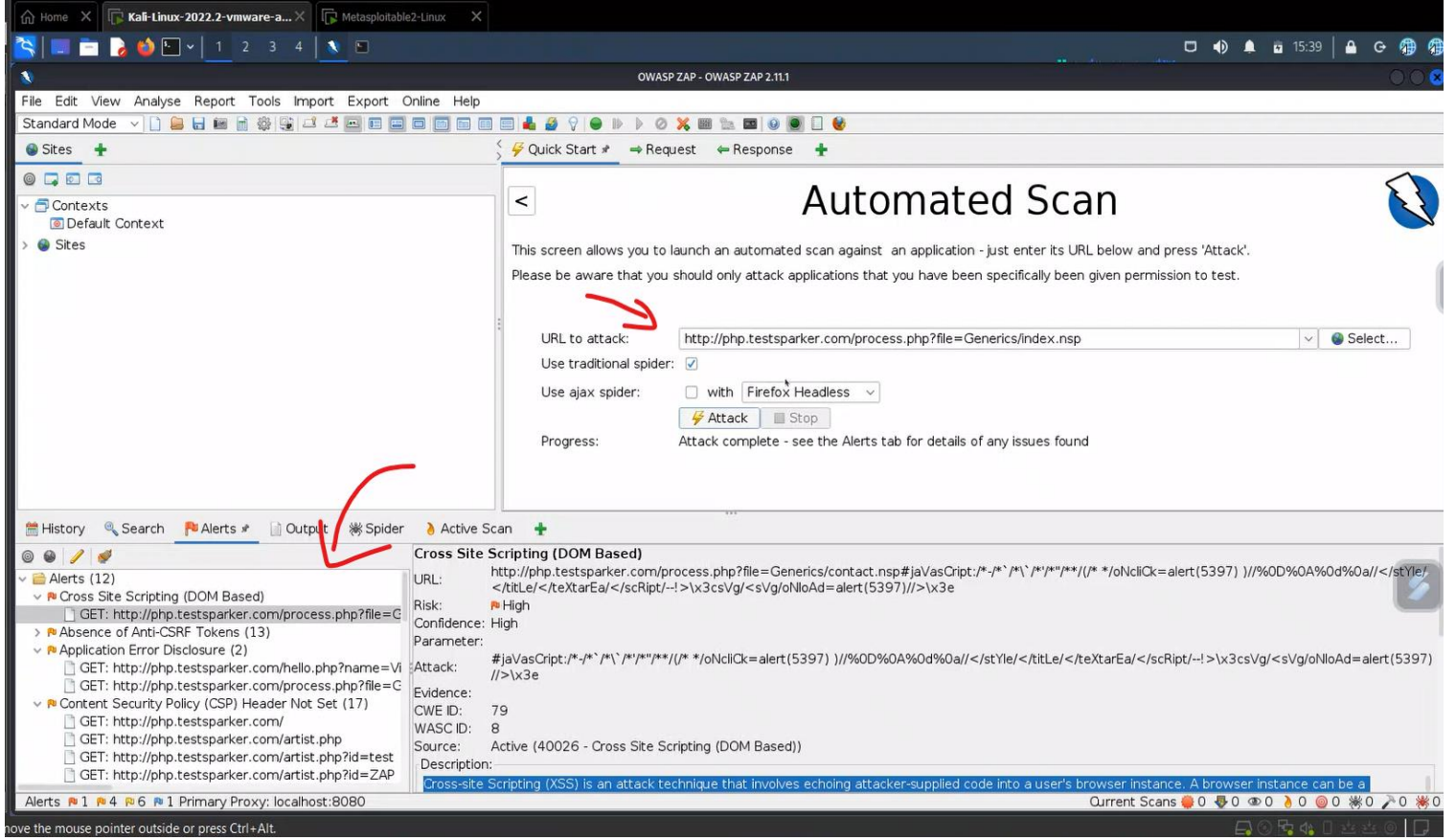

```
kali@kali: ~  
File Actions Edit View Help  
└─$ nmap -v -A php.testsparker.com  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-10 11:29 EDT  
NSE: Loaded 155 scripts for scanning.  
NSE: Script Pre-scanning.  
Initiating NSE at 11:29  
Completed NSE at 11:29, 0.00s elapsed  
Initiating NSE at 11:29  
Completed NSE at 11:29, 0.00s elapsed  
Initiating NSE at 11:29  
Completed NSE at 11:29, 0.00s elapsed  
Initiating Ping Scan at 11:29  
Scanning php.testsparker.com (107.20.213.223) [2 ports]  
Completed Ping Scan at 11:29, 0.13s elapsed (1 total hosts)  
Initiating Parallel DNS resolution of 1 host. at 11:29  
Completed Parallel DNS resolution of 1 host. at 11:29, 0.00s elapsed  
Initiating Connect Scan at 11:29  
Scanning php.testsparker.com (107.20.213.223) [1000 ports]  
Discovered open port 80/tcp on 107.20.213.223  
Discovered open port 443/tcp on 107.20.213.223  
Connect Scan Timing: About 31.95% done; ETC: 11:30 (0:01:06 remaining)  
Increasing send delay for 107.20.213.223 from 0 to 5 due to 11 out of 15 dropped probes since last increase.  
Increasing send delay for 107.20.213.223 from 5 to 10 due to 11 out of 16 dropped probes since last increase.  
Increasing send delay for 107.20.213.223 from 10 to 20 due to 11 out of 11 dropped probes since last increase.  
Increasing send delay for 107.20.213.223 from 20 to 40 due to 11 out of 11 dropped probes since last increase.  
Increasing send delay for 107.20.213.223 from 40 to 80 due to 11 out of 12 dropped probes since last increase.  
Completed Connect Scan at 11:31, 152.46s elapsed (1000 total ports)  
Initiating Service scan at 11:31  
Scanning 2 services on php.testsparker.com (107.20.213.223)  
Completed Service scan at 11:31, 5.01s elapsed (2 services on 1 host)  
NSE: Script scanning 107.20.213.223.  
Initiating NSE at 11:31  
Completed NSE at 11:31, 16.23s elapsed  
Initiating NSE at 11:31  
Completed NSE at 11:32, 7.36s elapsed  
Initiating NSE at 11:32  
Completed NSE at 11:32, 0.00s elapsed  
Nmap scan report for php.testsparker.com (107.20.213.223)  
Host is up (0.14s latency).  
rDNS record for 107.20.213.223: 223.213.20.107.in-addr.arpa  
Not shown: 998 filtered tcp ports (no-response)  
PORT      STATE SERVICE      VERSION  
80/tcp    open  tcpwrapped  
|_ http-server-header: Apache/2.2.8 (Win32) PHP/5.2.6  
443/tcp   open  tcpwrapped  
|_ http-title: Site doesn't have a title (text/html).  
|_ http-methods:  
|_ Supported Methods: OPTIONS  
|_ http-server-header: Apache/2.2.8 (Win32) PHP/5.2.6
```

Yukarıda yer alan görüntüde `nmap -v -A <hedef site>` komutu kullanılarak yapılan tarama sonucunda hedef siteye yada makineye ait işletim sistemi, kullanılan sistemlerin versiyonları gibi bilgilere erişmek mümkün.

Owasp Aracılığıyla Zafiyet Taraması İşlemi

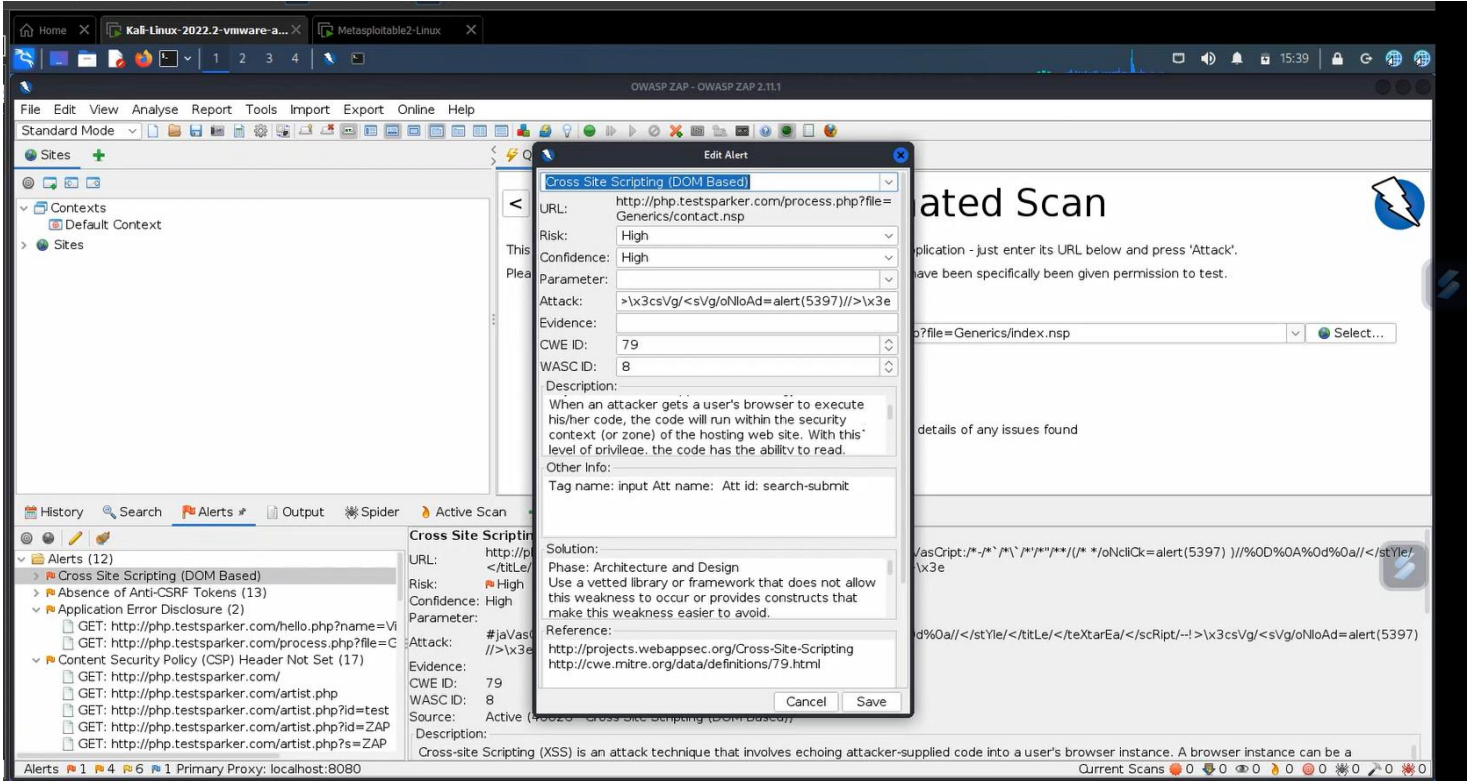
Öncelikle Owasp hakkında bilgi vermek gerekirse;

OWASP açılımı Open Web Application Security Project olarak tanımlanmaktadır. Web uygulamalarındaki güvenlik açıklarının kapatılması ve bu uygulamaların güvenli bir şekilde korunmasını sağlamak için çalışmalar yapan özgür bir topluluktur. OWASP'ye ait dokümanlar ve araçlar tüm dünyadaki herkesin kullanımına açık ve ücretsizdir. OWASP' in hiçbir özel şirket ve kuruluşla herhangi bir bağı yoktur. Çalışmalarını tamamen insanların ihtiyaçlarını gidermek üzere yürütmektedir. OWASP' in web sitesinde, web uygulamalarındaki zafiyetleri, bu zafiyetlerin nasıl oluştuğunu, hangi açıklıklardan kaynaklandığını, bu zafiyetlerin nasıl exploit edilebileceğini ve bu zafiyetlerin nasıl önlenebileceği ile ilgili ayrıntılı dokümanlar da mevcuttur. OWASP, birçok firma ve web uygulama sızma testleri ile ilgili çalışmalar yapan kişilerden bilgiler toplayarak ve bu bilgileri analiz ederek o yıla ait en riskli 10 güvenlik zafiyetinin istatistiğini çıkartmakta ve ücretsiz olarak sunmaktadır. OWASP topluluğunun her üç ya da dört senede bir yenilediği OWASP Top 10 ismiyle o yıla ait en riskli güvenlik zafiyetlerini listeledikleri Top 10 listesi hazırlamaktadır. OWASP İlk 10, en yaygın 10 uygulama güvenlik açığının listesidir. Ayrıca risklerini, etkilerini ve karşı önlemlerini de gösterir. Her üç ila dört yılda bir güncellenen en son OWASP güvenlik açıkları listesi 2018'de yayınlandı.



Yukarıda yer alan görüntüde Kali Linux içerisinde çalıştırılmış olan Owasp ile hedef siteye zafiyet taraması gerçekleştirilmekte.

Yapılan tarama işleminin ardından sol alt kısımda görüldüğü üzere hedef sitede bulunan zafiyetler listelenmiş durumda. Her bir zafiyetin üzerine tıklayarak zafiyet hakkında bilgi edinmek mümkün.



Görüntüden de anlaşılacağı gibi, listede yer alan bir hatanın üzerine çift tıklayarak böyle bir bilgilendirme ekranı sunuluyor ve burada başta hatanın risk düzeyi olmak üzere hata hakkında ve hatanın çözümü hakkında da bir açıklama bulmak mümkün.

Tespit Edilen Bazı Zafiyetler Hakkında Bilgiler

DOM tabanlı XSS saldırısı nedir?

Belge Nesne Modeli (DOM), HTML belgelerindeki nesneleri temsil etmek ve işlemek için kullanılan bir kuraldır. Tüm HTML belgelerinin, sayfayı oluşturan web tarayıcılarına belge özelliklerine çeviren nesnelerden oluşan ilişkili bir DOM'si vardır. İstemci tarafı komut dosyası yürütüldüğünde, HTML sayfasının farklı özelliklerine erişmek ve sayfayı düzgün bir şekilde oluşturmak için değerlerini değiştirmek için HTML sayfasının DOM'sini kullanır.

DOM tabanlı XSS saldırıları, yansıyan XSS saldırılarıyla aynı şekilde çalışır, çünkü kullanıcının kendisi muhtemelen saldırgandan aldığı bir bağlantıya tıklayarak kötü amaçlı komut dosyasını tetikler. Saldırgan, tarayıcının DOM özelliklerine erişirken yürüteceği kötü amaçlı bir komut dosyası sunan bir URL oluşturmak için DOM güvenlik açıklarından yararlanır.

DOM tabanlı bir XSS saldırısı ile yansıyan bir XSS saldırısı arasındaki en büyük fark, DOM tabanlı saldırıların tamamen istemci tarafında gerçekleşmesidir. Saldırgan, sunucuya herhangi bir kötü amaçlı komut dosyası göndermez. Bu, giriş çerezlerini çalmaya ve hesapları ele geçirmeye yönelik yaygın bir saldırdır.

Gerçek dünyadan XSS saldırı örnekleri

- 2018 yılında, Birleşik Krallık havayolu şirketi [British Airways](#) , 380.000 rezervasyon işlemini etkileyen bir veri ihlalinin kurbanı oldu. Saldırganlar, sitenin JavaScript'inden yararlanmak ve müşteri verilerini kontrol ettikleri bir sunucuya geri göndermek için siteler arası komut dosyası çalıştırmayı kullandılar.
- 2019'da popüler çevrimiçi video oyunu [Fortnite](#) , saldırıların 200 milyona kadar kullanıcının hesaplarına erişmesine izin veren bir veri ihlali yaşadı. Bu, Fortnite sunucularındaki güvenli olmayan tek bir web sayfasındaki bir XSS güvenlik açığından yararlanan siteler arası bir komut dosyası saldırısıydı. Bu krallığın kapılarını açmak için yeterliydi.
- Aralık 2015 ile Ocak 2016 arasında, çevrimiçi pazar yeri devi [eBay](#) , web sitesinin sunucularında bilinen bir XSS güvenlik açığına sahipti. Saldırganlar, eBay'in bazı listelerine kötü amaçlı JavaScript enjekte edebildi. Bir kullanıcı bu listelerden birine tıkladığında, kişisel bilgilerinin toplandığı meşru görünen ancak sahte bir eBay sayfasına götürüldü. eBay'i düzeltmeye motive etmek için BBC tarafından güvenlik açığı hakkında bir rapor aldı.

XSS saldırıları nasıl önlenir

Siteler arası komut dosyası çalıştırma saldırılarını önlemenin yolu, XSS güvenlik açığının türüne, kullanıcı girişi bağlamına ve programlama çerçevesine bağlıdır. Bununla birlikte, web uygulamanızı güvende tutmak için izlemeniz gereken bazı genel sağduyu yönergeleri vardır.

1. Bir web güvenlik açığı tarayıcısı kullanın ve düzenli taramalar yapın

Kişisel bir bilgisayarın bir virüs tarayıcıya (antivirüs) nasıl ihtiyaç duyduğuna benzer şekilde, web uygulamanızın veya web sitenizin bir web güvenlik açığı tarayıcısına ihtiyacı vardır. Farklı satıcılardan birçok farklı ürün var. Bir web güvenlik açığı tarayıcısı, web uygulamanızı/web sitenizi düzenli olarak tarar ve bulduğu sorunlar hakkında sizi uyarır.

2. İnşa ederken farkında olun

Web uygulamasını/sitesini oluşturmaya dahil olan herkesin XSS güvenlik açıklarının sonuçlarının farkında olması ve kod yazarken bunları akılda tutması gerekir. Geliştiricilerden KG'ye kadar kuruluşunuzdaki her ilgili grubun bu konularda yeterli eğitimi aldığından emin olun.

3. Kullanıcı girdisine güvenmeyin

Doğru. Hiçbir kullanıcı girdisine asla güvenmeyin. Kullanıcı girdisi içeren herhangi bir HTML çıktısı, bir XSS saldırısı riskini beraberinde getirir. Ve ne kadar çok olursa, risk o kadar büyük olur. Kimliği doğrulanmış kullanıcılardan, dahili kullanıcılardan ve genel kullanıcılardan gelen girdilere aynı şekilde davranın: ona güvenmeyin.

4. HttpOnly bayrağını ayarlayın

Tanımlama bilgileri için HttpOnly bayrağının ayarlanması, olası XSS güvenlik açıklarının azaltılmasına yardımcı olacaktır. HttpOnly bayrağının ayarlanması, tanımlama bilgilerinin istemci tarafı JavaScript

aracılığıyla (yukarıdaki tanımlama bilgisi çalma örneğimizde olduğu gibi) yalnızca HTTP aracılığıyla erişilebilir olmasını sağlar.

5. Kaçış/kodlama kullanın

Kullanıcı girişinin gerçekleştiği yere göre uygun bir çıkış/kodlama tekniği kullanın: HTML çıkışı, JavaScript çıkışı, URL çıkışı, vb. Ayrıca, kaçmak için kendi kitaplıklarınızı yazmak yerine mevcut kitaplıkları kullanmak daha güvenlidir.

6. HTML'nizi dezenfekte edin

Geçerli etiketleri kırmadan HTML içeren kullanıcı girişinden kaçamaz/kodlayamazsınız. Geçici bir çözüm olarak, HTML'yi ayrıştırmak ve temizlemek için güvenilir ve doğrulanmış bir kitaplık kullanın. Geliştirme diliniz için uygun kitaplığı seçtiğinizden emin olun.

7. İçerik Güvenliği Politikasını Etkinleştirin

XSS güvenlik açıklarının sonuçlarını daha da azaltmak için İçerik Güvenliği Politikası'nı (CSP) etkinleştirin. CSP, isteğin kaynağına göre hangi dinamik kaynakların yüklenebileceğini belirlemenizi sağlayan bir HTTP yanıt başlığıdır.

Siteler Arası İstek Sahteciliği CSRF (Cross Site Request Forgery) Nedir?

CSRF (Cross Site Request Forgery) genel yapı olarak bir web sitesinin açığından faydalanarak site kullanıcılarının istekleri dışında sanki o kullanıcıymış gibi erişerek işlem yapılması sürecini içerir. Genellikle GET requestleri ve SESSION işlemlerinin doğru kontrol edilememesi durumlarındaki açıklardan saldırganların faydalanmasını sağlamaktadır.

CSRF zafiyeti OWASP Top 10 listesinde yer alan ve en sık karşılaşılan çevrimiçi saldırılardan biridir. Bu zafiyet sıkça kullanılan popüler web uygulamalarında dahi görülmektedir.

CSRF Nasıl Gerçekleşir ?

CSRF saldırısı, daha önce kimliği doğrulanmış başka bir web sitesi aracılığıyla bir web uygulamasına istek gönderen kötü amaçlı bir bağlantı içerir. Elde edilen kimlik bilgileriyle mağdur kimliğine bürünülür ve kötü amaçlı faaliyetlerde kimlik doğrulama bilgisi atlanılmış olur. Örneğin, bankacılık sistemine giriş sayfası tarayıcıda açık bulunduğu bir durumda , mail adresine gelen tehlikeli bir bağlantı tıklanarak saldırgana kullanıcı bilgileri verilmiş olur. Saldırgan bu bilgilerle bankacılık sistemine girip para transferi gerçekleştirebilir.

Bu tür saldırılar genellikle bankacılık, sosyal medya ve ağ cihazları için kullanılan web arayüzlerine karşı gerçekleştirilir.

CSRF Zafiyetinde Alınabilecek Önlemler

Sistem Tarafı Alınabilecek Önlemler

- **Token Kullanımı**

Kullanıcıya her oturum için random ve benzersiz “token” bilgisi verilir.

- **Get Metodu Yerine Post Metodu Kullanımı**

Kullanıcıdan alınan önemli veriler POST metodu ile alınmalıdır.

- **CAPTCHA Kullanımı**

Bir web formunda captcha (Completely Automated Public Turing test to tell Computers and Humans Apart) bilgisi doğru girilmediği sürece işlem gerçekleştirilemeyeceği için “CSRF” saldırısına karşı alınacak bir önlem niteliğindedir.

Kullanıcı Tarafı Alınacak Önlemler

- Web uygulama verileri ve cookie bilgileri düzenli aralıklarla temizlenmelidir.
- Kişisel bilgilerin bulunduğu web uygulamalarının oturum bilgileri bilgisayarda kayıtlı tutulmamalıdır.
- Kaynağı belirsiz mail ve bağlantılara dikkat edilmelidir.

İçerik Güvenliği Politikası (CSP-Content Security Policy)

CSP, web uygulamalarımızı başta XSS olmak üzere, bir dizi güvenlik zafiyetine karşı korumak için ek bir güvenlik katmanı sunmaktadır. Elbette cürmü, CSP'nin desteklendiği browser'lar kadardır. Dolayısı ile mevzu bahis zafiyetler için (örneğin, XSS) tek başına yeterli olmayacak; CSP'nin desteklenmediği bir browserda, sitenizdeki zafiyet yine istismar edilebilecektir. CSP bir Derinliğine Savunma (defense-in-depth) olarak değerlendirilip, zafiyetin giderilmesi konusunda gerekli işlemler mutlaka yapılmalıdır.

Clickjacking Nedir ?

Clickjacking, kullanıcının görünmeyen veya başka bir öge olarak gizlenmiş bir web sayfası ögesini tıklaması için kandıran bir saldırıdır. Bu, kullanıcıların farkında olmadan kötü amaçlı yazılım indirmesine, kötü amaçlı web sayfalarını ziyaret etmesine, kimlik bilgileri veya hassas bilgiler sağlamasına, para transfer etmesine veya çevrimiçi ürün satın almasına neden olabilir.

Tipik olarak, tıklama hırsızlığı, kullanıcının gördüğü sayfanın üstünde, bir iframe içinde, görünmez bir sayfa veya HTML ögesi görüntülenerek gerçekleştirilir. Kullanıcı, görünen sayfayı tıkladığına inanır, ancak aslında, onun üzerine aktarılan ek sayfada görünmez bir öğeye tıklamaktadır. Görünmez sayfa, kötü amaçlı bir sayfa veya kullanıcının ziyaret etmeyi düşünmediği meşru bir sayfa olabilir - örneğin, kullanıcının bankacılık sitesinde para transferine izin veren bir sayfa.

Clickjacking saldırısının çeşitli varyasyonları vardır, örneğin:

- Likejacking – Facebook "Beğen" düğmesinin manipüle edildiği ve kullanıcıların gerçekten sevmedikleri bir sayfayı "beğenmesine" neden olan bir teknik.
- Cursorjacking - kullanıcının algıladığı konum için imleci başka bir konuma değiştiren bir kullanıcı arayüzü düzeltme tekniği. Cursorjacking, Flash ve Firefox tarayıcısındaki şu anda düzeltilmiş olan güvenlik açıklarına dayanır.

Server Leaks Information on via “X Powered By” Http Response Header Fields

Web/uygulama sunucusu, bir veya daha fazla "X-Powered-By" HTTP yanıt başlığı aracılığıyla bilgi sızdırıyor. Bu tür bilgilere erişim, saldırganların

web uygulamanızın bağımlı olduđu diğerk çerçevesleri/bileşenleri ve bu tür bileşenlerin maruz kalabileceğı güvenlik açıklarını tanımlamasını kolaylaştırabilir.