Hardening

Applying best-practises, securing default configurations.

- Fail2Ban - securing services and web applications

Fail2Ban - securing services and web applications

Determine, which firewall is running

```
systemctl is-active ufw
ufw status

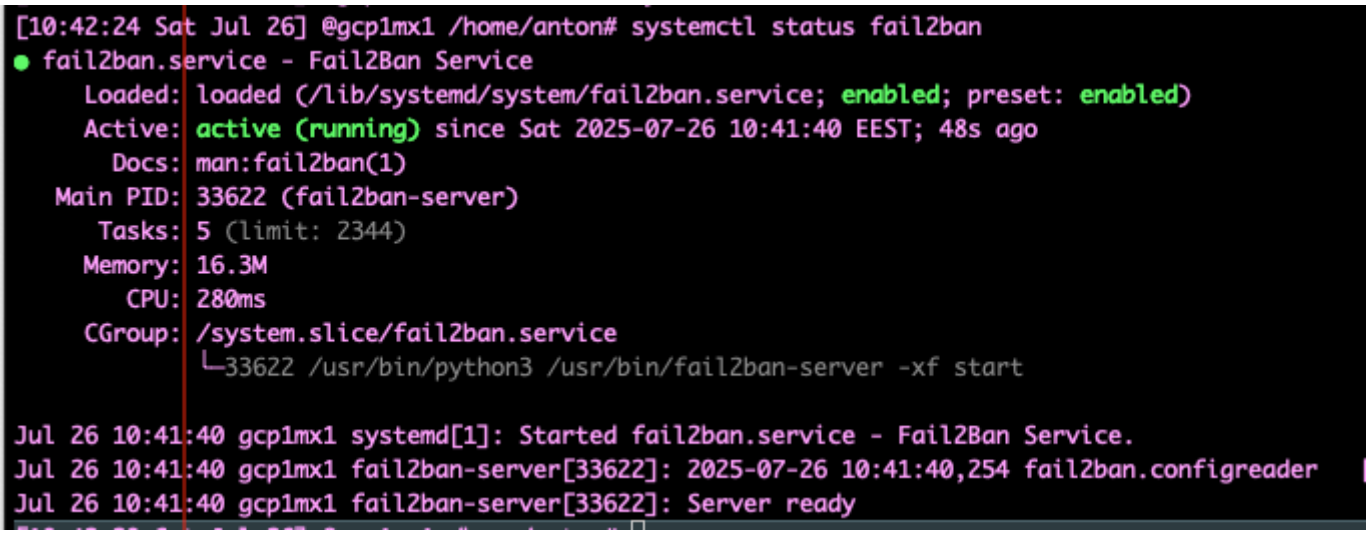
systemctl is-active nftables
nft list ruleset
```

iptables outputs, when active

```
iptables -L
```

Install, enable and start

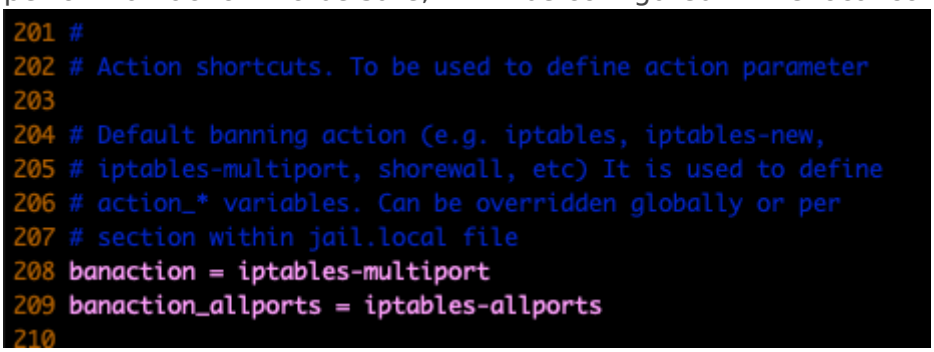
```
apt install fail2ban
systemctl enable fail2ban
systemctl start fail2ban
systemctl status fail2ban
```



Observe, take a coffee and understand the config. But do not make changes, as it will be overwritten on the update.

```
vi /etc/fail2ban/jail.conf
```

Specially, which method will be applied for the action. By default, iptables will be invoked to perform an action. To be sure, it will be configured in the local config file as well.



Instead, create a local copy of config (which will dominate on default config)

```
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Edit the local config

```
vi /etc/fail2ban/jail.local
```

```
[DEFAULT]
bantime = 1h
findtime = 10m
maxretry = 3
backend = systemd
destemail = to-xx@host
sender = from-fail2ban@host
chain = INPUT
banaction = iptables-multiport
action = %(action_mwl)s


[nginx-http-auth]
enabled = true

[nginx-botsearch]
enabled = true

[nginx-limit-req]
enabled = true
```

```
[nginx-bad-request]
enabled = true

[php-url-fopen]
enabled = true

[courier-smtp]
enabled = true

[postfix]
enabled = true

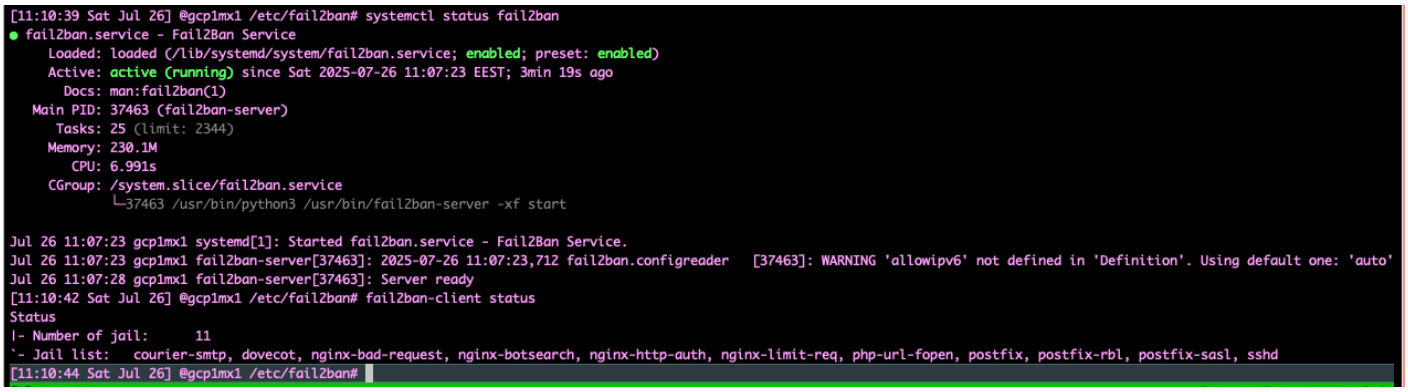
[postfix-rbl]
enabled = true

[dovecot]
enabled = true

[postfix-sasl]
enabled = true
```

Restart the service

```
systemctl restart fail2ban
systemctl status fail2ban
fail2ban-client status
```



To enable jail for nginx webserve

Requires modification to webserver configuration as described in

```
https://nginx.org/en/docs/http/ngx_http_limit_req_module.html
```

Create addition config file, which will be included by main configuration.

```
vi /etc/nginx/conf.d/ngx_http_limit_req_module.conf
```

```
limit_req_zone $binary_remote_addr zone=one:10m rate=50r/s;

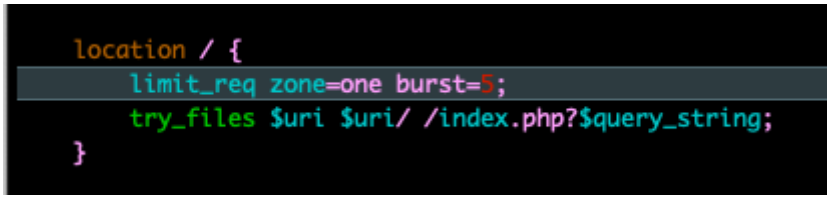
# for troubleshooting, enable limit_log logformat for access_log

limit_req_log_level info;
log_format limit_log '[$time_iso8601, $msec]: limit:$limit_req_status code:$status from:$remote_addr
to:$host$request_uri';
```

In addition to this, limitation need to be applied in every server config (per host):

```
vi /etc/nginx/sites-enabled/(host).conf
```

```
    location / {
        limit_req zone=one burst=50 nodelay;
    }
```



After webserver config changes, test and reload new config

```
nginx -t
nginx -s reload
```

To trigger 'nginx-botsearch'

```
for i in {1..60}; do curl -A "Trigger" http://dox.2dz.fi; done
```