

Forum: [Hack Forums](#)

Thread: [Achieve maximum Anonymity with NixOS](#)

Infinity. - May 28, 2025, 2:29 am

Hi, so in this thread we won't be going over what's nixos and how does it work, but why I am using it because

Declarative configs: Track all changes in configuration.nix.

Reproducibility: Rebuild your entire OS from scratch, imagine version control but for your OS.

keep in mind that to keep this setting unchanged Code boot.loader.systemd-boot.configurationLimit = 2;
to avoid cluttered rollback history (OPSEC risk).

or use init.d like a real man, what's systemd?

TOR EVERYTHING ?

```
Code services.tor = {  
  enable = true;  
  client.enable = true;  
  settings = {  
    SocksPort = "9050";  
    DNSPort = "5353";  
    AutomapHostsOnResolve = true;  
    TransPort = "9040";  
    VirtualAddrNetworkIPv4 = "10.192.0.0/10";  
    ExitPolicy = "reject *:*";  
  };  
};
```

then under the networking configuration specify the tor config and use iptables + transparent proxying to capture all leaks:

```
Code networking.proxy = {  
  default = "socks5://127.0.0.1:9050";  
  noProxy = "127.0.0.1,localhost";  
};  
Code networking.firewall.extraCommands = "  
  iptables -t nat -A OUTPUT -m owner ! --uid-owner tor -p tcp --syn -j REDIRECT --to-ports 9040  
  iptables -t nat -A OUTPUT -m owner ! --uid-owner tor -p udp -j DROP  
";
```

NOTE: for all your CLI apps you might need to use, use torsocks or even proxychains-ng

Code proxychains-ng nmap -sT -Pn example.com

torsocks curl ifconfig.me
Qubes + Whonix Integration ?

This assumes you are:-
Running NixOS inside AppVM in Qubes.

Routing through sys-whonix (default Tor gateway VM).

Ensure /etc/qubes-rpc/policy properly restricts inter-VM communications.

How the network flows:-

sys-whonix --> Tor gateway

nixos-whonix --> Isolated NixOS VM

Disposable VMs --> For risky tasks

? Pro tip:

Avoid GUI tools entirely in the NixOS AppVM. Use CLI-only environments.

Use disposables for sensitive sessions: e.g., `qvm-run --dispvm`.

Code `qvm-run --dispvm=nixos-dvm xterm`

NO LOGS ?

```
Code services.journald.extraConfig = "  
  Storage=volatile  
  Compress=no  
  ForwardToSyslog=no  
  ForwardToConsole=no  
  MaxRetentionSec=0  
";  
Code fileSystems."/tmp" = {  
  fsType = "tmpfs";  
  options = [ "nosuid" "nodev" "noexec" "size=1G" ];  
};
```

? Pro tip:

`services.syslogd.enable = false;`
You can also mount /home and /var/log as tmpfs , and use auditd only to monitor integrity, not log.

```
Code fileSystems."/var/log" = { fsType = "tmpfs"; options = [ "noatime" ]; };  
Code security.auditd.enable = true;  
security.audit.rules = [  
  "-a exit,always -F arch=b64 -S execve"  
];
```

enforce MAC ?

```
Code security.apparmor = {  
    enable = true;  
    policies = "strict";  
};
```

? Pro tip:

Write per-app AppArmor profiles for Firefox, GPG, and SSH.

E.g. profile for firefox

```
Code /usr/bin/firefox {  
    network inet stream,  
    /etc/tor/** r,  
    deny /proc/*/mem r,  
}
```

Encrypted Swap + No Hibernation ?

```
Code swapDevices = [{  
    device = "/swapfile";  
    randomEncryption.enable = true;  
}];  
powerManagement.enable = false;
```

```
boot.kernelParams = [ "nohibernate" ];
```

? Pro tip:

Consider full LUKS + detached header setup.

Kernel Hardening ?

```
Code boot.kernelParams = [  
    "slab_nomerge"  
    "init_on_alloc=1"  
    "init_on_free=1"  
    "page_alloc.shuffle=1"  
    "pti=on"  
    "vsyscall=none"  
    "kptr_restrict=2"  
    "dmesg_restrict=1"  
    "lsm=lockdown,yama,apparmor"  
];
```

Clarifying what the settings are mentioned above are beyond the scope of this tutorial.

Network isolation

```
Code programs.firejail.enable = true;  
hardware.bluetooth.enable = false;  
networking.wireless.enable = false;  
even blacklist the drivers  
Code boot.blacklistedKernelModules = [ "bluetooth" "iwlwifi" ];  
Just fucking isolate everything
```

```
Code programs.firejail = {  
    enable = true;  
    wrappedBinaries = {  
        firefox = "${pkgs.firefox}/bin/firefox";
```

```
chromium = "${pkgs.chromium}/bin/chromium";  
};  
};
```

? Pro tip:

Use USB tethering via burner phone for network (with USBGuard).

```
Code services.usbguard = {  
  enable = true;  
  rules = "  
    allow with-interface equals { 08:*:* }  
  ",  
};
```

Verification Checklist

Code `curl --socks5 127.0.0.1:9050 ifconfig.me` (Should show Tor exit)

Code `dmesg | grep -i blocked` (Verify module blacklisting)

Code `aa-status` (Check AppArmor enforcement)

Code `journalctl --disk-usage` (Should show <1MB usage)

Randomize Hostnames Code `networking.hostName = builtins.replaceStrings ["%RANDOM%"] ["${builtins.hashString "sha256" (toString builtins.currentTime)}"] "anon-%RANDOM%";`

Randomize MAC: Code `networking.networkmanager.wifi.macAddress = "random";`

Disable kernel namespaces Code `boot.kernel.sysctl."kernel.unprivileged_usersns_clone" = false;`

Use rage (age + SSH) to encrypt secrets with Tor hidden service keys.

Consider Heads or PureBoot for tamper-evident boot security.

Set up remote log watching via Tor hidden SSH forward, not over clearnet.

Avoid browser fingerprinting entirely by using text-only Tor Browser (e.g. torify lynx or w3m) or curl with user-agent spoofing.

Final notes: The chances of getting caught are very minimal for the above, but if you make 1 simple mistake, all the hard work is done for.

GI:

Pro Tip: Use your neighbor's wifi.

Let me know in the thread if you want an advanced level tutorial for anonymity

HerrWinston - May 31, 2025, 2:25 am

Holy shit this thread is good, well done man i just gotta understand first how to use everything.

Infinity. - May 31, 2025, 8:19 pm

(May 31, 2025 02:25 AM)HerrWinston Wrote: [\[CITE\]](#) Holy shit this thread is good, well done man i just gotta understand first how to use everything.

Thanks man, let me know if you have any doubts. I am looking forward to creating a LUKS encryption thread as well, just trying to find some time.

TheMightyKnighty - *September 14, 2025, 11:27 am*

Thanks a lot man, this was very useful! :)