ASSURED

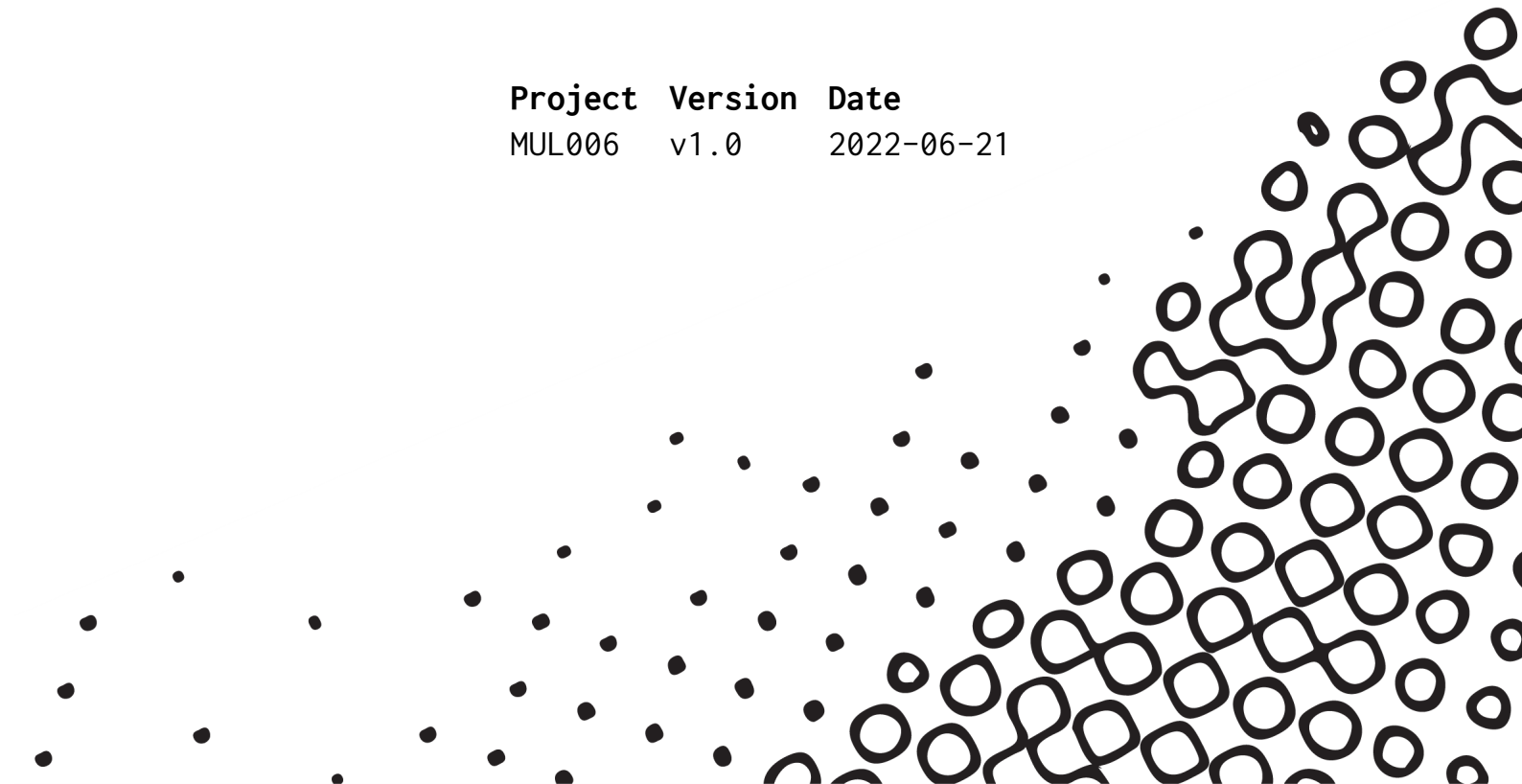
SECURITY CONSULTANTS

Report

Mullvad VPN relay audit

Wictor Olsson, Johanna Abrahamsson, Albin Eldstål-Ahrens

Project	Version	Date
MUL006	v1.0	2022-06-21



ASSURED
SECURITY CONSULTANTS

REPORT

Project	Version	Date
MUL006	v1.0	2022-06-21

Executive summary

Between 2022-04-25 and 2022-05-13 Assured Security Consultants performed a security audit on behalf of Mullvad.

Three different deployments of the Mullvad VPN relay servers were in scope, two of these used WireGuard and one OpenVPN.

This report is listing the security issues found, along with recommendations for fixing or mitigating them. In our conclusions we discuss the issues and address apparent patterns in areas where security is lacking.

Several findings of category Note reflect positive practices.

Issues were found with the following risk severity assessments (number of issues):

Critical 0 | High 0 | Medium 11 | Low 9 | Note 5

Some common issues were identified regarding best practices and hardening, there were issues identified in the categories of network access control, authentication, credentials handling, service configuration and system hardening. Externally the deployments have a quite strong security posture but internally there are some issues to be resolved. Our recommendations are to initially focus on improving access control to limit the attack surface, review and improve configuration of services, patch level and hardening as well as review and improve the deployment process with regards to credentials.

Assured would like to thank the Mullvad team for their support during this security audit. We are happy to answer any questions and provide further advice.

ASSURED
SECURITY CONSULTANTS

REPORT

Project	Version	Date
MUL006	v1.0	2022-06-21

Contents

ASSURED
SECURITY CONSULTANTS

REPORT

Project	Version	Date
MUL006	v1.0	2022-06-21

1 Introduction

1.1 Background

Assured AB (Assured) was contracted by Mullvad to perform a security assessment of their VPN relay servers.

1.2 Constraints and disclaimer

This report contains a summary of the findings found during the project period. This report should not be considered a complete list of all possible vulnerabilities, security flaws and/or misconfigurations.

1.3 Project period and staffing

Assured started the project on 2022-04-25 and finished on 2022-05-13.

This report was last reviewed on 2022-06-21.

Involved in the penetration testing were Assured consultants Wictor Olsson, Johanna Abrahamsson and Albin Eldstål-Ahrens.

1.4 Risk rating

In this report we have assessed the severity of issues and identified vulnerabilities. The levels of severity are rated according to the OWASP Risk Rating Methodology [1].

Table 1: OWASP Risk Rating overall severity model

Overall risk severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

As Table 1 visualizes, the overall risk assessment is determined from a combined likelihood and impact of an identified vulnerability or security issue. A value from 0 to 9 is assessed for each variable, where 0-2 is determined LOW, 3-5 is MEDIUM and 6-9 is HIGH.

Likelihood is dependant on attributes related to threat actors and the identified vulnerability, with factors such as: the skill level and motivations of the threat agents; how easily the vulnerability can be found and exploited, and; how likely an exploit may be detected.

Impact depends on technical and business factors, such as: level of loss of confidentiality, integrity, availability and accountability; potential financial damage; potential brand damage, and; potential violations of privacy.

Please note that the severity assessment is made by Assured consultants and ratings may differ from the resource owners' ratings.

2 Scope and methodology

2.1 Scope

2.1.1 Security assessment of VPN relays

Assured consultants were tasked with performing a security assessment of the Mullvad VPN relay servers. Three different server deployments were part of this audit, two of which run WireGuard and one which runs OpenVPN.

The main areas of interest were to ensure the following properties:

- No logs or information leakage exposing the clients using the server
- No known vulnerabilities in the exposed services
- Service and system configuration should follow best security practices

For the remainder of this report, the term *customer logging* is used to mean logging of customer data, and the term *system logging* is used to refer to logging of data which is strictly unrelated to customer activity.

2.2 Methodology

2.2.1 System audit

Assured consultants were given remote administrative access to the target servers. Manual and automated analyses were performed with the aid of several tools and scripts.

The servers were running services such as WireGuard and OpenVPN to handle the tunneling, bind9 for DNS and other services for socks5 proxying, system management and monitoring.

The following positions of attack were evaluated during the audit:

- External
- Authenticated Mullvad VPN client (OpenVPN and Wireguard)

The scenario of assumed compromise was also investigated where analysis was performed with regards to privilege escalation from different contexts within the system.

Traffic forwarding tests were performed by peering with the relays and attempting communication with various interfaces and networks to verify certain access control scenarios.

Limited traffic corruption tests (fuzzing) were executed using a lab setup of certain services of interest, with a debugging configuration as similar to the target system as possible.

The following system properties were audited:

- Running processes, services and scheduled jobs
- Patch level of exposed services
- Configuration of exposed services
- Firewall rule-set
- User accounts and groups
- Administrative groups and privileges
- Privileges of running services
- Hardening of kernel and running binaries
- Services and system log collection and erasure
- Common privilege escalation vectors

2.3 Limitations

The time allotted to the test was limited. The bulk of auditing was focused on networking, service and system configuration, as well as Mullvad-specific customization scripts. Audits of third-party software packages were only carried out to a limited extent.

3 Observations

These are the observations made during the security assessment. The observations are split and grouped into a few categories based on the deployment types as there are similarities between deployments.

3.1 Common to multiple deployments

The following observations apply to more than one server deployment. If nothing else is specified in a finding, it is applicable to all three server types.

3.1.1 Medium User-writable scripts run by root

Likelihood: LOW (1), Impact: HIGH (7)

During analysis of scripts and similar that run on the system, several scripts were identified to be running as root. These scripts were also owned by other system users as well as a user account for a service.

This results in a potential privilege escalation vector which could allow an attacker with access to the `promtail` service account to obtain root access. An unnamed (redacted) administrator account is also affected. Due to the required access conditions the risk rating is lowered.

Example 1 lists scripts writable by regular users but executed by root via cron jobs.

Example 1: User-writable scripts run by root

```
Several scripts under the path(see crontab): /home/redacted/*
/usr/share/promtail/update_promtail_*.sh
```

We recommend that scripts be owned by the executing user account and have restricted write privileges. This is of particular importance for scripts run by root.

3.1.2 Medium Permissive firewall policy

Likelihood: MEDIUM (4), Impact: MEDIUM (5)

The `INPUT` iptables chain (for both IPv4 and IPv6) has a default policy of `ACCEPT` and no final wildcard `DROP` rule. As a result, only rate limiting and specific blocking is performed for local services. This could allow an attacker to access services or interfaces which were not intended.

Example 2: iptables INPUT chains for IPv4 and IPv6

```
Chain INPUT (policy ACCEPT)
target    prot opt source            destination
RATE-LIMIT udp -- 10.0.0.0/8         0.0.0.0/0          udp dpt:53
...
RATE-LIMIT tcp -- 100.64.0.0/24      0.0.0.0/0          tcp dpt:53
ACCEPT    udp  -- 0.0.0.0/0          0.0.0.0/0          udp dpt:52000
SNMP      udp  -- 0.0.0.0/0          0.0.0.0/0          udp dpt:161
DROP      tcp  -- 0.0.0.0/0          0.0.0.0/0          multiport dports 1022



Chain INPUT (policy ACCEPT)
target    prot opt source            destination
RATE-LIMIT udp    ::1                ::/0               udp dpt:53
...
```

The IPv6 rules allow all incoming traffic, and only apply rate-limiting.

We recommend to implement white listing instead and that the default policy on all chains be changed to `DROP` and more detailed rules added to allow specific traffic by port or source.

3.1.3 Medium Possible to access the internal interface

Likelihood: MEDIUM (4), Impact: MEDIUM (4)

Due to the current internal firewall rules it is possible for an authenticated Mullvad user/peer to reach the local network interface used for the Mullvad internal WireGuard network, this could potentially expose listening services intended only for the internal network to an attacker or malicious user. However the peer is not allowed to forward traffic through the interface to reach the internal network, which is good.

We recommend to apply firewall rules to limit the peers/clients from accessing services potentially listening on this interface.

3.1.4 Medium Shared SNMP credentials

Likelihood: MEDIUM (4), Impact: MEDIUM (4)

The SNMP service running on all deployments have the same user and credentials deployed. If an attacker compromises the user credentials it will be possible to access the SNMP interfaces of all the relay servers from the right context.

We recommend that each deployed machine receive its own unique credentials for inbound services, to enable revocation in case of a detected compromise.

3.1.5 Medium Unauthenticated Grafana Loki

Likelihood: MEDIUM (5), Impact: MEDIUM (5)

There is a promtail service, acting as a client, which reports performance metrics to a centralized internal system, this service is deployed on all of the relays.

Based on the configuration of this service there is no authentication towards the Grafana Loki backend, leaving it open for abuse.

To access this backend service the attacker needs to compromise one of the relays that are peered to the internal WireGuard network or get access to the same network through other means. The forwarding rules in place on the relays inhibits any regular authenticated Mullvad user peer to access this internal network directly.

We recommend to configure authentication on the service to limit unauthorized access.

3.1.6 Medium Known vulnerabilities

Likelihood: MEDIUM (5), Impact: MEDIUM (5)

The audit indicates that the following known vulnerabilities are not patched.

- openssh-server-7.6p1-4ubuntu0.6: CVE-2021-41617 (no fix for bionic)
- openvpn-mullvad-2.5.0: CVE-2022-0547 (unknown patch status.)
- rsyslog-8.32.0-1ubuntu4: CVE-2019-17041 (no fix for bionic)
- rsyslog-8.32.0-1ubuntu4: CVE-2019-17042 (no fix for bionic)

All issues except CVE-2022-0547 required either an already very privileged position to abuse or specific requirements/configuration which are not applicable to the deployments.

CVE-2022-0547 requires the OpenVPN instance to have deferred authentication configured. A plugin with custom logic is indeed used for deferred authentication:

Example 3: Excerpt from OpenVPN server configuration

```
plugin /home/redacted/vpnserver/libvpnauth.so ”/home/redacted/vpnserver/vpnauth_auth.py /home/
    redacted/vpnserver/vpnauth_portforward.py”
```

CVE-2022-0547 was not fully triaged/confirmed during the audit, and since Mullvad is building the OpenVPN application inhouse we strongly recommend to investigate if the appropriate patch has been applied to the running OpenVPN application.

3.1.7 Medium Plaintext protocols in use

Likelihood: LOW (2), Impact: HIGH (6)

There are two services running on the relays used for reporting performance metrics to a internal server, promtail and telegraf. Both promtail (Grafana Loki) and telegraf (InfluxDB) are using plaintext HTTP to communicate to their corresponding database/API backends. These services communicate over an internal WireGuard tunnel with the corresponding backends but an attacker in the right position could potentially intercept clear-text traffic of these services.

```
From Telegraf configuration:
...
[[outputs.influxdb]]
  urls = [”http://XXXXX:8086”] # required
  database = ”mullvad” # required
  skip_database_creation = true
...

From Promtail configuration:
...
clients:
  – url: http://XXXXX/loki/api/v1/push
    external_labels:
...
```

We recommend to enable transport security (TLS) for these connections.

3.1.8 Medium Shared credentials for consumed services and APIs

Likelihood: LOW (2), Impact: HIGH (6)

When analyzing service configuration several services across deployments were found to be using the same credentials to communicate towards the Mullvad backend API. Based on the configuration, if an attacker gains access to one of the relays or services these credentials would have to be invalidated and recommissioned for all of the Mullvad relays.

The `telegraf` configurations on multiple machines share the same InfluxDB credentials.

The `wireguard-manager` configurations on multiple machines share the same API

credentials.

The `blocklist-service` configurations on multiple machines share the same API credentials.

Furthermore the `wireguard-manager` and `blocklist-service` share API credentials. The shared password comes from a non-random source, with visibly low entropy.

We recommend that each deployed machine receive its own unique credentials for outbound use, to enable revocation in case of a detected compromise. Credentials should be generated by a randomized source with sufficient entropy.

3.1.9 Low Unnecessary installed software and leftovers

There are some installed packages, such as tcpdump, netcat and nmap, on the server(s), which are not necessary for the functionality and also can be useful for an attacker who gets code execution on a server. There is also compilation software, such as gcc installed. For a hardened production server, it is considered best practice to remove this kind of software.

No Linux containers are configured, but the `lxcfs` daemon is still running. This applies to the WireGuard-stboot and OpenVPN deployments.

It is recommended to remove unnecessary software and residual configuration from the servers to minimize the attack surface.

3.1.10 Low Binaries lacking instrumented hardening

Likelihood: LOW (2), Impact: MEDIUM (3)

A few binaries/applications running on the target system lack certain automatic security mitigations. These mechanisms (RelRO, Stack canary, PIE, FORTIFY_SOURCE) will potentially protect and/or make it harder to exploit the application. They typically require no changes to the source code, and can be enabled by passing the relevant flags to the compiler at build time.

Example 4: Partial checksec.sh output, Mullvad binaries

```
RELRO          STACK CANARY    PIE        FORTIFY FILE
Partial RELRO  No canary found No PIE       No    /usr/local/bin/tcp2udp
No RELRO       No canary found No PIE       No    /usr/local/bin/blocklist-service
No RELRO       No canary found No PIE       No    /usr/local/bin/wireguard-manager
```

We recommend to review the build options of these application and enable suitable mechanisms.

A number of OS and third-party applications also lack binary hardening

options:

Example 5: Partial checksec.sh output, upstream packages

```
RELRO          STACK CANARY    PIE         FORTIFY FILE
No RELRO       No canary found No PIE        No    /usr/bin/telegraf
Partial RELRO  No canary found No PIE        No    /opt/promtail/installers/2.2.1/promtail-linux-
    amd64
Partial RELRO  Canary found    No PIE        Yes   /usr/bin/python3.6
Full RELRO     Canary found    PIE enabled   No    /sbin/lvmetad
Full RELRO     Canary found    PIE enabled   No    /usr/lib/policykit-1/polkitd
```

Our recommendation is to put pressure on the upstream package maintainers to implement these build options in their distribution pipelines.

3.1.11 Low Externally accessible WireGuard service

Likelihood: LOW (2), Impact: LOW (1)

UDP port 52000 hosts a WireGuard service, which listens on all IPv4 and IPv6 interfaces. The firewall opens this port explicitly, from all sources.

Example 6: iptables INPUT chain

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
...
ACCEPT     udp  -- 0.0.0.0/0            0.0.0.0/0           udp dpt:52000
...
```

This WireGuard interface appears to be for internal use and likely does not need to be accessible from everywhere.

We recommend reducing the range of acceptable source addresses in the firewall, to match the intended use of the service.

3.1.12 Note Exposed blocklist and wireguard-manager service

Wireguard-manager does not apply to the OpenVPN deployment.

The `blocklist-service` and `wireguard-manager` application listens on ephemeral IPv6 udp ports on all interfaces(intended for loopback communications), and connections are not restricted via iptables. This appears to be an internal service, which connects to the Mullvad API, and used for transmission only.

3.1.13 Low Service accounts with shells

Likelihood: LOW (2), Impact: MEDIUM (3)

A number of service-specific accounts are configured to have login shells.

Example 7: Excerpt from /etc/passwd

```
telegraf:x:601:601::/home/telegraf:/bin/sh
promtail:x:999:1010:System user for Grafana Promtail:/home/promtail:/bin/sh
monitor:x:998:998::/home/redacted/monitor:/bin/sh
dante:x:600:600::/home/dante:/bin/sh
```

It is recommended to set the shell of accounts like these to `/usr/sbin/nologin` or `/bin/false` to make unauthorized login more difficult.

3.1.14 Low AppArmor profiles

Likelihood: LOW (2), Impact: MEDIUM (3)

Out of the listening network services, only `named` is confined by AppArmor. Notably, the external-facing `openvpn`, `danted`, `tcp2udp` and `blocklist-service` as well as `wireguard-manager` are unconfined.

We recommend to review the output from the util `aa-unconfined` and consider adding/creating profiles to restrict these services by following the AppArmor tutorial from Canonical [2] and/or use aa-genprof to profile relevant applications during runtime.

3.1.15 Low Exposed BIND version

Likelihood: MEDIUM (3), Impact: LOW (2)

On the back-end DNS servers, the BIND version is set to a nonsense string, concealing the actual service version in use. The `named` configuration of the VPN relays lacks this option. This service is the client-facing DNS server for the compulsory DNS hijacking. As a result, the BIND version is always exposed to VPN clients.

Our recommendation is that the BIND version string be hidden consistently in all layers of the DNS hierarchy, to reduce the ability of an attacker to fingerprint services.

3.1.16 Note Administrators

Likelihood: LOW (0), Impact: LOW (0)

The administrative users that have access to the servers through SSH all essentially have root privileges on the servers.

Instead of giving all of the users full privileges one could create different administrative groups which have access to different services and resources to limit a potential attackers possibilities once foothold has been established on the server(s).

3.1.17 Note SSH access limited

The running OpenSSH server is protected in multiple ways:

- The firewall opens access only from specific IP addresses
- Password authentication is not permitted
- The daemon allows only specific user accounts to log in
- root login is not permitted

3.1.18 Note Service logs disabled

The following system services were audited, and found to have their system and customer logging disabled entirely:

- `danted`, a SOCKS proxy server,
- `named`, the BIND domain name server,
- `openvpn`, a VPN service,
- `wireguard`, a VPN service,
- `blocklist-service`, which adds automatic block rules in iptables,
- `wireguard-manager`, internal application

The SSH server has verbose system logging enabled, but does not come in contact with customer data. Only administrators from pre-authorized IP addresses are allowed to attempt login via SSH.

The Fail2Ban service has its log level set to `CRITICAL`, discarding all messages except service failures. The persistent database of failed attempts is disabled.

3.2 WireGuard relay servers

Since the two WireGuard server deployments included in the test, `se99-wireguard.mullvad.net` and `de999-wireguard.mullvad.net`, had very similar configuration this section covers both.

3.2.1 Low Kernel hardening options

Likelihood: LOW (2), Impact: MEDIUM (3)

This applies to the Wireguard stboot deployment.

The sysctl key `kernel.unprivileged_bpf_disabled` is set to 2 (temporarily disabled). Best practice is to set this to 1 (permanently disabled) unless BPF support is needed by unprivileged users.

The key `net.core.bpf_jit_harden` is set to 0, disabling certain BPF hardening mechanisms. We recommend setting this value to 2 (enabled for all users).

The key `kernel.kptr_restrict` is set to 1 (redact logged kernel pointers for most sources). We recommend setting this to 2 (redact logged kernel pointers from all sources).

3.2.2 Note Tcp2udp service

On the relay servers running WireGuard there is a service called `tcp2udp` that translates incoming `TCP` traffic to `UDP` and forwards it to the WireGuard service. This is used by customers that cannot connect through UDP, e.g. because of egress filtering beyond their control such as is common on public WiFi networks. This service seems to be developed in-house by Mullvad and the source is available at `https://github.com/mullvad/udp-over-tcp`.

Since we deemed the attack surface exposed by the service to be fairly low we only covered this service through brief static code analysis and very limited edge-case testing.

3.3 OpenVPN server

3.3.1 Medium Sensitive commands with sudo access

Likelihood: LOW (1), Impact: HIGH (8)

The `openvpn` daemon runs as user `nobody`. This user is granted rights to use `sudo` to run certain commands as `root`:

Example 8: Excerpt from sudoers

```
nobody ALL=(root) NOPASSWD:/sbin/ip
```

The `/sbin/ip` command, in combination with the kernel option `CONFIG_NET_NS`, allows for arbitrary command execution. This is a privilege escalation vector allowing any compromised service running as `nobody` to escalate to full `root` privileges.

We recommend constraining the arguments allowed for `/sbin/ip` to exclude the `ip netns` subcommand. In addition, these capabilities should be restricted to an account other than `nobody`, to prevent accidental sharing with other services in the future.

3.3.2 Medium Fail2Ban daemon running as root

Likelihood: MEDIUM (4), Impact: MEDIUM (5)

The `fail2ban` daemon is running with root privileges. According to its documentation, this application supports running as a non-root user, which is preferable. `fail2ban` handles data from external sources (log text), which makes it part of the external attack surface.

We recommend that all services be running as separate non-root users, with capabilities constrained to the minimum required for operation.

3.3.3 Medium Logging of invalid authentication attempts

Likelihood: MEDIUM (5), Impact: MEDIUM (5)

The authentication script configured for OpenVPN logs invalid VPN logins and passes them to `fail2ban`. If the login failure is due to an invalid user ID, the attempt is logged to disk. The log is cleared on an hourly basis via a cron job. The `findtime` for this log is set to 1 minute, i.e. users are temporarily banned if they fail too many attempts within one minute.

Example 9: Log entry for a user with an invalid account number

```
2022-04-29 07:05:16,764:INFO:Unauthorized attempt from 1.2.3.4
```

This logging only takes place for attempts with invalid account numbers, not for expired or valid accounts. As the log contains client information (the offending IP address), it should be cleared more frequently. If the log file is not necessary for the operation of fail2ban, it should be disabled entirely.

3.3.4 Low Debug setting would allow logging of customer information

Likelihood: LOW (1), Impact: MEDIUM (4)

The script `/home/redacted/monitor/monitor.py` queries all running OpenVPN instances for connected client information. The purpose of this is statistics collection (number of connected clients) and to terminate connections of clients with expired credentials. This is done by connecting to each OpenVPN instance over a special management socket and sending the `status 2` command. The response `CLIENT_LIST` is a comma-separated list of information about each connected client, including their public IP address, connection time, data statistics.

The monitor script has its log level set to `INFO` in the current configuration. At the more verbose `DEBUG` level, code is in place to log the response line in its entirety:

Example 10: Excerpt from monitor.py

```
101         for status_line in output.splitlines():
102             if status_line.startswith('CLIENT_LIST'):
103                 logging.debug('Read client line: %s', status_line)
```

We recommend removing the logging call entirely, to avoid accidental logging of sensitive customer information in the future.

3.3.5 Low Kernel hardening options

Likelihood: LOW (2), Impact: MEDIUM (3)

The `CONFIG_REFCOUNT_FULL` kernel option enables stricter checks on internal reference counters. This offers improved protection against memory corruption vulnerabilities, at the expense of a small performance decrease.

The `CONFIG_IO_STRICT_DEVMEM` kernel option restricts the ability of `root` to access `/dev/mem` in address ranges bound to kernel drivers. With this option disabled, an attacker with `root` privileges may be able to access sensitive information which has not been explicitly logged, by inspecting the memory of the kernel.

4 Conclusions and recommendations

Some common issues were identified regarding best practices and hardening.

There were issues identified in relation to network access control where firewall rules could be improved to limit the possibilities for an authenticated peer to access services or interfaces. External input rules for all of the deployments could also be stricter to limit current and future services from being exposed.

A service which communicates with an internal monitoring service lacked authentication as well as proper transport security and several shared credentials were found to be in use in various services.

Service and application configurations generally followed best practices. System hardening/minimization and sandboxing of external services could be improved to further minimize attackers possibilities.

A few running applications appear to be custom and built by Mullvad. These are hence not handled by the operating system maintainers, it is important to keep track of any vulnerabilities published in regards to these applications as well as using best practices when building and configuring them.

In regards to information leakage and logging of customer data the configuration is sound and did not display signs of any direct customer information.

In summary; externally the deployments have a quite strong posture but internally there are some issues to be resolved. Our recommendations are to initially focus on improving access control to limit the attack surface, review and improve configuration of services, patch level and hardening as well as review and improve the deployment process in regards to service credentials.

References

[1] OWASP, “OWASP Risk Rating Methodology.” https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology, 2019.

[2] Canonical, “Apparmor tutorial.” https://ubuntu.com/tutorials/beginning-apparmor-profile-development#1-overview, 2022.