Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Pentest-Report NordVPN Server & Infra 09.-10.2022

Cure53, Dr.-Ing. M. Heiderich, J. Larsson, MSc. D. Weißer, M. Elrod

Index

- Introduction
- Scope
- Identified Vulnerabilities
  - NV-03-002 WP1: *telegraf* user arbitrary file read via sudo abuse (Medium)
- Miscellaneous Issues
  - NV-03-001 WP1: Extraneous tool installation (Info)
  - NV-03-003 WP1: Local code execution and potential privilege escalation (High)
  - NV-03-004 WP1: Docker container escape via mounted *docker.sock* (Medium)
  - NV-03-005 WP1: Unnecessary sudo rules present on host systems (Info)
  - NV-03-006 WP1: File permissions facilitate potential privilege escalation (Info)
  - NV-03-007 WP1: Information disclosure via appropriated *Consul* token (Info)
  - NV-03-008 WP1: Critical components persist unpatched CVEs (Low)
  - NV-03-009 WP1: Hardening suggestions for Docker privilege escalation (Info)
  - NV-03-010 WP1: Hardening suggestions for Docker file systems (Info)
  - NV-03-011 WP1: Hardening suggestions for Docker users (Info)
- Conclusions

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Introduction

"We strive to make the internet better than it is today. It can be free from online threats, censorship, and surveillance, as envisioned in 1989 - the year the World Wide Web was invented."

From https://nordvpn.com/about-us/

This report - entitled NV-03 - details the scope, results, and conclusory summaries of a penetration test and source code audit against the NordVPN servers and infrastructure. The work was requested by Nord Security in May 2022 and initiated by Cure53 in September and October 2022, namely between CW38 and CW40. A total of twenty-five days were invested to reach the coverage expected for this project. All assessments for this audit were compiled into a single work package (WP), as follows:

- **WP1**: Test, audits, and assessments against NordVPN servers and infrastructure

In context, this security review represents the second proportion of a two-fold campaign requested by NordVPN. All information relating to the previous instance was documented under report NV-02. Cure53 was provided with sources, pertinent documentation, and any alternative means of access or information required to ensure a smooth test completion. For this purpose, the methodology chosen was white-box and a team of four senior testers was assigned to the project's preparation, execution, and finalization. All preparatory actions were completed in September 2022, namely in CW37, to ensure that testing could proceed without hindrance or delay.

Communications were facilitated via a dedicated, shared Slack channel deployed to combine the workspaces of Nord Security and Cure53, thereby creating an optimal collaborative working environment. All participatory personnel from both parties were invited to partake throughout the test preparations and discussions.

In light of this, communications proceeded smoothly on the whole. The scope was well-prepared and transparent, no noteworthy roadblocks were encountered throughout testing, and cross-team queries remained minimal as a result. The Nord Security team delivered excellent test preparation and assisted the Cure53 team in every respect to procure maximum coverage and depth levels for this exercise. Cure53 gave frequent status updates concerning the test and any related findings, whilst simultaneously offering prompt queries and receiving efficient, effective answers from the maintainers.

Live reporting was not requested, which in hindsight proved a sufficient decision considering the relatively low severity levels of the findings detected.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Regarding the findings, the Cure53 team achieved comprehensive coverage over the single scope item, detecting a total of eleven. Positively, only one of the findings was categorized as a security vulnerability, whilst the remaining ten were deemed general weaknesses with lower exploitation potential.

Generally speaking, the overall yield of findings documented in this report is relatively moderate, which represents a positive indication of the perceived security state of the NordVPN servers and infrastructure.

This impression is also corroborated by the fact that out of the eleven findings, only a single one was deemed a security vulnerability, whereas all other findings were considered miscellaneous in nature and should be trivially easy to address and mitigate.

All in all, following the completion of this audit and considering the significant findings unearthed during the previous round of testing, the Cure53 team is pleased to confirm that the NordVPN servers and infrastructure exhibit a relatively stable security foundation. The Nord Security should allocate enough time and resources toward addressing the findings identified in this report and elevating the security posture for the components in focus to a first-rate standard.

The report will now shed more light on the scope and testing setup as well as provide a comprehensive breakdown of the available materials. Subsequently, the report will list all findings identified in chronological order, starting with the detected vulnerabilities and followed by the general weaknesses unearthed. Each finding will be accompanied by a technical description and Proof of Concepts (PoCs) where applicable, plus any relevant mitigatory or preventative advice to action.

In summation, the report will finalize with a conclusion in which the Cure53 team will elaborate on the impressions gained toward the general security posture of the NordVPN servers and infrastructure, giving high-level hardening advice where applicable.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Scope

- **WP1: Test, audits, and assessments against NordVPN servers and infrastructure**
  - **Inspected servers:**
    - *The hosts in scope for the assessment were shared by accessing NordVPN's dashboard, whereby Cure53 was able to gain shell access to hosts.*
      - dkms-v1.0.20211208-nlx.3
      - radius-v1.0.3
  - **Key focus aspects:**
    - Assessing the security concepts and maturity of NordVPN's architecture. This was achieved by analyzing the configurations, services, and frameworks used to host and maintain the VPN Infrastructure.
    - Assessing the overall security concepts implemented on the provided VPN servers, use host and publish VPN access to NordVPN's users.
    - Assessing the security posture and configuration for the container workloads, associated services, and features used within to evaluate VPN hosts in scope for the engagement.
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Identified Vulnerabilities

The following section lists all vulnerabilities and implementation issues identified throughout the testing period. Please note that findings are listed in chronological order rather than by their degree of severity and impact. The aforementioned severity rank is given in brackets following the title heading for each vulnerability. Furthermore, each vulnerability is given a unique identifier (e.g., *NV-03-001*) to facilitate any future follow-up correspondence.

NV-03-002 WP1: *telegraf* user arbitrary file read via sudo abuse *(Medium)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

Whilst analyzing the NordVPN server's sudo configuration, the observation was made that the current implementation persists an overly permissive behavior. Specifically, one sudo rule grants the *telegraf* user full access to the *grep* binary without requiring a password. Said user can subsequently abuse this to read arbitrary files on the system holding *root*-user permissions.

Example affected host:
vm-us4377

Affected file:
/etc/sudoers.d/10-telegraf-_bin_grep

Affected code:

```
telegraf ALL=(ALL:ALL) NOPASSWD:/bin/grep
```

PoC:

```
root@vm-us4373:~# ls -alh /etc/shadow
-rw-r----- 1 root shadow 997 Sep 22 09:38 /etc/shadow

telegraf@vm-us4373:/etc/sudoers.d$ sudo grep -v '' /etc/shadow
root:
$6$dW.SgKfUhrGGQUi8$sHF.yp[REDACTED]ihvxuttKF62.fwN7XgCoXU89qzF1/3O3z5FbsgfFuAkA
DI.:19252:0:99999:7:::
[...]
```

To mitigate this issue, Cure53 advises integrating additional hardening improvements to the sudo permissions for all production hosts. All sudo rules should be specifically configured for a job and not allow generic access to a tool that can be abused to read arbitrary files, in this case, as *root*. This should be considered particularly important for sudo rules that set the *nopasswd* option.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Miscellaneous Issues

This section covers any and all noteworthy findings that did not lead to an exploit but might assist an attacker in successfully achieving malicious objectives in the future. Most of these results are vulnerable code snippets that did not provide an easy way to be called. Conclusively, while a vulnerability is present, an exploit might not always be possible.

NV-03-001 WP1: Extraneous tool installation *(Info)*

***Note from NordVPN**: After investigation it was decided to accept the risk since these tools are critical for server-side functionality.*

Whilst examining the deployed hosts and containers, the testing team noted usage of an extensive variety of tools. Some of these tools - including those integrated for network probing and development purposes - were deemed surplus to requirement in relation to key server operations and may prove useful for potential attackers to leverage.

Example Tools:

```
root@vm-us4376:~# which gcc nc curl wget
/usr/bin/gcc
/bin/nc
/usr/bin/curl
/usr/bin/wget
```

To mitigate this issue, Cure53 advises deploying servers with only the tools and software packages that are deemed business critical installed. This will ensure that the implementation cannot assist a potential attacker in their efforts to instigate network exploitation or application backdooring.

NV-03-003 WP1: Local code execution and potential privilege escalation *(High)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

Whilst validating the custom scripts present on NordVPN's servers, a local certificate parsing script was confirmed vulnerable to code execution. This script was also present in the sudo whitelisted for the *telegraf* user with the *nopasswd* option. The vulnerability is triggered when a malicious file is placed into a certain directory. Subsequently, either a periodic cron job or a user leveraging the sudo rule can initiate the vulnerable process. The code execution is triggered when the malicious file is parsed. Due to the *nopasswd* sudo rule, this vulnerability can be used to elevate privileges from the *telegraf* Linux user to *root* user.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Notably, this finding was categorized as a miscellaneous issue rather than a vulnerability, due to the fact that the *telegraf* user cannot write in any other directories that *checkcerts.py* searches in, which means that it cannot be triggered in the current implementation.

Affected file #1:
/usr/local/lib/telegraf/plugins/checkcerts.py

Affected code #1:

```
# vulnerable function #1
def check_cert(path, cert):
    [...]
    p = subprocess.Popen(
        'echo "%s" | openssl x509 -noout -subject -dates' % (cert,),
        shell=True,


# vulnerable function #2
def parsedate(date):
    p = subprocess.Popen(
        'date -d "%s" +%%s' % (date,),
        shell=True,
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
    )
```

Affected file #2:
/etc/sudoers.d/10-telegraf-_usr_local_lib_telegraf_plugins_checkcerts_py

Affected code #2:

```
telegraf ALL=(ALL:ALL) NOPASSWD:/usr/local/lib/telegraf/plugins/checkcerts.py
```

PoC:

```
telegraf@vm-us4374:~$ id
uid=999(telegraf) gid=999(telegraf) groups=999(telegraf)

telegraf@vm-us4374:~$ cat /etc/mysql/evil
-----BEGIN CERTIFICATE-----
(Your Primary SSL certificate: your_domain_name.crt)
`touch /tmp/PWNED`
-----END CERTIFICATE-----

telegraf@vm-us4374:~$ sudo /usr/local/lib/telegraf/plugins/checkcerts.py
cert_expiry,path=/etc/mysql/evil valid=False,valid_for_days=-
1,starttime=0,endtime=0,error=true
telegraf@vm-us4374:~$ ls -alh /tmp/PWNED
-rw-r--r-- 1 root root 0 Sep 22 11:02 /tmp/PWNED
```

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

To mitigate the code execution vulnerability, the developer team should block passing untrusted input to the dangerous *subprocess.Popen* function. Generally speaking, unsanitized data should never be passed to a function that creates a new process. Additional security against command injection vulnerabilities can also be achieved by leveraging the library's appropriate measures. For Python's *subprocess.Popen* specifically, safer shelling out can be implemented by setting *shell=False*.

NV-03-004 WP1: Docker container escape via mounted *docker.sock* *(Medium)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

During the analysis of the Docker setup, the observation was made that one can escape from the *NRPE* Docker container, which is facilitated by a Docker control socket mounted into it. As a result, a container compromise will simultaneously incur a full server compromise in addition.

Whilst some may argue that Docker containers are not primarily a security mechanism but also a deployment mechanism, deploying a container that can be broken out from in this way is still considered a negative practice. This scenario therefore represents a golden opportunity for server hardening and service isolation.

Configuration:

```
"Mounts": [
    {
        "Type": "bind",
        "Source": "/var/run/docker.sock",
        "Destination": "/var/run/docker.sock",
        "Mode": "ro",
        "RW": false,
        "Propagation": "rprivate"
    }
```

Note: While the Docker socket is mounted with the option *RO* (read only), this does not actually prevent the issuing of commands via this socket.

PoC:

```
root@vm-us4377:~# docker exec -it 0a347a809941 /bin/bash
root@us-us1:/# docker run -it -v /:/host -t alpine sh
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
213ec9aee27d: Pull complete
Digest: sha256:bc41182d7ef5ffc53a40b044e725193bc10142a1243f395ee852a8d9730fc2ad
Status: Downloaded newer image for alpine:latest
```

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

```
/ # chroot /host
root@04e9f30e930e:/# cat /etc/shadow
root:
$6$dW.SgKfUhrGGQUi8$sHF.ypZRbZBwUO5[REDACTED]AihvxuttKF62.fwN7XgCoXU89qzF1/3O3z5
FbsgfFuAkADI.:19252:0:99999:7:::
```

To mitigate this issue, Cure53 advises avoiding Docker socket mounting into containers unless deemed absolutely necessary. If this is the case, one can recommend integrating an ACL proxy as a filter to prevent harmful commands from being called. By correctly deploying these security measures, the deployed container technology will benefit from an additional layer of security and provide assistance against breaches by impeding the lateral movement of potential attackers.

NV-03-005 WP1: Unnecessary sudo rules present on host systems *(Info)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

Whilst validating and assessing the sudo mechanism present in NordVPN's servers, the testing team noted a number of nonessential sudo rules. This should not be considered a security vulnerability in isolation, yet could represent a potential attack vector should the binaries in question be installed and subsequently abused. Additionally, the presence of extraneous configuration files in such a security-sensitive area as sudo rules can convolute the view and distract from other more pressing issues.

Example file #1:
*/etc/sudoers.d/10-nagios-_usr_lib64_nagios_plugins_*

Example config #1:
```
nagios ALL=(ALL:ALL) NOPASSWD:/usr/lib64/nagios/plugins/
```

PoC #1:
```
root@vm-us4375:/etc/sudoers.d# ls -alh /usr/lib64/nagios/plugins/
ls: cannot access '/usr/lib64/nagios/plugins/': No such file or directory
```

Example file #2:
/etc/sudoers.d/10-telegraf-_usr_bin_wg

Example config #2:
```
telegraf ALL=(ALL:ALL) NOPASSWD:/usr/bin/wg
```

PoC #2:
```
root@vm-us4375:/etc/sudoers.d# ls -alh /usr/bin/wg
ls: cannot access '/usr/bin/wg': No such file or directory
```

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

To mitigate this issue, the developer team should validate the necessity of all sudo rules then remove all that are not considered an essential business requirement, which will improve the overall security of the systems by establishing enhanced deployment clarity.

NV-03-006 WP1: File permissions facilitate potential privilege escalation *(Info)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

Whilst assessing the inner functionality of the varying Docker containers, the observation was made that some containers exhibited what could be seen as a suboptimal and potentially dangerous practice. The startup sequence of the Docker containers executes multiple scripts and programs as root, which were writable by all other users on the system. Since Docker containers primarily operate with one sole user, this behavior should not necessarily be considered a direct security vulnerability and remains tangibly inexploitable in the current setup. Nevertheless, root users should never run a script writable to the world.

Vulnerable file #1:
/entrypoint.sh

Vulnerable code:

```
#!/usr/bin/env bash
entrypoint(){ sleep infinity; wait -n; }
       test -s /srv/salt/init.sls && salt-call --local --state-output=mixed
state.apply init
       test -f /run.sh && source /run.sh
Entrypoint
```

Vulnerable file #2:
/srv/salt/init.sls

Vulnerable file #3:
/run.sh

PoC:

```
root@us-us1:~/scan# ls -alh /srv/salt/init.sls
-rw-rw-rw- 1 root root 939 Sep 16 05:32 /srv/salt/init.sls

root@us-us1:~/scan# ls -alh /run.sh
-rw-rw-rw- 1 root root 73 Sep 16 05:32 /run.sh
```

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

To mitigate this issue, one can recommend integrating file permissions with stronger restrictions. Generally speaking, files executed by root should be owned by root and non-writable for any other user.

NV-03-007 WP1: Information disclosure via appropriated *Consul* token *(Info)*

***Note from NordVPN**: After investigation it was decided to accept the risk as this information is needed for runtime operations.*

Whilst inspecting the Docker containers, an authentication token belonging to the *Consul* service could be harvested and then appropriated to obtain information concerning the network and server or service enumeration within. Despite this, the present issue should be considered an information leak rather than a vulnerability in itself. The disclosed information should be considered sensitive, since it may prove valuable to an attacker and assist toward instigating any number of potential attack scenarios.

Configuration file:
/etc/consul-template/consul-template.hcl

Configuration values:

```
consul {
  address = "127.0.0.1:8500"
  namespace = ""
  token = "1ddxxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx3a"
```

Info leak excerpt:

- Services 10
  - all IPs/types/checks
- Nodes 3615
  - all IPs/types/checks

To mitigate this issue, one can recommend restricting the permissions that the deployed *Consul* token can request to the minimum required access. Enhanced protection for information of this nature will only strengthen the security foundation of the NordVPN server landscape as a whole.

NV-03-008 WP1: Critical components persist unpatched CVEs *(Low)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

Whilst assessing the critical services that NordVPN relies upon to run and maintain its infrastructure, the testing team placed emphasis on determining the presence of

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

potential vulnerabilities associated with the service versions. This was conducted by comparing the service versions and patch levels against databases of known vulnerable software versions and vendor patches. These efforts unearthed several services in the latest patch level that suffer from unpatched CVEs[1].

Some of these vulnerabilities could be considered problematic if exploitable[2][3], yet the primary issue identified here pertains to the fact that unpatched CVEs were located in a live system. This behavior indicates a systemic issue with patch management procedures that should be addressed.

To mitigate this issue, all services should exhibit the latest patch level, since unpatched services represent a primary vector for compromise. Since the task of retaining up-to-date services is by no means trivial, Cure53 advises identifying and integrating a software solution to assist with this effort.

Utilizing a service of this nature to monitor the deployed version and update the status of all software, then compare this information against sets of known CVEs and vulnerabilities, can play a crucial role in increasing awareness of unpatched services and retaining optimum security processes.

Finally, the importance of ensuring that a company's patch management is executed proficiently cannot be overstated and should be considered one of the fundamental pillars of practical information security.

NV-03-009 WP1: Hardening suggestions for Docker privilege escalation *(Info)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

Whilst analyzing the running configuration used within the Docker ecosystem utilized by NordVPN, the discovery was made that the current configuration was set to allow containers to append additional privileges. This should be seen as an insecure default and must be altered to ensure that containers cannot acquire new privileges.

By default, containers are permitted to acquire new privileges, so this configuration must be explicitly set. If any containers were to be compromised, an attacker could leverage the current configuration to gain additional privileges.

[1] https://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures
[2] https://www.cvedetails.com/cve/CVE-2022-36633/
[3] https://nvd.nist.gov/vuln/detail/CVE-2022-29153

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

To mitigate this issue, Cure53 recommends restricting container capabilities from acquiring additional privileges to increase the Docker service's overall security posture. If this is deemed impossible within the given environment, another approach to minimize any potential privilege-escalation attack would be to remove the *setuid* and *setgid* permissions for images.

NV-03-010 WP1: Hardening suggestions for Docker file systems *(Info)*

***Note**: This issue was fixed and the fix was verified as working properly by Cure53 via inspecting the respective diff. The problem as described no longer exists.*

In order to further increase the security posture of the Docker service used by NordVPN, some additional container-specific hardening could be implemented. By extending the current Docker configuration with the suggestions below, the overall attack surface of the Docker ecosystem would be greatly reduced.

The current Docker configuration used by NordVPN is set to enable both read and write operations from within a container. This explicitly enables the container process to read and write files owned by the container's actual runtime user and constitutes the default configuration for containers running within Docker.

In order to further improve the security posture of the current configuration, Cure53 advises mounting a temporary file system for containers requiring read and write capabilities. In cases where a writable file system remains unavoidable, Docker provides auditing and alteration rollback.

The file system in a Docker container is stacked as a series of layers. When a new container is created, a new layer is added as an overlay that can be written to. The Docker storage driver obfuscates this behind the scenes and presents it as a regular file system to the user. Note that writes made to a running container are made to this new layer.

NV-03-011 WP1: Hardening suggestions for Docker users *(Info)*

***Note from NordVPN**: Proposed hardening suggestions will be implemented in the nearest future.*

Generally speaking, when running production systems using Docker, containers operating with a unique user is considered sound practice. Conversely, container operations with a *root* user should be avoided when possible. In other words, one should ensure that the Docker file for each container image contains either *USER* or *ID*.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Specifically, the username or ID refers to the user identified in the container base image. If a specific user is not created in the container base image, the NordVPN process should utilize the *useradd* command to add a specific user before the *USER* instruction in the Docker file.

Example to command:
The following could be integrated to the Docker file to create a user within a given container:

RUN useradd -d /home/username -m -s /bin/bash username USER username

In addition to specifying a unique user or ID to a Docker container, Cure53 advises enabling Docker content trust within the Docker infrastructure. By doing so, the Docker service will enforce signing and verification of *image* tags, which will provide digital signatures that allow client-side verification of the integrity of specific *image* tags.
In order to enable Docker content trust, one must explicitly set it within the Docker ecosystem. Additional guidance regarding Docker content trust can be perused on the Docker Documentation website[4].

[4] https://docs.docker.com/engine/security/trust/

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Conclusions

The impressions gained during this report - which details and extrapolates on all findings identified during the CW38 through CW40 testing against the NordVPN servers and infrastructure by the Cure53 team - will now be discussed at length. To summarize, the confirmation can be made that the components under scrutiny have garnered a relatively strong impression, with a moderate yield of findings identified though plenty of opportunity for security growth observed.

The assessment focused on a few key areas that were deemed most relevant to NordVPN's network security in general. These namely constituted the network, host machines, containers, and services. Regarding testing coverage, the provided patch files for the Radius server and Wireguard kernel module were assessed for potential memory corruptions or authentication-related flaws. Positively, no associated issues were identified here.

Additionally, the network and firewall rules detected on all hosts and containers were subject to deep-dive evaluation. Similarly, these efforts did not unearth any issues or accidentally-exposed external services.

The network setup was assessed for the presence of any misconfiguration issues that could unintentionally incur component abuse, though none were identified in this regard. Considering the necessary complexity of the NordVPN infrastructure, this should be considered a praiseworthy outcome for the NordVPN team. Elsewhere, testing confirmed that the local network and internal communication used by the assessed services could benefit from additional hardening configuration. This viewpoint is corroborated by the instance of information leakage via the *Consul* service, which is further detailed in ticket NV-03-007.

The Cure53 team noted that the host setup and configuration was cleanly and concisely constructed, with evidence of sound security-principle implementation. However, one caveat to this constituted a local sudo-rule configuration that was deemed unrefined, partially superfluous, and potentially dangerous. In light of this, some issues related to this behavior are detailed in tickets NV-03-005, NV-03-002, and NV-03-003.

Furthermore, with reference to Docker containers and Linux servers in general, file system permissions must be explicitly set. Some occurrences of problematic file system permissions were observed, for instances within which the root user executes files writable or owned by other users (see NV-03-006).

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

The testing team noted that some of the network services and features were out of date. Whilst patch management is evidently by no means trivial to implement, dedicating adequate resources toward integrating it into the NordVPN lifecycle will prove significantly beneficial toward ensuring the company's product and reputation for excellence remains future proofed (see NV-03-008).

The overall Docker and container concept leveraged by NordVPN garnered a rather mixed impression. Though Cure53 did observe evidence of hardening and lockdown measures, the overall configuration more or less relied on default configurations. This viewpoint stands in contrast to the overall network exposure of the docker containers and attached services, which were observably and deliberately well maintained from a security perspective. The NordVPN team has clearly focussed on securing the network layer of its service.

The native Docker configuration established on the hosts assessed during this audit could benefit from additional hardening. The local Docker configuration relies on several configurations that should be considered insecure defaults and should be addressed at the earliest possible convenience. These suggested Docker configuration alterations can be found in tickets NV-03-009, NV-03-010, NV-03-011, NV-03-004, and NV-03-009.

Whilst assessing the container and local configuration aspects of the NordVPN infrastructure, the observation was made that extensive efforts had been made to secure the network layer or services exposed from the assessed host, which should be considered an integral measure toward providing comprehensive NordVPN asset protection. However, one can strongly recommend applying similar dedication toward securing the local Docker and host environments in order to further increase the security posture of the infrastructure as a whole. In summation for this NordVPN infrastructure assessment, Cure53 can only conclude that the current security posture exhibits a relatively strong impression from a security standpoint. Ample evidence of sound security configurations and well-maintained concepts for key aspects of the infrastructure were observed and positively noted by the testing team.

Nevertheless, the detection of eleven notable (albeit mostly miscellaneous) findings indicates ample leeway for hardening improvement. Moving forward, Cure53 strongly recommends initiating a follow-up security review of the NordVPN servers to ensure that all mitigations have been implemented correctly.

Cure53 would like to thank Lukas Jokubauskas and Karolis Pabijanskas from the Nord Security team for their excellent project coordination, support and assistance, both before and during this assignment.