Fine penetration tests for fine websites

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

Pentest-Report IVPN Gateway, Server & Setup 02.2023

Cure53, Dr.-Ing. M. Heiderich, MSc. D. Weißer

Index

- Introduction
- Scope
- Identified Vulnerabilities
  - IVP-05-001 WP1: World-readable config template reveals API key (Low)
  - IVP-05-003 WP2: Invalid DNS response crashes dnsfilter (Medium)
  - IVP-05-004 WP2: Outdated Go dependencies with known vulnerabilities (Low)
- Miscellaneous Issues
  - IVP-05-002 WP2: Files of deploy user can be overwritten (Info)
  - IVP-05-005 OOS: Secret keys present in Git repositories (Low)
  - IVP-05-006 WP2: Script not owned by root executed as root (Low)
  - IVP-05-007 WP1: Weak user-passwords on Linux can be easily cracked (Low)
  - IVP-05-008 WP1: Su command can be used by anyone (Low)
- Conclusions

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

Introduction

"What you do online can be tracked by organizations you may not know or trust and become part of a permanent record. A VPN can't solve this on its own, but can prevent your ISP from being able to share or sell your data."

From https://www.ivpn.net/

This report describes the results of a security assessment of the IVPN complex, spanning the IVPN VPN gateway server, as well as the IVPN server OS and OS setup. Carried out by Cure53 in February 2023, the project was a source code-assisted penetration test.

Registered as *IVP-05,* the project was requested by Privatus Limited in December 2022 and then scheduled for the beginning of 2023 to allow ample time for preparations on both sides. As for the precise timeline and specific resources, Cure53 completed the examination in CW08 of 2023. A total of six days were invested to reach the coverage expected for this assignment. In addition, it should be noted that a team of two senior testers has been composed and tasked with this project's preparation, execution and finalization.

For optimal structuring and tracking of tasks, the work was split into two separate work packages (WPs):

- **WP1**: Penetration tests and config review of IVPN VPN gateway server
- **WP2**: Source code-assisted penetration tests against IVPN server OS and OS setup

It can be derived from above that white-box methodology was utilized. Cure53 was provided with access to a test-server and all other means of access required to complete the tests. Moreover, all in-scope repositories were shared to make sure the project can be executed in line with the agreed-upon framework.

The project progressed effectively on the whole. All preparations were done in CW07 to foster a smooth transition into the testing phase. Over the course of the engagement, the communications were done using a private, dedicated RocketChat channel provided by Privatus Limited. All relevant personnel could join the discussions related to the project via RocketChat.

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

The discussions throughout the test were very good and productive and not many questions had to be asked. Ongoing interactions positively contributed to the overall outcomes of this project. The scope was well-prepared and clear, greatly contributing to the fact that no noteworthy roadblocks were encountered during the test.

Cure53 offered frequent status updates about the test and the emerging findings. Live-reporting was carried out by adding the findings into the GitHub provided by Privatus Limited. Additionally, the issues were occasionally shared by the testers via the mentioned RocketChat channel.

The Cure53 team managed to get very good coverage over the WP1-WP2 scope items. Among eight security-relevant discoveries, three were classified to be security vulnerabilities and five to be general weaknesses with lower exploitation potential. The overall number of flaws can be considered as relatively moderate and this can be interpreted as a positive sign in regard to the security of the inspected IVPN components. It can also be acknowledged that no vulnerabilities of *Critical* or *High* severity were unveiled during this audit, which further supports the positive impression gained throughout the audit.

In the following sections, the report will first shed light on the scope and key test parameters, as well as the structure and content of the WPs. Next, all findings will be discussed in grouped vulnerability and miscellaneous categories, then following a chronological order in each group. Alongside technical descriptions, PoC and mitigation advice are supplied when applicable.

Finally, the report will close with broader conclusions pertinent to this February 2023 project. Cure53 elaborates on the general impressions and reiterates the verdict based on the testing team's observations and collected evidence. Tailored hardening recommendations for the IVPN complex are also incorporated into the final section.

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

Scope

- **Penetration tests & Assessments against IVPN VPN gateway, server OS & OS setup**
  - **WP1**: Penetration tests and config review of IVPN VPN gateway server
    - **Access to test server**
      - *SSH access was granted to Cure53*
  - **WP2**: Source code-assisted penetration tests against IVPN server OS & OS setup
    - **Sources:**
      - Shared via WP1:
        - */var/tmp/repos/*
    - **Repositories in scope:**
      - aaa-failover
        - Commit:
          - *f4abb8cfd84c090883e3a10df9250852659a7a8b*
      - Gateway scripts
        - Commit:
          - *eb1038922b9f3001c51734ca1a20ead6678443f2*
      - Go-services
        - Applications:
          - *dnsfilter*
          - *wgdaemon*
        - Commit:
          - *3934d221067d828ce2c6fb48047aa1012aefa2d1*
      - Puppet repository
        - Commit:
          - *4a8938c42096dcb07358f5206faf3e65ccc14f18*
      - Wgportfw
        - Commit:
          - *64bbe425e822d178d47c76c020f2dbec3e6a75e4*
  - **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 sections list both vulnerabilities and implementation issues spotted during the testing period. Note that findings are listed in chronological order rather than by their degree of severity and impact. The aforementioned severity rank is simply given in brackets following the title heading for each vulnerability. Each vulnerability is additionally given a unique identifier (e.g., *IVP-05-001*) for the purpose of facilitating any future follow-up correspondence.

IVP-05-001 WP1: World-readable *config* template reveals API key *(Low)*

Fix notes: *This issue was fixed by IVPN during the testing time and the fix was verified by Cure53.*

While investigating the file permissions of sensitive files on the server, it was found that one *config* file is not properly secured against access from unprivileged users. Any local user on the system can access the contents of this file, including a secret API key for managing port-forwarding. However, as an attacker would already need to have some foothold into the system, the overall impact is *Low*.

Accessing *portfw-api.template*:

```
[c53dario@xb2 ~]$ cat /opt/ivpn/ivpn-gateway/etc/portfw-api.template
api_url          ████████████████████
api_key_id       ████████████████████
api_secret_key   ████████████████████
```

The other sensitive configuration files on the system were found to be properly secured. It is recommended to apply the correct file flags on this file as well to prevent access from unauthorized users.

IVP-05-003 WP2: Invalid DNS response crashes *dnsfilter* *(Medium)*

Studying the *dnsfilter* service reveals the application being prone to a crash with a malformed DNS response. Even though the service restarts automatically, this allows for DoS with relatively low effort. The flaw can, for example, be exploited by users with the ability to issue queries through the *dnsfilter*.

The following command demonstrates how the issue can be reproduced locally on the gateway server. A custom DNS server runs behind the domain *dd.h4x.tv*. The byte stream 00010001000000000000 represents a DNS record of type A but it is cut short, which is why it is invalid.

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

Receive invalid DNS answer:

```
dig 0001000100000000000-RAW.reflects.dd.h4x.tv @10.0.254.3
```

Crashlog of *dnsfilter*:

```
with error: Cannot check block list for IP: <nil>
panic: runtime error: invalid memory address or nil pointer dereference
[signal SIGSEGV: segmentation violation code=0x1 addr=0x0 pc=0x69bef3]

goroutine 29 [running]:
github.com/miekg/dns.(*Msg).SetReply(...)
       /Users/jurajhilje/go/pkg/mod/github.com/miekg/dns@v1.1.38/defaults.go:16
ivpn.net/go-services/services/dnsfilter/pkg/proxy.
(*DNSProxy).HandleRequest(0xc000064060, 0x779230, 0xc000118200, 0xc0000e2000)
       /Users/jurajhilje/golang/src/ivpn.net/go-services/services/dnsfilter/
pkg/proxy/proxy.go:68 +0x213
github.com/miekg/dns.HandlerFunc.ServeDNS(0xc0005249f0, 0x779230, 0xc000118200,
0xc0000e2000)
       /Users/jurajhilje/go/pkg/mod/github.com/miekg/dns@v1.1.38/server.go:37
+0x44
github.com/miekg/dns.(*ServeMux).ServeDNS(0xc0000d4020, 0x779230, 0xc000118200,
0xc0000e2000)

/Users/jurajhilje/go/pkg/mod/github.com/miekg/dns@v1.1.38/serve_mux.go:103 +0x5d
github.com/miekg/dns.(*Server).serveDNS(0xc0000d6360, 0xc00078c000, 0x54, 0x200,
0xc000118200)
       /Users/jurajhilje/go/pkg/mod/github.com/miekg/dns@v1.1.38/server.go:650
+0x2fd
github.com/miekg/dns.(*Server).serveUDPPacket(0xc0000d6360, 0xc00023bdf4,
0xc00078c000, 0x54, 0x200, 0x777498, 0xc00000e020, 0xc0000d41a0, 0x0, 0x0)
       /Users/jurajhilje/go/pkg/mod/github.com/miekg/dns@v1.1.38/server.go:590
+0xed
created by github.com/miekg/dns.(*Server).serveUDP
       /Users/jurajhilje/go/pkg/mod/github.com/miekg/dns@v1.1.38/server.go:520
+0x395
```

In the *HandleRequest* function, the request is passed to *proxy.getResponse.* The latter responds with a DNS message object upon successful resolution. However, in case of an invalid response, an error message *"Cannot check block list for IP"* is printed and *null* is returned. Despite the error, the *SetReply* function is called on that object, leading to the *null* pointer dereference shown in the backtrace above.

Affected file:

/go-services/services/dnsfilter/pkg/proxy/proxy.go

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

Affected code:

```
func (proxy *DNSProxy) HandleRequest(w dns.ResponseWriter, r *dns.Msg) {

	switch r.Opcode {
	case dns.OpcodeQuery:
		m, err := proxy.getResponse(r)
		if err != nil {
			log.Printf("Failed lookup for %s with error: %s", r,
err.Error())
			m.SetReply(r)
			w.WriteMsg(m)
			return
		}

		m.SetReply(r)
		w.WriteMsg(m)
	}
}
[...]
func (proxy *DNSProxy) getResponse(req *dns.Msg) (*dns.Msg, error) {
	[...]
	for _, a := range resp.Answer {
		switch a.Header().Rrtype {
		case dns.TypeA:
			[...]

			contains, err := proxy.blockList.Contains(aRR.A)
			if err != nil {
				return nil, fmt.Errorf("Cannot check block list for
IP: %s", aRR.A)
			}
```

Something as simple as a malformed DNS message should not be able to crash an application. It is recommended to make sure these scenarios are properly handled. In this case it should suffice to make sure that the object pointer is not *null.*

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

IVP-05-004 WP2: Outdated Go dependencies with known vulnerabilities *(Low)*

Some of the third-party dependencies linked to the tested Go applications are quite old and have known issues with CVEs dating back to 2019. As the vulnerable functions of the libraries are not used, this does not impact the security of the tested applications. However, this can quickly change, for example when new features are added to the complex.

To assess the security of third-party dependencies, the tools *govulncheck*[1], *trivy*[2] and *osv-scanner*[3] were used. Selected results of working with these tools against the *go-services* repository are shown below.

Results from *trivy* and *osv-scanner*:

```
test@test /tmp/go-services :D % trivy fs .
[...]
2023-02-20T15:32:03.216+0100     INFO   Detecting gomod vulnerabilities...

go.mod (gomod)

Total: 29 (UNKNOWN: 0, LOW: 3, MEDIUM: 8, HIGH: 15, CRITICAL: 3)
┌──────────────────────────┬────────────────────┬──────────┬─────────────┬...┐
│          Library         │    Vulnerability   │ Severity │ Installed   │...│
├──────────────────────────┼────────────────────┼──────────┼─────────────┼...┤
│ github.com/aws/aws-sdk-go │ CVE-2020-8911      │ MEDIUM   │ 1.37.15     │...│
│                          │                    │          │             │...│
│                          │                    │          │             │...│
│                          ├────────────────────┼──────────┤             ├...┤
│                          │ CVE-2020-8912      │ LOW      │             │...│
│                          │                    │          │             │...│
│                          │                    │          │             │...│
├──────────────────────────┼────────────────────┼──────────┼─────────────┼...┤
[...]
test@test /tmp/go-services :D % % osv-scanner  -r .
Scanning dir .
Scanning /tmp/go-services/ at commit 3934d221067d828ce2c6fb48047aa1012aefa2d1
Scanned /tmp/go-services/go.mod file and found 42 packages
Scanned /tmp/go-services/webservices/dnsleaktest/assets/package-lock.json file
and found 929 packages
┌──────────────────────────┬...┬──────────────────────────┬─────────┬...┐
│ OSV URL (ID IN BOLD)     │...│ PACKAGE                  │ VERSION │...│
├──────────────────────────┼...┼──────────────────────────┼─────────┼...┤
│ https://osv.dev/GO-2022-0646 │...│ github.com/aws/aws-sdk-go │ 1.37.15 │...│
[...]
```

[1] https://github.com/golang/vuln
[2] https://trivy.dev/
[3] https://github.com/google/osv-scanner

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

Just because no vulnerable path exists now, this possibility cannot be excluded from occurring in the future. It is generally recommended to keep dependencies up-to-date in order to prevent the IVPN complex from becoming vulnerable to known issues. Therefore, the versions of third-party packages in the *go.mod* files should be reviewed and updated.

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

Miscellaneous Issues

This section covers those noteworthy findings that did not lead to an exploit but might aid an attacker in achieving their malicious goals 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.

IVP-05-002 WP2: Files of *deploy* user can be overwritten *(Info)*

Fix notes: *This issue was fixed by IVPN during the testing time and the fix was verified by Cure53.*

The *sessioncounter* bash script consistently counts the number of active users and writes it to a temporary file. If a local, unprivileged attacker manages to gain access to the system, it would be possible to place a symlink in the location to which the scripts are written to, hence gaining a capacity to overwrite files owned by the user running the script.

The problem could potentially lead to a minor DoS but the impact is simply very limited, while the required access is extensive, hence explaining the downgraded severity of the issue. Shown below is an excerpt from the script, showing that a counter is written to */dev/shm/ovpnusers*.

Affected file:
/gateway-scripts/bin/sessioncounter

Affected code:

```
while true
do
  if read -r line <$pipe; then
    [...]
  echo "$counter" > /dev/shm/ovpnusers
done
```

It is recommended to write to a location which only the user running the script (*deploy*) can access. This makes sure that no unexpected symlinks are encountered.

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

IVP-05-005 OOS: Secret keys present in Git repositories *(Low)*

***Note**: The affected Git repository is private and not visible to the public*

While investigating the shared source code repositories, it was discovered that the *go-services* repository contained secret API keys used to process Bitcoin payments. There are more keys in the affected file, but these seem to be solely used for testing. However, the affected files were not in scope of this test.

Affected file:
/go-services/services/vpnapi/config/config.toml

Affected code:

```
[BtcPay]
URL= "https://pay1.ivpn.net"
Key = "eeb3b22[...]9d3fa3b236"
StoreId = "9cMvP1Frv[...]bS7hvYFwA2ao"
WebhookSecret = "Toq6[...]Cje1x9z9P"
```

Secrets should not be stored alongside the source code and deployed from an external source when setting up the application. In a follow-up discussion with the IVPN team, it was clarified that *puppet* rules for deploying secrets are already in place.

It is recommended to remove the affected file from the Git repository. Due to the nature of Git on the whole, the file would still remain in the commit history, which is why issuing new secrets is advised.

In addition, it is advisable to search the repositories for similar issues and possibly even automate this process in order to prevent such problems from happening in the future.

IVP-05-006 WP2: Script not owned by *root* executed as *root* *(Low)*

It was discovered that a *shell* script not owned by the *root* user could be executed with elevated privileges. The file belonging to the user *deploy* is executed as *root* when a connection is established to the socket on which a *socat* process listens. An attacker with the privileges of the *deploy* user could modify the script in order to escalate privileges. However, this requires having certain rights on the system before, therefore lowering the overall impact of the issue.

Process running as *root*:

```
root      2338870    4203  0 16:12 ?        00:00:00 socat
UNIX-LISTEN:/opt/ivpn/ivpn-gateway/var/spool/ivpn-connectd/connectd_cmd,user=roo
t,group=nobody,mode=0660,fork,setsid
EXEC:/opt/ivpn/ivpn-gateway/libexec/connectd
```

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

```
root     2338871 2338870  0 16:12 ?        00:00:00 /bin/bash /opt/ivpn/ivpn-
gateway/libexec/connectd
```

Script ownership:

```
[root@xb2 etc]# ls -al /opt/ivpn/ivpn-gateway/libexec/connectd
-rwxr-xr-x. 1 deploy deploy 9495 Feb 23 16:32
/opt/ivpn/ivpn-gateway/libexec/connectd
```

Files which are executed with high privileges should not be owned by low-privileged users. It is recommended to apply *root* ownership to the entire */opt/ivpn/ivpn-gateway/* directory, except for the files which need to be writable for the *deploy* user.

IVP-05-007 WP1: Weak user-passwords on Linux can be easily cracked *(Low)*

While running *hashcat* against the hashes in the system's shadow file, it was discovered that one password could be broken in a relatively quick manner. Using a GTX980 GPU with a set of basic passwords and rules, the process took only around 30 minutes.

Although reading the shadow file requires elevated privileges, this behavior can become a problem as the same hashes are globally deployed. For example, an attacker who has gained *root* access to system A and low-privileged access to system B could use the cracked password to fully compromise B as well. The problem described in IVP-05-008 further aids the abuse potential of cracked passwords. Shown below are the executed *hashcat* command and the corresponding results.

***Hashcat* command:**

```
hashcat -w3 -O –username -m 1800 hashfile passwords.txt -r best64.rule
```

Cracked hash:

```
$6$VBYfvJs7sEdZLv[redacted]
[redacted]IszVC1.:e[redacted]0
```

Even though the password does not seem overly weak, it is basically just a mutation of a word. It is recommended to consider enforcement of stronger passwords.

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

IVP-05-008 WP1: *Su* command can be used by anyone *(Low)*

Fix notes: *This issue was fixed by IVPN during the testing period and the fix was verified by Cure53.*

The *su* command is used to log in as another user. If the system is not configured properly, everyone is allowed to take advantage of this *util* and can authenticate as users whose passwords are known. For example, a compromised service, which runs on the server, makes it possible for an attacker to use or guess passwords for other accounts. This could directly assist the goals of escalating privileges.

As there is no common use-case for a service-user to be privy to *su,* it is advised to disallow this kind of usage. The issue can be solved in two steps. Firstly, all users who should be allowed to use *su* must be added to the *wheel* group. Secondly, the *su*-configuration needs some changes. A specific line shown next needs to be incorporated.

Affected file:
/etc/pam.d/su

Recommended changes in the *su*-configuration:

```
auth            required        pam_wheel.so use_uid
```

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

Conclusions

Cure53 concludes that the inspected IVPN aspects and components appear to already be in a quite solid state of security. This is evidenced by several observations made by two testers involved in this February 2023 project.

First, it should be underlined that the final list of findings is relatively short. Second, the severity scores ascribed to eight findings documented during this *IPV-05* project can be characterized as non-threatening, given the absence of *Critical* and *High*-level problems. Third, Cure53 must acknowledge that the IVPN team has successfully resolved multiple issues while the test was still ongoing. Hence, the security of the IVPN VPN gateway server, as well as the IVPN server OS and OS setup, is clearly on the right path.

To reiterate, the scope of the test included the IVPN gateway servers and related applications. These handle DNS filtering, port forwarding and distribution of WireGuard keys. The tasks performed by Cure53 included auditing the sources of the apps, reviewing the deployment process, and checking the server configuration for potential privilege escalation issues.

As a first area the *dnsfilter* application was inspected. This was dictated by the fact that it handles DNS queries coming from users. As such, it is the only part of the scope that gets in direct contact with user-input. The Go application does not offer much attack surface, but it was spotted that invalid DNS replies crash the process, thereby leading to Denial-of-Service (IVP-05-003).

None of the other Go applications talk to end-users directly, but rather connect to internal IVPN services like AMQP or MySQL, or they simply interface with the local filesystem. It was checked if compromised targets could impact the gateway, but no issues could be found here.

What leaves a bit of a bad taste is that the third-party dependencies of the Go applications are not kept up-to-date and have CVEs dating back to 2019. Even though this did not translate to direct security implications, it is still something that can be easily tackled and should clearly be avoided (IVP-05-004).

Some of the scripts running on the gateway are written in *bash*. Therefore, it was checked how they interacted with remote sources and the local filesystem. Notably, they do not deal with user-input directly and the usernames they handle are generated by IVPN. In effect, no meaningful attack vector could be identified, although room for improvement was demonstrated for the process of writing files locally (see IVP-05-002).

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

The deployment of the applications and configurations is handled by *Puppet.* The corresponding definitions were reviewed. *Puppet* also handles the deployment of secrets, so they do not have to be stored alongside the source code. While no issues were found with the *Puppet* deployment itself, it was found that some secrets find their way into the Git repositories (IVP-05-005).

Next, it was inspected how the underlying Linux system is configured. Cure53 was interested in routes for local attackers to elevate privileges. What immediately stuck out was that the system and all installed packages were up-to-date. Moreover, some safe defaults were configured, e.g., preventing the process list and *dmesg* from being seen by anyone. There were also no dangerous custom SUID binaries or cronjobs discovered on the system. However, a minor hardening suggestion to prevent non-authorized users from using the *su* command was given in IVP-05-008. This issue could be exploited in combination with weak passwords (IVP-05-007).

Even though most configurations containing secrets are protected from being read by anyone, one file where this is not the case was found and discussed in IVP-05-001. Furthermore, a non-*root*-owned *shell* script executed by *root* was also found (IVP-05-006).

Except for IVP-05-003 the discovered issues are not easy to exploit, requiring some sort of prior access to the local system. Therefore, most issues do not pose a direct risk and serve more as a guideline for further security ameliorations. This signifies an overall good impression of the tested sources and the Linux-based system gained by Cure53 during this February 2023 project.

Cure53 would like to thank Nick Pestell and Iain Douglas from the Privatus Limited team for their excellent project coordination, support and assistance, both before and during this assignment.