Radically Open Security

Penetration Test Report

Open Technology Fund

V 1.1
Amsterdam, January 23rd, 2025
Confidential

Document Properties

Client	Open Technology Fund
Title	Penetration Test Report
Targets	Devops systems OONI API Backend services Cross-platform development architecture plan OONI Run V2 Implementation
Version	1.1
Pentester	Morgan Hill
Authors	Morgan Hill, Marcus Bointon
Reviewed by	Marcus Bointon
Approved by	Melanie Rieback

Version control

Version	Date	Author	Description
0.1	October 16th, 2024	Morgan Hill	Initial draft
0.2	October 18th, 2024	Marcus Bointon	Review
0.3	October 23rd, 2024	Morgan Hill	Updates to severities
0.4	October 25th, 2024	Morgan Hill	Updates to severities
1.0	October 29th, 2024	Marcus Bointon	1.0
1.1	January 23rd, 2025	Morgan Hill	Retest updates

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

Table of Contents

1 Executive Summary

1.1 Introduction

Between August 5, 2024 and October 18, 2024, Radically Open Security B.V. carried out a penetration test for Open Technology Fund.

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

1.2 Scope of work

The scope of the penetration test was limited to the following targets:

- Devops systems
- OONI API
- Backend services
- Cross-platform development architecture plan
- OONI Run V2 Implementation

The scoped services are broken down as follows:

- Devops IaC review: 2-3 days
- Service architecture review: 2-3 days
- Static analysis services code base: 1-2 days
- Email magic link auth for Run links review: 2-3 days
- Reporting: 1 days
- Retesting: 1 days
- PM/Review: 2 days
- **Total effort: 11 - 15 days**

1.3 Project objectives

ROS will perform a penetration test of OONI with OTF in order to assess the security of OONI's new services and infrastructure as code. To do so ROS will access OONI's public repositories and guide OTF 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 August 5, 2024 and October 18, 2024.

1.5 Results In A Nutshell

During this crystal-box penetration test we found 14 Low, 2 High, 2 Moderate and 5 N/A-severity issues.

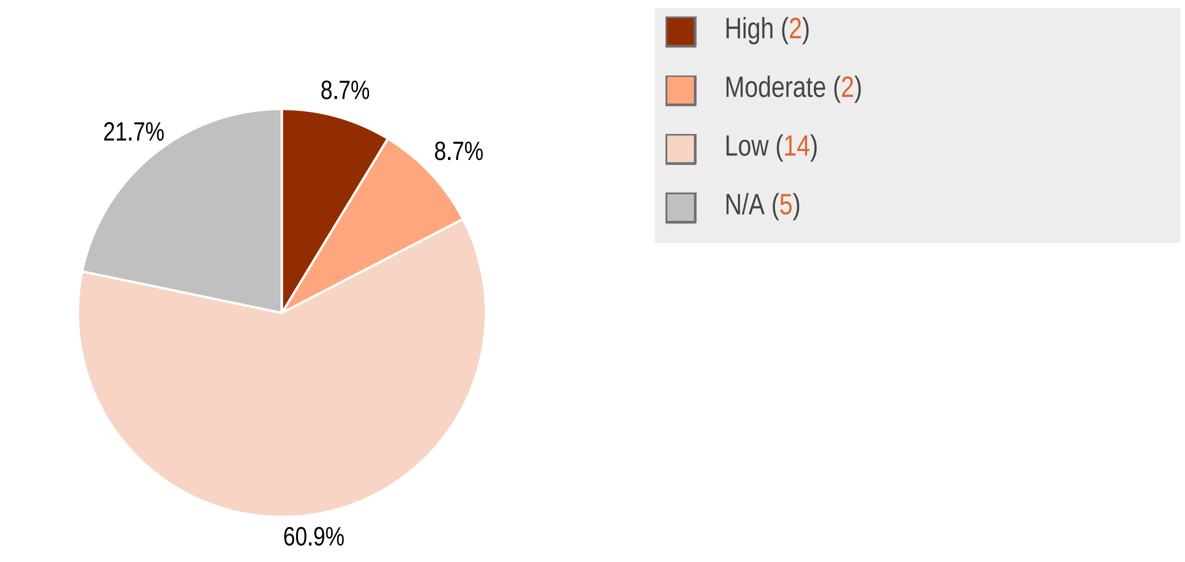
We did not discover any new initial entry points in this audit. Overall, we assess that the focus to date has been on perimeter security rather than defence in depth. The bulk of the findings would form bases for lateral movement and privilege escalation post initial entry. The KICS tool found many valid issues with the AWS configuration, and we strongly suggest making that part of regular maintenance procedures. Initial entry through any of the services would result in rapid compromise of the ClickHouse and PostgreSQL databases as well as code execution inside JupiterHub. Compromise of a member of the OONI DevOps team would also be very difficult to contain and would likely result in the effective loss of the infrastructure and a need to perform a clean restore.

1.6 Summary of Findings

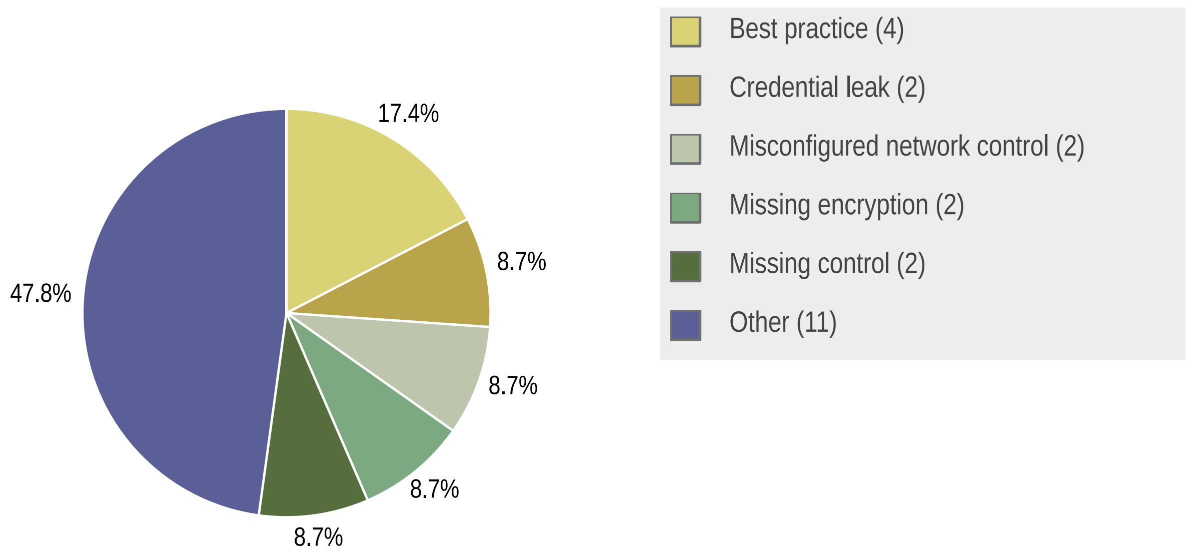
ID	Type	Description	Threat level
OTF-047	Privilege escalation	SSH users have passwordless sudo to root.	High
OTF-057	Shared credentials	The devops repository suggests that IAM user credentials are shared amongst contributors.	High
OTF-001	Misconfigured network control	The RDS PostgreSQL database instance is exposed to the internet.	Moderate
OTF-004	Credential leak	The RDS password is set randomly, but is stored as plain text in the Terraform state and may also be logged.	Moderate
OTF-043	Misconfiguration	ClickHouse ships a default user with no password that has not been removed in production.	Low
OTF-046	Credential leak	The JupyterHub admin password is recorded in the shell history during setup.	Low
OTF-045	Exposed credential	The password for JupyterHub is weak and is exposed in the repository.	Low
OTF-044	Weak credentials	The admin user of the ClickHouse database has a weak password.	Low
OTF-002	Missing encryption	Storage encryption is not enabled for the RDS PostgreSQL instance, so data at rest is not encrypted.	Low
OTF-009	Best practice	HTTP services do not automatically redirect to HTTPS	Low
OTF-010	Missing control	The AWS Application Load balancer can drop requests with invalid headers, but this is not enabled.	Low

OTF-012	Missing encryption	Service logs are not encrypted, resulting in increased exposure should sensitive fields appear in logs.	Low
OTF-018	Best practice	IAM policies should not be directly assigned to users, but assigned via roles.	Low
OTF-020	Best practice	There are SecretsManager secret resources without a KMS ID supplied, leading to the default Customer Master Key being used.	Low
OTF-048	Credential longevity	The session expiry is 180 days and the login expiry is 365 days in the ooniauth service.	Low
OTF-049	Authorization	The OONIDevopsPolicy is highly privileged.	Low
OTF-054	Input validation	The length of an email address is not validated before the email address is checked for validity with a regular expression.	Low
OTF-055	Misconfigured network control	Several hosts expose SSH access to the internet.	Low
OTF-008	Missing control	The ooniapi, oonidata, and oonith services are exposed without a web application firewall.	N/A
OTF-015	Best practice	ESC task definitions use the bridge networking mode rather than the recommended awsvpc mode.	N/A
OTF-022	Missing observability	VPC flow logs are not enabled in the VPC.	N/A
OTF-030	Cost optimization	The CloudWatch retention period is not set.	N/A
OTF-056	Network architecture	There are EC2 instances and ECS services that expose HTTP/HTTPS to the internet rather than only accepting connections via the load balancer.	N/A

1.6.1 Findings by Threat Level



1.6.2 Findings by Type



1.7 Summary of Recommendations

ID	Type	Recommendation
OTF-043	Misconfiguration	• Disable the default user.

		• Reorder the tasks in the role so that the configuration is rendered before ClickHouse is started.
OTF-046	Credential leak	• Don't provide passwords as command line parameters. Prefer providing them via standard input with no echo to shell, or via environment variables.
OTF-047	Privilege escalation	• Implement step-up authentication, like a password. • Consider whether all admins really need to be able to run *everything* as root; perhaps admins could manage with a subset of privileged commands.
OTF-057	Shared credentials	• Ensure each individual contributor has their own IAM user, and enforce MFA.
OTF-045	Exposed credential	• Set a strong random password instead of this default value. • If this value is in use, rotate it out and ensure the new value isn't checked in to the git repo. • Make use of git pre-commit hooks to avoid adding secrets to the repository.
OTF-001	Misconfigured network control	• Populate the `allowed_cidr_blocks` with a restrictive set of subnets for each environment. • If that's not viable, implement a point-to-point VPN tunnel from the external provider to the AWS VPC.
OTF-004	Credential leak	• Use the secrets manager to manage the password directly.
OTF-044	Weak credentials	• Set a strong random password for admin users. • Consider creating a separate admin account for each person performing an admin role.
OTF-002	Missing encryption	• Enable storage encryption.
OTF-009	Best practice	• Only provide services over HTTPS; redirect all HTTP requests to HTTPS.
OTF-010	Missing control	• Enable the feature to drop invalid headers.
OTF-012	Missing encryption	• Encrypt logs.
OTF-018	Best practice	• Assign policies through roles. • Ensure that each entity such as a person or automation has their own IAM user with roles commensurate with their role within OONI.
OTF-020	Best practice	• Create a KMS key for the secrets manager secret and assign it explicitly.
OTF-048	Credential longevity	• Reduce session lifetime to 30 days. • Allow users to revoke other sessions.
OTF-049	Authorization	• Enforce strong authentication for users with this role. • Review required permissions to cut down policies to only what is needed.
OTF-054	Input validation	• Shortcut email validation by doing a cheap length check before the more expensive regular expression. • Use a more accurate expression for matching addresses, or a proper address parser.
OTF-055	Misconfigured network control	• Restrict which source IPs have SSH access to the servers. • Provide SSH access via jumphost/bastion servers.

OTF-008	Missing control	• Evaluate the costs and potential effectiveness of a WAF.
OTF-015	Best practice	• Switch to `awsvpc` network mode for ECS tasks.
OTF-022	Missing observability	• Enable VPC flow logs with a reasonable retention period.
OTF-030	Cost optimization	• Set a log retention period of 7 – 30 days.
OTF-056	Network architecture	• Enforce that all HTTP/HTTPS traffic must go through the ALBs to ensure that any controls applied to the ALBs will be effective.

2 Methodology

2.1 Planning

Our general approach during audits is as follows:

1. **Context gathering**
   We attempt to gather as much information as possible about the target. By reading the available documentation and select snippets of code we gain an a high level understanding of the target. We aim to establish which services exist, how they interact and where they are deployed.
2. **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.
3. **Threat modeling**
   Using the understanding we have gathered about the target we seek to identify potential paths through the infrastructure. We pay explicit attention to network segmentation and authorizations to identify where they are looser than required. How an operator would defend actively against threats is also considered.
4. **Code audit**
   We question and inspect each line of code, to determine whether what it does matches what it was intended to do. We also contemplate the broader context, and suggest best practice alternatives where they exist to achieve the same application behaviour.

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 Findings

We have identified the following issues:

3.1 OTF-043 — ClickHouse default user not removed

Vulnerability ID: OTF-043 **Status:** Resolved

Vulnerability type: Misconfiguration

Threat level: Low

Description:

ClickHouse ships a `default` user with no password that has not been removed in production.

Technical description:

The default user is effectively an admin user with the ability to create accounts and grant permissions: https://clickhouse.com/docs/en/operations/access-rights#access-control-usage.

New user accounts are created, but the default account is not removed: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/ansible/roles/clickhouse/templates/ooni_users.xml.

ClickHouse is also started before the configuration is rendered, so even if the config had removed the default user, there would still be a time window during which the default user exists and is accessible: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/ansible/roles/clickhouse/tasks/main.yml#L59.

Impact:

Authentication bypass to database admin from any network where ClickHouse is reachable.

Recommendation:

Disable the default user by adding this snippet to the users section of the config:

```
<default remove="remove"></default>
```

Reorder the tasks in the role so that the configuration is rendered before ClickHouse is started.

Update 2024-10-23 14:54:

The severity of this finding has been reduced from high as the affected ansible role is not deployed in production.

Update 2024-12-05 17:00:

Significant changes have been made to make the configuration production ready. The default user can now only be accessed via localhost (https://github.com/ooni/devops/blob/main/ansible/group_vars/clickhouse/vars.yml#L177) and only has readonly permissions (https://github.com/ooni/devops/blob/main/ansible/group_vars/clickhouse/vars.yml#L157).

3.2 OTF-046 — JupyterHub admin password leaked to shell history

Vulnerability ID: OTF-046 **Status:** Not Retested

Vulnerability type: Credential leak

Threat level: Low

Description:

The JupyterHub admin password is recorded in the shell history during setup.

Technical description:

The password is passed as a command line parameter, and it will therefore end up in the shell history: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/ansible/roles/jupyterhub/tasks/main.yml#L33.

Impact:

An adversary that can read the shell history can become admin in Jupyterhub.

Recommendation:

- Don't provide passwords as command line parameters. Prefer providing them via standard input with no echo to shell, or via environment variables.

Update 2024-10-23 14:54:

The severity of this finding has been reduced from high as the affected ansible role is not deployed in production.

3.3 OTF-047 — SSH users root by another name

Vulnerability ID: OTF-047	**Status:** Resolved
Vulnerability type: Privilege escalation	
Threat level: High	

Description:

SSH users have passwordless sudo to root.

Technical description:

Admin users have passwordless privilege escalation to root via `sudo`. It is considered a best practice to re-authenticate users before they perform a privileged action. This form of unrestricted sudo also allows the circumvention of any logging or session capture that sudo would otherwise enable. https://github.com/ooni/devops/blob/main/ansible/roles/ssh_users/templates/sudoers#L3

Impact:

All admins have full root privileges via sudo. If an admin's SSH agent/ SSH key is compromised, then an attacker immediately obtains root access.

Recommendation:

- Implement step-up authentication, like a password.
- Consider whether all admins really need to be able to run *everything* as root; perhaps admins could manage with a subset of privileged commands.

Update 2024-12-05 17:00:

Further conversations have been had with the developers about the impact, higlighting avenues of SSH compromise such as malitious packages and editor plugins. The risk is understood.

The developers have expressed that it is not trivial to implement such a step-up authentication for sudo due to the requirement to syncronise accounts accross multiple machines.

Update `2025-01-13 17:00`:

Documentatin has been added to remind developers to password protect their SSH keys: https://github.com/ooni/devops/commit/b7590b3fe9771b02eaac9cead3e21cb4dddf93d8

It is agreed that this will reduce the risk.

3.4 OTF-057 — Shared IAM credentials

Vulnerability ID: OTF-057	**Status:** Resolved
Vulnerability type: Shared credentials	
Threat level: High	

Description:

The devops repository suggests that IAM user credentials are shared amongst contributors.

Technical description:

There is a `oonidevops_user` user in both development and production environments (a separate user in each env) that individual contributors are sharing credentials for. This user has extensive administrative privileges and is used to instantiate the infrastructure from the Terraform projects. The READMEs and the policy assignment OTF-018 (page 30) clearly suggest this practice.

Impact:

- Repudiability of individual contributors actions
- Increased credential exposure
- Limited MFA options (only 8 devices per user)

As an example, let's say Alice and Bob are sharing credentials for the `oonidevops_user`. Malory pops a shell on Bob's workstation via his favorite obscure editor plugin and proceeds to deploy some cryptocurrency mining workloads into OONI's AWS accounts in a region OONI doesn't use, so Alice and Bob won't notice in the console dashboard. The next AWS invoices come in and trigger an alarm as they are outrageously high. All the actions were carried out with the shared account, Alice and Bob both claim ignorance, and you're inclined to believe them, but know you need to run forensics on both of their workstations. In the meantime, your availability is impacted, because you revoked the users credentials as part of eradication so neither Alice nor Bob can perform routine maintenance or respond to errors, until you have identified the initial entry vector and are confident that new credentials can be issued without them immediately being compromised.

In a counterexample where each user has their own credentials, it is clear that Bob's credentials have been compromised and used. So you can initially revoke Bob's, account while Alice can continue working, and prioritize forensics on Bob's workstation. You can then test any indicators of compromise found from Bob's workstation on Alice's to ensure they haven't both been compromised, but you can do this after Malory has been successfully eradicated from Bob's workstation so Bob can work again while Alice's workstation is being checked.

Recommendation:

Ensure each individual contributor has their own IAM user, and enforce MFA. If the root account has credentials, those credentials should only be available to a very limited set of people either in a password vault that logs access to them or on hard copy in the incident response binders; strong MFA such as FIDO2 should also be required.

Update 2024-10-30 17:00:

In discussion with the developers they were able to point to the Terrform code that assigns the admin role to individual IAM users: https://github.com/ooni/devops/blob/edef980364e3f5d11f01ddef73329378b1d4b9c5/tf/environments/prod/main.tf#L87. We are therefore satisfied, that IAM crednentials are not being shared, and that the best practice is being followed.

Update 2025-01-13 17:00:

The developers have expanded on how they are doing auth:

https://github.com/ooni/devops/blob/main/tf/environments/prod/main.tf#L45 and https://github.com/ooni/devops/blob/main/tf/environments/prod/main.tf#L32, we actually have this setup in such a way where each administrator (which currently is only 2 people listed here: https://github.com/ooni/devops/blob/main/tf/environments/prod/main.tf#L76) has the permission to assume the role of the devops user. The reason why we do it this way is that we are able to apply slightly more narrowly scoped permissions to the user that performs actions on behalf of terraform (see: https://github.com/ooni/devops/blob/main/tf/modules/adm_iam_roles/main.tf#L11) as opposed to the individual user account (art and mehul) which have admin privileges.

3.5 OTF-045 — JupyterHub weak admin password

Vulnerability ID: OTF-045	**Status:** Resolved
Vulnerability type: Exposed credential	
Threat level: Low	

Description:

The password for JupyterHub is weak and is exposed in the repository.

Technical description:

The JupyterHub password stored in the Ansible vars is weak. It is also exposed in the public git repository: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/ansible/roles/jupyterhub/vars/main.yml#L4.

Impact:

If this password is current, anyone that finds it can log in to JupyterHub and run code.

Recommendation:

- Set a strong random password instead of this default value.
- If this value is in use, rotate it out and ensure the new value isn't checked in to the git repo.
- Make use of git pre-commit hooks to avoid adding secrets to the repository.

Update `2024-10-25 14:54`:

The severity of this finding has been reduced from High as the affected ansible role is not deployed in production.

Update `2024-11-23 17:00`:

The deployment of Jypyterhub has changed. Authentication is now done via PAM and there is no longer an admin user. https://github.com/ooni/devops/blob/db8334abe90276b5437f0db21bceca6a45c1cb39/ansible/roles/oonidata/tasks/jupyterhub.yml

3.6 OTF-001 — RDS publicly accessible

Vulnerability ID: OTF-001	**Status:** Resolved
Vulnerability type: Misconfigured network control	
Threat level: Moderate	

Description:

The RDS PostgreSQL database instance is exposed to the internet.

Technical description:

The RDS PostgreSQL instance has a public IP address. Access to the instance is restricted by a security group. The security group only allows PostgreSQL traffic (TCP 5432) from what should be a restricted set of subnets. The subset restriction is in the module level variable `allow_cidr_blocks` which defaults to `0.0.0.0/0`. This variable is not set in either environment, and consequently the development and production databases are exposed to the internet.

KICS flagged this issue:

```
{
  "query_name": "RDS DB Instance Publicly Accessible",
  "query_id": "35113e6f-2c6b-414d-beec-7a9482d3b2d1",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
db_instance#
publicly_accessible",
  "severity": "CRITICAL",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Insecure Configurations",
  "experimental": false,
  "description": "RDS must not be defined with public interface, which means the field
 'publicly_accessible' should not be set to 'true' (default is 'false').",
  "description_id": "c145a47f",
  "files": [
    {
      "file_name": "../../path/postgresql/main.tf",
      "similarity_id": "cd4f74514f2e57e13064da2eab13983ab1bd74badf6680f08e5517d92359748a",
      "line": 74,
      "resource_type": "aws_db_instance",
      "resource_name": "pg",
      "issue_type": "IncorrectValue",
      "search_key": "aws_db_instance[pg].publicly_accessible",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'publicly_accessible' should be set to false or undefined",
      "actual_value": "'publicly_accessible' is set to true",
      "remediation": "{\"after\":\"false\",\"before\":\"true\"}",
      "remediation_type": "replacement"
    }
  ]
}
```

Impact:

As long as updates are not being deferred then AWS will be keeping the RDS instance reasonably up to date. However, in the case of zero days against PostgreSQL, the exposure window may be too long and the instance could be compromised.

Regardless of PostgreSQL exploits in the wild, public exposure increases the volume of general internet noise such as failed login attempts in the logs, making security monitoring harder and more expensive.

Recommendation:

Populate the `allowed_cidr_blocks` with a restrictive set of subnets for each environment. If database access from outside AWS is required, and the originating subnet of the external queries is hard/impossible to control, then a point-to-point VPN tunnel from the external provider to the AWS VPC may provide a viable solution.

Update `2025-01-15 17:00`:

The PostgreSQL configuration has been tightened to only allow connections from private networks: https://github.com/ooni/devops/pull/153/files#diff-a341e7c3122b17f50d6f2d6c19efe73c4b657fd2875d0ec7092b51d995ab7f5eR172

3.7 OTF-004 — RDS password stored unencrypted in Terraform state

Vulnerability ID: OTF-004	**Status:** Resolved
Vulnerability type: Credential leak	
Threat level: Moderate	

Description:

The RDS password is set randomly, but is stored as plain text in the Terraform state and may also be logged.

Technical description:

Using the password field for configuring an RDS resource is not a best practice as it implies plain text storage of the password at least within the Terraform state.

The resource in question is here: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/tf/modules/postgresql/main.tf#L68.

KICS highlighted this issue, though perhaps more by accident than design as it believes the password is hard-coded. The password is not hard coded; it is generated with AWS secrets manager, but this is not a best practice either.

KICS output:

```
{
  "query_name": "Passwords And Secrets - Generic Password",
  "query_id": "487f4be7-3fd9-4506-a07a-eae252180c08",
  "query_url": "https://docs.kics.io/latest/secrets/",
  "severity": "HIGH",
  "platform": "Common",
  "cwe": "798",
  "cloud_provider": "COMMON",
  "category": "Secret Management",
  "experimental": false,
  "description": "Query to find passwords and secrets in infrastructure code.",
  "description_id": "d69d8a89",
  "files": [
    {
      "file_name": "../../path/postgresql/main.tf",
      "similarity_id": "97bef57776ee0f649eb19861de2b6cb4b81c983df22db396a41c9ece5797c482",
      "line": 68,
      "issue_type": "RedundantAttribute",
      "search_key": "",
      "search_line": 0,
      "search_value": "",
      "expected_value": "Hardcoded secret key should not appear in source",
      "actual_value": "Hardcoded secret key appears in source"
    }
  ]
}
```

Impact:

Gaining access to the Terraform state or logs will provide full access to the RDS instance.

Recommendation:

Use the secrets manager to manage the password directly, as described here: https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/db_instance#managed-master-passwords-via-secrets-manager-specific-kms-key.

Update 2025-01-15 17:00:

The RDS password is now managed and stored in secrets manager encrypted with KMS: https://github.com/ooni/devops/pull/153/files#diff-a341e7c3122b17f50d6f2d6c19efe73c4b657fd2875d0ec7092b51d995ab7f5eR239

Update 2025-01-22 17:00:

There is an issue in progress to reduce the scope and privilege levels of PostgreSQL users: https://github.com/ooni/devops/issues/156#issue-2803909862

3.8 OTF-044 — ClickHouse weak admin password

Vulnerability ID: OTF-044	**Status:** Resolved
Vulnerability type: Weak credentials	
Threat level: Low	

Description:

The admin user of the ClickHouse database has a weak password.

Technical description:

The admin user is restricted to logins only from localhost. However, it is plausible that a remote attacker could find a way to submit queries from localhost.

Impact:

Local privilege escalation.

Recommendation:

- Set a strong random password for admin users.
- Consider creating a separate admin account for each person performing an admin role.

Update 2024-10-25 14:54:

The severity of this finding has been reduced from High as the affected ansible role is not deployed in production.

Update 2024-11-22 17:00:

The new ClickHouse deployment uses a random password managed by the secrets manager: https://github.com/ooni/devops/blame/84ce023fcf290583bbd4a20d73f597f5636cc144/ansible/group_vars/clickhouse/vars.yml#L191

3.9 OTF-002 — RDS unencrypted data at rest

Vulnerability ID: OTF-002 **Status:** Resolved

Vulnerability type: Missing encryption

Threat level: Low

Description:

Storage encryption is not enabled for the RDS PostgreSQL instance, so data at rest is not encrypted.

Technical description:

The PostgreSQL RDS instance does not have storage encryption enabled, leaving data at rest in plain text. The nature of some of the data in PostgreSQL is non-public.

KICS highlighted this issue:

```
{
  "query_name": "DB Instance Storage Not Encrypted",
  "query_id": "08bd0760-8752-44e1-9779-7bb369b2b4e4",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
db_instance#storage_encrypted",
  "severity": "HIGH",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Encryption",
  "experimental": false,
  "description": "AWS DB Instance should have its storage encrypted by setting the parameter to
 'true'. The storage_encrypted default value is 'false'.",
  "description_id": "88ca11bc",
  "files": [
    {
      "file_name": "../../path/postgresql/main.tf",
      "similarity_id": "c57ac09bb835af63aff4fbe423308d0e1fa5fdfd5a139848b677b487e0837bdf",
      "line": 57,
      "resource_type": "aws_db_instance",
      "resource_name": "pg",
      "issue_type": "MissingAttribute",
      "search_key": "aws_db_instance[pg]",
      "search_line": 57,
      "search_value": "",
      "expected_value": "'storage_encrypted' should be set to true",
      "actual_value": "'storage_encrypted' is undefined or null"
    }
  ]
}
```

Impact:

A lack of encryption at rest increases the risk of data exposure in the event that an adversary obtains access to a database snapshot, or the storage volume attached to the database instance.

Recommendation:

Enable storage encryption. It must be enabled from the creation of the instance. Migration can be done from a clone of a snapshot. Ensure that the unencrypted snapshots and volumes are destroyed once migrated to minimize exposure.

Update `2025-01-13 17:00`:

The developers have concluded that the data in the RDS instance is not sensitive and therefor does not necessitate encryption at rest.

3.10 OTF-009 — HTTP does not redirect to HTTPS

Vulnerability ID: OTF-009

Vulnerability type: Best practice

Threat level: Low

Description:

HTTP services do not automatically redirect to HTTPS

Technical description:

Redirecting clients from HTTP to HTTPS ensures that clients will not use an insecure connection to you services.

It is straightforward to have the Application Load Balancer redirect all HTTP requests to HTTPS automatically.

KICS highlights this issue:

```
{
  "query_name": "ALB Listening on HTTP",
  "query_id": "de7f5e83-da88-4046-871f-ea18504b1d43",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
lb_listener",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Networking and Firewall",
  "experimental": false,
  "description": "AWS Application Load Balancer (alb) should not listen on HTTP",
  "description_id": "47a8608d",
  "files": [
    {
      "file_name": "../../path/ooni_dataapi/main.tf",
      "similarity_id": "1f0e8d6934229a62924641d68ec873e752e79985974ab035ceffca947e589b5b",
      "line": 315,
      "resource_type": "aws_alb_listener",
      "resource_name": "front_end",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb_listener[front_end].default_action",
      "search_line": 315,
      "search_value": "",
      "expected_value": "'default_action.redirect.protocol' should be equal to 'HTTPS'",
      "actual_value": "'default_action.redirect' is missing"
    },
    {
      "file_name": "../../path/ooniapi_service/main.tf",
      "similarity_id": "4278bffe333874ec433efb878d46037d7f8166380660b924fd8da3bf8642deb4",
      "line": 186,
      "resource_type": "aws_alb_listener",
      "resource_name": "ooniapi_service_http",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb_listener[ooniapi_service_http].default_action",
      "search_line": 186,
      "search_value": "",
      "expected_value": "'default_action.redirect.protocol' should be equal to 'HTTPS'",
      "actual_value": "'default_action.redirect' is missing"
    },
    {
      "file_name": "../../path/oonith_service/main.tf",
      "similarity_id": "e9efe3303042ba8c1f16723e4c78c27d7b6a834515acc3570fc6f12ce434cca0",
      "line": 182,
      "resource_type": "aws_alb_listener",
      "resource_name": "oonith_service_http",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb_listener[oonith_service_http].default_action",
      "search_line": 182,
      "search_value": "",
      "expected_value": "'default_action.redirect.protocol' should be equal to 'HTTPS'",
      "actual_value": "'default_action.redirect' is missing"
    }
  ]
}
```

Impact:

A client could choose to communicate over HTTP, allowing the interception of their requests and credentials.

Recommendation:

- Only provide services over HTTPS; redirect all HTTP requests to HTTPS.

3.11 OTF-010 — ALB does not filter invalid headers

Vulnerability ID: OTF-010 **Status:** Not Retested

Vulnerability type: Missing control

Threat level: Low

Description:

The AWS Application Load balancer can drop requests with invalid headers, but this is not enabled.

Technical description:

One feature of the ALB is that is can drop requests that contain invalid headers, reducing load from malformed requests on your service. This option is not on by default, and therefore must be enabled.

KICS highlights this issue:

```
{
  "query_name": "ALB Not Dropping Invalid Headers",
  "query_id": "6e3fd2ed-5c83-4c68-9679-7700d224d379",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
lb#drop_invalid_header_fields",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Best Practices",
  "experimental": false,
  "description": "It's considered a best practice when using Application Load Balancers to drop
 invalid header fields",
  "description_id": "7560e4d2",
  "files": [
    {
      "file_name": "../../path/ooniapi_frontend/main.tf",
      "similarity_id": "66a200d7f359d16ff68cc3c884b394f56c0c0d08cec83244ad2e71588f97b093",
      "line": 5,
      "resource_type": "aws_alb",
      "resource_name": "${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[{{ooniapi}}]",
      "search_line": 5,
      "search_value": "",
      "expected_value": "aws_alb[{{ooniapi}}].drop_invalid_header_fields should be set to true",
      "actual_value": "aws_alb[{{ooniapi}}].drop_invalid_header_fields is missing",
      "remediation": "drop_invalid_header_fields = true",
      "remediation_type": "addition"
    },
    {
      "file_name": "../../path/ooni_dataapi/main.tf",
      "similarity_id": "33777d229b04aa5aa067d0cb60459c721f4e7b4922efb8c53674f7c23eac5deb",
      "line": 302,
      "resource_type": "aws_alb",
      "resource_name": "ooni-tier1-oonidataapi",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[{{oonidataapi}}]",
      "search_line": 302,
      "search_value": "",
      "expected_value": "aws_alb[{{oonidataapi}}].drop_invalid_header_fields should be set to true",
      "actual_value": "aws_alb[{{oonidataapi}}].drop_invalid_header_fields is missing",
      "remediation": "drop_invalid_header_fields = true",
      "remediation_type": "addition"
    },
    {
      "file_name": "../../path/ooniapi_service/main.tf",
      "similarity_id": "dc0c089db7477648b3481404becba51b0fc21cf0201a8f8dc9c4c7a2539e974c",
      "line": 173,
      "resource_type": "aws_alb",
      "resource_name": "${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[{{ooniapi_service}}]",
      "search_line": 173,
      "search_value": "",
      "expected_value": "aws_alb[{{ooniapi_service}}].drop_invalid_header_fields should be set to
 true",
      "actual_value": "aws_alb[{{ooniapi_service}}].drop_invalid_header_fields is missing",
      "remediation": "drop_invalid_header_fields = true",
      "remediation_type": "addition"
    },
    {
      "file_name": "../../path/oonith_service/main.tf",
      "similarity_id": "46b0916096634ea834931bad4fa06b7250e79ae616cfbd2ae2a6d42f35052cd5",
      "line": 165,
      "resource_type": "aws_alb",
      "resource_name": "${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[{{oonith_service}}]",
      "search_line": 165,
      "search_value": "",
      "expected_value": "aws_alb[{{oonith_service}}].drop_invalid_header_fields should be set to
 true",
      "actual_value": "aws_alb[{{oonith_service}}].drop_invalid_header_fields is missing",
      "remediation": "drop_invalid_header_fields = true",
      "remediation_type": "addition"
    }
  ]
}
```

Impact:

Malformed requests are allowed to proceed too far into the application stack, being forwarded and creating load on the underlying services. Invalid headers could trigger unknown vulnerabilities in the services.

Recommendation:

- Enable the feature to drop invalid headers.

Update 2025-01-13 17:00:

An issue is open to make the changes: https://github.com/ooni/devops/issues/144.

3.12 OTF-012 — Logs are not encrypted with KMS

Vulnerability ID: OTF-012

Vulnerability type: Missing encryption

Threat level: Low

Description:

Service logs are not encrypted, resulting in increased exposure should sensitive fields appear in logs.

Technical description:

Logs should be encrypted to reduce exposure in case sensitive information leaks into them, for example from a crash. Encryption in CloudWatch log groups is not enabled by default, and must be provided with a key from AWS KMS in order to begin encrypting log entries.

KICS highlighted this issue:

```
{
  "query_name": "CloudWatch Log Group Without KMS",
  "query_id": "0afbcfe9-d341-4b92-a64c-7e6de0543879",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
cloudwatch_log_group",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Encryption",
  "experimental": false,
  "description": "AWS CloudWatch Log groups should be encrypted using KMS",
  "description_id": "4258abe6",
  "files": [
    {
      "file_name": "../../path/ooniapi_service/main.tf",
      "similarity_id": "d667807cde615c1d7b30f9cb5eafc8b9ad800e07b4d7252e82e2919ebdd5a81f",
      "line": 39,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "ooni-ecs-group/${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[ooniapi_service]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "Attribute 'kms_key_id' should be set",
      "actual_value": "Attribute 'kms_key_id' is undefined"
    },
    {
      "file_name": "../../path/ooni_dataapi/main.tf",
      "similarity_id": "0df8d7200e8629aa163874ca9ba17a0aebf873745ab8090a7743ab130cbee19f",
      "line": 1,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "tf-ecs-group/app-dataapi",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[app]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "Attribute 'kms_key_id' should be set",
      "actual_value": "Attribute 'kms_key_id' is undefined"
    },
    {
      "file_name": "../../path/ecs_cluster/main.tf",
      "similarity_id": "e2adfd88e61faa7bcf0ff3d05eece7a7402277499de9ad222347a8fb423472ab",
      "line": 1,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "ooni-ecs-group/${var.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[ooniapi_services]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "Attribute 'kms_key_id' should be set",
      "actual_value": "Attribute 'kms_key_id' is undefined"
    },
    {
      "file_name": "../../path/oonith_service/main.tf",
      "similarity_id": "762be4542b4e13584cfc37a3f5f9edbbb3bbaa6582f7aea717a4e9e1987e0c14",
      "line": 39,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "ooni-ecs-group/${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[oonith_service]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "Attribute 'kms_key_id' should be set",
      "actual_value": "Attribute 'kms_key_id' is undefined"
    }
  ]
}
```

Impact:

Sensitive information leaked into logs is more exposed due to a lack of encryption at rest.

Recommendation:

Create a KMS key for logs and provide it to the log groups to start encrypting logs. Review any existing logs for sensitive information disclosure and delete as required.

3.13 OTF-018 — IAM policies are assigned directly to users

Vulnerability ID: OTF-018	**Status:** Resolved
Vulnerability type: Best practice	
Threat level: Low	

Description:

IAM policies should not be directly assigned to users, but assigned via roles.

Technical description:

It is considered best practice to assign policies to roles rather than assigning policies directly to users.

For example the devops policy should be assigned to a devops role and individual contributors should have their own IAM users with these roles. This scenario is preferred for several reasons:

- No shared credentials
  - non-repudiation
  - simpler revocation
  - reduced exposure of credentials
- Less complex policies with many users
  - policies can grant access to role principals rather than lists of individual users
  - familiar model for onboarding new contributors

- Simplifies migration to SSO
    - AWS roles could be assumed via login with an external IdP

KICS highlighted this issue:

```
{
  "query_name": "IAM Policies Attached To User",
  "query_id": "b4378389-a9aa-44ee-91e7-ef183f11079e",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
iam_policy_attachment",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Access Control",
  "experimental": false,
  "description": "IAM policies should be attached only to groups or roles",
  "description_id": "32ec58ec",
  "files": [
    {
      "file_name": "../../path/oonidevops_github_user/main.tf",
      "similarity_id": "d592f1b26479d5e0cd4015b4a3307f9ccbab60ab36e68205eb0f8345fadb6956",
      "line": 15,
      "resource_type": "aws_iam_user_policy_attachment",
      "resource_name": "oonidevops_github",
      "issue_type": "RedundantAttribute",
      "search_key": "aws_iam_user_policy_attachment[{{oonidevops_github}}].user",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'user' is redundant",
      "actual_value": "'user' exists"
    },
    {
      "file_name": "../../path/ooniapi_user/main.tf",
      "similarity_id": "5b3edb255f67656dfa0ac1585a9d1289bf35e1464d0461a0818d42c3900225c0",
      "line": 7,
      "resource_type": "aws_iam_user_policy",
      "resource_name": "oonidevops-ooniapi-policy",
      "issue_type": "RedundantAttribute",
      "search_key": "aws_iam_user_policy[{{ooniapi}}].user",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'user' is redundant",
      "actual_value": "'user' exists"
    }
  ]
}
```

Impact:

Increased overhead and complexity when managing multiple users and their access.

Recommendation:

- Assign policies through roles.
- Ensure that each entity such as a person or automation has their own IAM user with roles commensurate with their role within OONI.

Update 2025-01-13 17:00:

The developers have provided further detail on their use of users/roles:

These are special users that are not assigned to humans, but rather to services. We use these users to allow our CI to run checks on terraform with very narrowly scoped permissions. Since we don't expect the changes to affect multiple users, we don't see this adding additional complexity in managing multiple users. We could create a role for each of these, but the role would always be 1-1 mapped onto these users and so that would be unnecessary additional overhead.

3.14 OTF-020 — SecretsManager default Customer Master Key

Vulnerability ID: OTF-020	**Status:** Not Retested
Vulnerability type: Best practice	
Threat level: Low	

Description:

There are SecretsManager secret resources without a KMS ID supplied, leading to the default Customer Master Key being used.

Technical description:

When the KMS ID is not set for a SecretManager Secret then a default CMK will be used/created named 'aws/secretsmanager'. As this is not explicit, it will not be tracked in Terraform state. Policies such as key rotation are also not applied to the default CMK.

KICS highlighted this issue:

```
{
  "query_name": "Secretsmanager Secret Without KMS",
  "query_id": "a2f548f2-188c-4fff-b172-e9a6acb216bd",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
secretsmanager_secret#kms_key_id",
  "severity": "MEDIUM",
 "platform": "Terraform",
 "cloud_provider": "AWS",
 "category": "Encryption",
 "experimental": false,
 "description": "AWS Secretmanager should use AWS KMS customer master key (CMK) to encrypt the
secret values in the versions stored in the secret",
 "description_id": "c9c95b59",
 "files": [
   {
     "file_name": "../../path/ooniapi_user/main.tf",
     "similarity_id": "62a72c853154f8f5ab90d4925a2916fcc114cb41155a0e95ca126fb9945b93f8",
     "line": 39,
     "resource_type": "aws_secretsmanager_secret",
     "resource_name": "oonidevops/ooniapi_user/aws_access_key_id",
     "issue_type": "MissingAttribute",
     "search_key": "aws_secretsmanager_secret[{{aws_access_key_id}}]",
     "search_line": -1,
     "search_value": "",
     "expected_value": "aws_secretsmanager_secret.kms_key_id should be defined and not null",
     "actual_value": "aws_secretsmanager_secret.kms_key_id is undefined or null"
   },
   {
     "file_name": "../../path/oonidevops_github_user/main.tf",
     "similarity_id": "ea9ec955cbc0ff47c0da0e7b03459118cccff8faa57e061665273205e54e9c70",
     "line": 23,
     "resource_type": "aws_secretsmanager_secret",
     "resource_name": "oonidevops/github_user/access_key_json",
     "issue_type": "MissingAttribute",
     "search_key": "aws_secretsmanager_secret[{{oonidevops_github}}]",
     "search_line": -1,
     "search_value": "",
     "expected_value": "aws_secretsmanager_secret.kms_key_id should be defined and not null",
     "actual_value": "aws_secretsmanager_secret.kms_key_id is undefined or null"
   },
      {
     "file_name": "../../path/postgresql/main.tf",
     "similarity_id": "5fe96fc884007b8e98c1aa135a625602ba30816e782060af6ed0f7c83a57f115",
     "line": 46,
     "resource_type": "aws_secretsmanager_secret",
     "resource_name": "oonidevops/name of the postgresql instance/pg_password",
     "issue_type": "MissingAttribute",
     "search_key": "aws_secretsmanager_secret[{{pg_password}}]",
     "search_line": -1,
     "search_value": "",
     "expected_value": "aws_secretsmanager_secret.kms_key_id should be defined and not null",
     "actual_value": "aws_secretsmanager_secret.kms_key_id is undefined or null"
   },
   {
     "file_name": "../../path/ooniapi_user/main.tf",
     "similarity_id": "d53ed25b423dcbcb71e3b8e739726b01174b8d2b1ff9a3b9b5aa6d33aeeba715",
     "line": 28,
     "resource_type": "aws_secretsmanager_secret",
     "resource_name": "oonidevops/ooniapi_user/aws_secret_access_key",
     "issue_type": "MissingAttribute",
     "search_key": "aws_secretsmanager_secret[{{aws_secret_access_key}}]",
     "search_line": -1,
     "search_value": "",
     "expected_value": "aws_secretsmanager_secret.kms_key_id should be defined and not null",
     "actual_value": "aws_secretsmanager_secret.kms_key_id is undefined or null"
   },
   {
      "file_name": "../../path/adm_iam_roles/main.tf",
      "similarity_id": "17b9e986da6a8da3ef49f6d3b86c826885ceef7b63bf7759c30d08fcab824252",
      "line": 81,
      "resource_type": "aws_secretsmanager_secret",
      "resource_name": "oonidevops/deploy_key/ssh_key_private",
      "issue_type": "MissingAttribute",
      "search_key": "aws_secretsmanager_secret[{{oonidevops_deploy_key}}]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "aws_secretsmanager_secret.kms_key_id should be defined and not null",
      "actual_value": "aws_secretsmanager_secret.kms_key_id is undefined or null"
    }
  ]
}
```

Impact:

An untracked KMS key resource is created, that may be accidentally deleted, losing access to secrets stored in the secrets manager.

Recommendation:

- Create a KMS key for the secrets manager secret and assign it explicitly.

Update 2025-01-13 17:00:

There is an issue open to make these changes: https://github.com/ooni/devops/issues/146.

3.15 OTF-048 — Long token validity in ooniauth

Vulnerability ID: OTF-048	**Status:** Resolved
Vulnerability type: Credential longevity	
Threat level: Low	

Description:

The session expiry is 180 days and the login expiry is 365 days in the ooniauth service.

Technical description:

Long session/login lifetimes are set in the task environment: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/tf/environments/prod/main.tf#L466.

Impact:

Long session/login lifetimes give an attacker an extended period of access if they compromise a user's session/login token. The user would have very little ability to evict the attacker and regain control over their account beyond creating a new account with a new email address and potentially asking OONI to remove the compromised account manually.

Recommendation:

Session lifetime is always a trade-off for usability though 180 days seems quite extreme. 30 days would be more reasonable. A token revocation system would also help.

Update `2025-01-13 17:00`:

The expiry has been retuce to 2 and 7 days respevivly and key rotation implemented: https://github.com/ooni/devops/pull/143.

3.16 OTF-049 — Broad AWS policy

Vulnerability ID: OTF-049	**Status:** Not Retested
Vulnerability type: Authorization	
Threat level: Low	

Description:

The `OONIDevopsPolicy` is highly privileged.

Technical description:

OONIDevopsPolicy is effectively "allow everything" as it has `iam:*`, set, so it can simply create and attach a new policy with whichever actions it desires: https://github.com/ooni/devops/blob/2ad136daa82acbaee868865eac39abaa8b3e7312/tf/modules/adm_iam_roles/main.tf#L16.

Full access to logs could allow an attacker to delete logs, destroying evidence that would be useful when investigating intrusions: https://github.com/ooni/devops/blob/99cd52ddc82ed9c5bc8fa5a44f4ff28eea63940d/tf/modules/ecs_cluster/templates/profile_policy.json#L41.

https://github.com/ooni/devops/blob/99cd52ddc82ed9c5bc8fa5a44f4ff28eea63940d/tf/modules/ooni_dataapi/templates/instance_profile_policy.json#L20

https://github.com/ooni/devops/blob/99cd52ddc82ed9c5bc8fa5a44f4ff28eea63940d/tf/modules/ooniapi_service/templates/profile_policy.json#L20

Impact:

Users/roles with the Devops policy are full administrators that can do almost everything apart from root-only actions such as closing the account. Compromise of users/roles with this policy would be catastrophic.

Recommendation:

Tightly control any users/roles with this policy by enforcing strong authentication. Review which permissions are actually used by users fulfilling this rule and cut down the policy to just the required actions.

Update 2025-01-13 17:00:

An internal issue has been created. The aim is to resolved the issue before the report is published.

3.17 OTF-054 — Email address length not restricted

Vulnerability ID: OTF-054	**Status:** Not Retested
Vulnerability type: Input validation	
Threat level: Low	

Description:

The length of an email address is not validated before the email address is checked for validity with a regular expression.

Technical description:

Email addresses have a maximum length of 320 ASCII characters according to RFC 3696 (https://www.rfc-editor.org/rfc/rfc3696#section-3). The registration endpoint does not enforce this length constraint before validating the email address with a regular expression: https://github.com/ooni/backend/blob/4fcc22b823617bcba1d0ff2e9bd11145937152de/api/ooniapi/auth.py#L292

This means computation resources are wasted further validating out of spec email addresses and that an attacker can lock up resources for longer by sending unreasonably long email addresses.

We also note that while the regular expression used allows an arguably justifiable *intentional* subset of valid addresses (that notably excludes Unicode email addresses, which have been valid since 2012), it also allows some invalid addresses, which should never be permitted.

Impact:

This could be used for a denial of service.

Recommendation:

- Shortcut email validation by doing a cheap length check before the more expensive regular expression.
- Use a more accurate expression for matching addresses, or a proper address parser.

Update 2025-02-13 17:00:

An issue has been opened: https://github.com/ooni/devops/issues/147.

3.18 OTF-055 — SSH open to the internet

Vulnerability ID: OTF-055 **Status:** Not Retested

Vulnerability type: Misconfigured network control

Threat level: Low

Description:

Several hosts expose SSH access to the internet.

Technical description:

Several hosts expose SSH to the internet on TCP port 22.

`oonibackend_proxy` instances are associated with the `nginx_sg` which permits SSH access from `0.0.0.0/0`.

`container_host` instances are associated with the `container_host` security group which allows SSH access from `admin_cidr_ingress`. The default value for this module level variable is `0.0.0.0/0` and it is not set in either environment.

`codesign_box` instances are associated with the `hsm` security group which permits SSH access from `0.0.0.0/0`.

`app` instances are associated with the `instance` security group which allows SSH from `admin_cidr_ingress`. The default for this module level variable is `0.0.0.0/0` and it is not set in either environment.

`clickhouse_server_prod_ter1` is associated with `clickhouse_sg` which allows SSH from `admin_cidr_ingress`. This has no default and clickhouse is not included in either environment.

Impact:

Increased exposure to potential SSH vulnerabilities. Automatic security updates are enabled, however the exposure window may be sufficient for compromise.

Each host also has resources wasted and logs polluted by failed login attempts.

Recommendation:

Access these hosts via a Jumpbox/Bastion hosts to limit exposure. AWS SSM may be an option if a vendor-specific solution is acceptable. These hosts do not appear to need public IP addresses, so the security groups can restrict SSH access to the jumphost.

Update `2025-01-13 17:00`:

An internal issue has been created with the aim of resolving the finding before the report is published.

3.19 OTF-008 — Services are exposed without a WAF

Vulnerability ID: OTF-008	**Status:** Not Retested
Vulnerability type: Missing control	
Threat level: N/A	

Description:

The ooniapi, oonidata, and oonith services are exposed without a web application firewall.

Technical description:

A web application firewall (WAF) is a layer-7 control that can intercept and block or rate limit HTTP requests based on the detection of common attacks or headers such as user agent.

AWS offers a WAF that can be integrated directly with their Application Load Balancer service. No WAF has been enabled in the environment.

KICS spotted this issue:

```
{
  "query_name": "ALB Is Not Integrated With WAF",
  "query_id": "0afa6ab8-a047-48cf-be07-93a2f8c34cf7",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
wafregional_web_acl_association",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Networking and Firewall",
  "experimental": false,
  "description": "All Application Load Balancers (ALB) must be protected with Web Application
 Firewall (WAF) service",
  "description_id": "4e4c668d",
  "files": [
    {
      "file_name": "../../path/oonith_service/main.tf",
      "similarity_id": "e28bdbed061d340a9c4e3b78df9b67224ec28e1eb255a8015350847320639b81",
      "line": 165,
      "resource_type": "aws_alb",
      "resource_name": "${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[oonith_service]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'aws_alb[oonith_service]' should not be 'internal' and has a
 'aws_wafregional_web_acl_association' associated",
      "actual_value": "'aws_alb[oonith_service]' is not 'internal' and does not have a
 'aws_wafregional_web_acl_association' associated"
    },
    {
      "file_name": "../../path/ooni_dataapi/main.tf",
      "similarity_id": "57f4c037965bb8ea5e0b97a1e5276749a2a338e7837882d2bfb4aa378f9af5c8",
      "line": 302,
      "resource_type": "aws_alb",
      "resource_name": "ooni-tier1-oonidataapi",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[oonidataapi]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'aws_alb[oonidataapi]' should not be 'internal' and has a
 'aws_wafregional_web_acl_association' associated",
      "actual_value": "'aws_alb[oonidataapi]' is not 'internal' and does not have a
 'aws_wafregional_web_acl_association' associated"
    },
         {
      "file_name": "../../path/ooniapi_frontend/main.tf",
      "similarity_id": "a3b2824bf9f9cdb0d6268fa61a0648fcbb385a476c34175cbc26fe321d436476",
      "line": 5,
      "resource_type": "aws_alb",
      "resource_name": "${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[ooniapi]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'aws_alb[ooniapi]' should not be 'internal' and has a
 'aws_wafregional_web_acl_association' associated",
      "actual_value": "'aws_alb[ooniapi]' is not 'internal' and does not have a
 'aws_wafregional_web_acl_association' associated"
    },
    {
      "file_name": "../../path/ooniapi_service/main.tf",
      "similarity_id": "0400e779a687ab90cdf5f11ce3e58c9724de126bdf6281b09f394a86e815c78c",
      "line": 173,
      "resource_type": "aws_alb",
      "resource_name": "${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_alb[ooniapi_service]",
      "search_line": -1,
      "search_value": "",
      "expected_value": "'aws_alb[ooniapi_service]' should not be 'internal' and has a
 'aws_wafregional_web_acl_association' associated",
      "actual_value": "'aws_alb[ooniapi_service]' is not 'internal' and does not have a
 'aws_wafregional_web_acl_association' associated"
    }
  ]
}
```

Impact:

Any bugs such as XSS or SQL injection introduced in the service are potentially more exposed to exploitation. The ability to rate-limit based on headers/user agents is also limited.

Recommendation:

Evaluate the costs and potential effectiveness of a WAF. A WAF and in particular the AWS WAF might not be the best fit. Many CDNs offer rate limiting and other WAF like functionality which may be more effective at mitigating the threats within OONI's threat model.

Update 2025-01-13 17:00:

An issue has been opened to track this finding: https://github.com/ooni/devops/issues/148.

3.20 OTF-015 — ECS network mode not recommended

Vulnerability ID: OTF-015	**Status:** Not Retested
Vulnerability type: Best practice	
Threat level: N/A	

Description:

ESC task definitions use the `bridge` networking mode rather than the recommended `awsvpc` mode.

Technical description:

The `bridge` networking mode assigns one Elastic Network Interface (ENI) per host, and uses a bridge on each host to share that interface among containers. Whereas the `awsvpc` mode assigns an ENI for each container. Having an ENI for each container is preferable as security groups and monitoring (e.g. flow logs) can then be applied at a container level.

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/security-network.html#security-network-task-networking

KICS highlighted this issue:

```
{
  "query_name": "ECS Task Definition Network Mode Not Recommended",
  "query_id": "9f4a9409-9c60-4671-be96-9716dbf63db1",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
ecs_task_definition#network_mode",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Insecure Configurations",
  "experimental": false,
  "description": "Network_Mode should be 'awsvpc' in ecs_task_definition. AWS VPCs provides the
 controls to facilitate a formal process for approving and testing all network connections and
 changes to the firewall and router configurations",
  "description_id": "61f295c5",
  "files": [
    {
      "file_name": "../../path/ooniapi_service/main.tf",
      "similarity_id": "6d3fc967499422bcdc3973c587f657f87bd741405cba2bfe33f2382976653701",
      "line": 58,
      "resource_type": "aws_ecs_task_definition",
      "resource_name": "ooniapi_service",
      "issue_type": "IncorrectValue",
      "search_key": "aws_ecs_task_definition[ooniapi_service].network_mode",
      "search_line": 58,
      "search_value": "",
      "expected_value": "'network_mode' should equal to 'awsvpc'",
      "actual_value": "'network_mode' is equal to 'bridge'",
      "remediation": "{\"after\":\"awsvpc\",\"before\":\"bridge\"}",
      "remediation_type": "replacement"
    },
    {
      "file_name": "../../path/oonith_service/main.tf",
      "similarity_id": "38288bc77351f1056f8d8ebcf44dc22b9af17e3a4316ab6d9d9189dd43523d95",
      "line": 59,
      "resource_type": "aws_ecs_task_definition",
      "resource_name": "oonith_service",
      "issue_type": "IncorrectValue",
      "search_key": "aws_ecs_task_definition[oonith_service].network_mode",
      "search_line": 59,
      "search_value": "",
      "expected_value": "'network_mode' should equal to 'awsvpc'",
      "actual_value": "'network_mode' is equal to 'bridge'",
      "remediation": "{\"after\":\"awsvpc\",\"before\":\"bridge\"}",
      "remediation_type": "replacement"
    }
  ]
}
```

Impact:

Decreased precision in network controls and observability.

Recommendation:

- Switch to `awsvpc` network mode for ECS tasks.

Update `2025-01-13 17:00`:

An issue has been opened regarding this finding: https://github.com/ooni/devops/issues/149.

3.21 OTF-022 — VPC flow logs disabled

Vulnerability ID: OTF-022	**Status:** Not Retested
Vulnerability type: Missing observability	
Threat level: N/A	

Description:

VPC flow logs are not enabled in the VPC.

Technical description:

VPC flow logs are a record of network traffic within the VPC. Flow logs detail how much data was transmitted between which locations (IP and port) when. This type of log is extremely useful for security monitoring, forensics, and also debugging. It is advisable to have them on before you need them.

KICS highlights this issue:

```
{
  "query_name": "VPC FlowLogs Disabled",
  "query_id": "f83121ea-03da-434f-9277-9cd247ab3047",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/vpc",
  "severity": "MEDIUM",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Observability",
  "experimental": false,
  "description": "Every VPC resource should have an associated Flow Log",
  "description_id": "cdbdeb30",
  "files": [
    {
      "file_name": "../../path/network_noipv6/main.tf",
      "similarity_id": "77826325ec7c0d797257e68c76147b01530c18c8816298cfa8266adebb886fab",
      "line": 6,
      "resource_type": "aws_vpc",
      "resource_name": "main",
      "issue_type": "IncorrectValue",
      "search_key": "aws_vpc[main]",
      "search_line": 6,
      "search_value": "",
      "expected_value": "aws_vpc[main] should be the same as Flow Logs VPC id",
      "actual_value": "aws_vpc[main] is not the same as Flow Logs VPC id"
    },
    {
      "file_name": "../../path/network/main.tf",
      "similarity_id": "9d3b5e67674361cfa6b2ca32a8a8cf6af7de66323356c8864f97050503d76e5e",
      "line": 6,
      "resource_type": "aws_vpc",
      "resource_name": "main",
      "issue_type": "IncorrectValue",
      "search_key": "aws_vpc[main]",
      "search_line": 6,
      "search_value": "",
      "expected_value": "aws_vpc[main] should be the same as Flow Logs VPC id",
      "actual_value": "aws_vpc[main] is not the same as Flow Logs VPC id"
    }
  ]
}
```

Impact:

Less information is available in the event of a compromise, and a reduced probability of detecting intrusion.

Recommendation:

- Enable VPC flow logs with a reasonable retention period.

Update 2025-01-13 17:00:

An issue has been opened to track this finding: https://github.com/ooni/devops/issues/150.

3.22 OTF-030 — CloudWatch retention unset

Vulnerability ID: OTF-030	**Status:** Not Retested
Vulnerability type: Cost optimization	
Threat level: N/A	

Description:

The CloudWatch retention period is not set.

Technical description:

Logs should be kept only for as long as they are useful, after which time they should be deleted. This reduces exposure of any sensitive information in them and also reduces storage costs. When retention is not set, logs are retained indefinitely.

KICS highlights this issue:

```
{
  "query_name": "CloudWatch Without Retention Period Specified",
  "query_id": "ef0b316a-211e-42f1-888e-64efe172b755",
  "query_url": "https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/
cloudwatch_log_group",
  "severity": "INFO",
  "platform": "Terraform",
  "cloud_provider": "AWS",
  "category": "Observability",
  "experimental": false,
  "description": "AWS CloudWatch Log groups should have retention days specified",
  "description_id": "64f08509",
  "files": [
    {
      "file_name": "../../path/ooniapi_service/main.tf",
      "similarity_id": "c7661abcd04e0d2f47b43fd3dd7b360400a602c3e4c78e4ef7f98ee26f605417",
      "line": 39,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "ooni-ecs-group/${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[ooniapi_service]",
      "search_line": 39,
      "search_value": "",
      "expected_value": "Attribute 'retention_in_days' should be set and valid",
      "actual_value": "Attribute 'retention_in_days' is undefined",
      "remediation": "retention_in_days = 7",
      "remediation_type": "addition"
    },
    {
      "file_name": "../../path/ooni_dataapi/main.tf",
      "similarity_id": "570f61edf10d76fd2a98cad27c558bcdb8ead5d6659d1ea2aae6a51a6e97e5a5",
      "line": 1,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "tf-ecs-group/app-dataapi",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[app]",
      "search_line": 1,
      "search_value": "",
      "expected_value": "Attribute 'retention_in_days' should be set and valid",
      "actual_value": "Attribute 'retention_in_days' is undefined",
      "remediation": "retention_in_days = 7",
      "remediation_type": "addition"
    },
       {
      "file_name": "../../path/ecs_cluster/main.tf",
      "similarity_id": "7d87024bd4234261809fceaaa047846f8778d5cc62fbde4c78059469a2975162",
      "line": 1,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "ooni-ecs-group/${var.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[ooniapi_services]",
      "search_line": 1,
      "search_value": "",
      "expected_value": "Attribute 'retention_in_days' should be set and valid",
      "actual_value": "Attribute 'retention_in_days' is undefined",
      "remediation": "retention_in_days = 7",
      "remediation_type": "addition"
    },
    {
      "file_name": "../../path/oonith_service/main.tf",
      "similarity_id": "f511182273f89ae749491ca293bc726c89e1403bb2d67d7ea574328bf9ab55a5",
      "line": 39,
      "resource_type": "aws_cloudwatch_log_group",
      "resource_name": "ooni-ecs-group/${local.name}",
      "issue_type": "MissingAttribute",
      "search_key": "aws_cloudwatch_log_group[oonith_service]",
      "search_line": 39,
      "search_value": "",
      "expected_value": "Attribute 'retention_in_days' should be set and valid",
      "actual_value": "Attribute 'retention_in_days' is undefined",
      "remediation": "retention_in_days = 7",
      "remediation_type": "addition"
    }
  ]
}
```

Impact:

Logs are retained for longer than they are useful, and incur unnecessary cost.

Recommendation:

- Set a log retention period of 7 – 30 days.

Update 2025-01-13 17:00:

An issue has been created for this finding: https://github.com/ooni/devops/issues/151"

3.23 OTF-056 — HTTP/HTTPS exposed directly from instances to the internet

Vulnerability ID: OTF-056	**Status:** Not Retested
Vulnerability type: Network architecture	
Threat level: N/A	

Description:

There are EC2 instances and ECS services that expose HTTP/HTTPS to the internet rather than only accepting connections via the load balancer.

Technical description:

There are web services that should be publicly accessible, however the instances themselves do not need to be publicly accessible to function as there are application load balancers to mediate public traffic to private instances.

`oonibackend_proxy` exposes HTTP on a public address despite the instances being part of an ALB target group.

The `ooniapi_service` and `oonith_service` both allow direct access despite bing load balanced.

The `app` EC2 instance running the `oonidataapi` ECS service allows HTTP/HTTPS directly from the internet.

Impact:

The Application Load Balancers aren't effective choke points for HTTP/HTTPS traffic therefore any controls applied to them (such as DDoS protection) can be bypassed.

Recommendation:

- Enforce that all HTTP/HTTPS traffic must go through the ALBs to ensure that any controls applied to the ALBs will be effective.

Update 2025-01-13 17:00:

An issue for tracking this finding has been created:https://github.com/ooni/devops/issues/152.

4 Non-Findings

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

4.1 NF-042 — Separate AWS accounts for dev and prod

There are separate AWS accounts for development and production. This is a good practice to maintain separation between development and production environments. Good account hygiene should be practiced to ensure this separation is effective in practice.

4.2 NF-051 — OONI auth assertion used for validation

An assertion is used to check that the login time has been set in the token. Assertions should not be used for data validation as they are typically disabled when running python in production mode.

Login time is either generated in process or taken from the token which is trusted, in the sense that it is encrypted with a secret: https://github.com/ooni/backend/blob/ffe750e4ebd0416f2061d4d81587537e621d73ae/ooniapi/services/ooniauth/src/ooniauth/routers/v2.py#L204

If it was somehow unset, an exception would be raised when trying to get the timestamp from the unset login time.

5 Future Work

- **SSH known hosts update**

  Currently there is a Terraform module with shell script for collecting known host keys. This improves on the typical TOFU model of SSH host keys by sharing the known hosts among multiple users. Ensuring that the host key person A trusts is the same key trusted by person B.

  An even stronger alternative could be to employ an SSH host CA, whereby a certificate is provisioned for each host (or the host has credentials to obtain a certificate) and users trust these certificates via trusting a CA in their SSH config. Certificates can also be used for user authentication, eliminating the need to distribute admin SSH keys to each host. HashiCorp Vault has basic SSH CA capabilities that may be worth considering. If the project is growing its number of admins much beyond current numbers then there isn't much value in doing this, but should it ever expand, SSH certs are an often overlooked good option.
- **Auth JWT shared secret**

  OONI service uses the HMAC variant of JWTs for authorization tokens. HMAC requires a shared secret that each service has to have access to in order to verify the token. With access to this secret, a token can be signed, so its exposure should be minimized. As it stands, a vulnerability in any of the services could leak the shared secret, allowing an attacker to sign tokens for their own purposes.

  As an alternative, asymmetric cryptography can be used in the form of either an RSA or ECDSA key pair. By employing asymmetric cryptography for the token signature, secret exposure can be reduced. Only the auth service would need the private key and the other services would only need the public key to verify the authenticity of the token. This could help in restricting post compromise lateral movement. It is also important to consider how these keys would be rotated.
- **Limit egress traffic from EC2 instances**

  Limiting egress will make lateral movement harder and also make the instances less useful for an attacker post compromise. If connections out to the internet are restricted then it is much more difficult to establish a reverse shell or exfiltrate information. It is likely that during normal operation, instances only need to connect to update servers and could therefore have their egress dramatically restricted.
- **Run an IaC scanning tool regularly**

  Run a tool such as KICS (https://www.kics.io/) or Checkov (https://checkov.io) regularly against Ansible roles and Terraform modules to keep on top of best practices.
- **Retest of findings**

  When mitigations for the vulnerabilities described in this report have been deployed, a repeat test should be performed to ensure that they are effective and have not introduced other security problems.

- **Regular security assessments**
  Security is a process that must be continuously evaluated and improved; this penetration test is just a single snapshot. Regular audits and ongoing improvements are essential in order to maintain control of your corporate information security.

6 Conclusion

We discovered 14 Low, 2 High, 2 Moderate and 5 N/A-severity issues during this penetration test.

OONI has recently begun the process of rearchitecting services with an emphasis on infrastructure as code (IaC). This audit has taken place at a relatively early stage in the transition, seeking to provide security-minded input, to be incorporated into a firm foundation for a strong security posture. Special attention was paid to core components, such as the email-link-based authentication system.

During the course of the audit, we did not discover any issues that would lead to the immediate compromise of any part of OONI's applications or infrastructure. The scope was principally optimized for breadth, with automated scans establishing a quality baseline, across all services looking only for common defects. More depth-first scopes in each service, such as that applied to the email authorization, may uncover deeper issues.

A key component of the scope was infrastructure. Basic IaC scanning raised a substantial volume of leads. Most of these leads ended up as findings after further manual review and modeling. These findings were typically hardening issues that place a lot of importance on the perimeter holding. Areas such as applying network segmentation and the principle of least privilege could be improved to restrict lateral movement and add further layers of defense.

Many of the findings were common AWS misconfigurations, caught with automated tooling. As highlighted in future work (page 49), running a tool such as KICS in CI would provide a high-impact, low-cost path to an improved security posture. Given the complexity of modern cloud platforms, it is difficult to keep track of the best practices for every service, which makes automated scanning of IaC extremely valuable.

Another area of focus was the email-based authentication for run links. The core concept is a sensible compromise between usability and strong authentication. A typical services user account is only as protected as the recovery email account, so it makes sense not to force another password on the user. However, provisions for "day 2" operations such as credential rotation and revocation could be improved.

All in all, the OONI services and their infrastructure in the scope of this audit have a reasonable security posture. There are clear improvements that can be made to ratchet up defenses. We hope that the recommendations in this report are actionable, and that early-stage security involvement pays dividends in establishing a sustainable platform.

Working on such projects is a great privilege, as well as an exciting challenge, to translate enterprise-focused concepts into something appropriate for projects operating with small core teams. We appreciate The Open Technology Fund's enablement of this work on OONI. We also look forward to seeing OONI's important work prosper into the future.

We recommend fixing all of the issues found and then performing a retest in order to ensure that mitigations are effective and that no new vulnerabilities have been introduced.

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

Morgan Hill	Morgan is a seasoned security consultant with a background in IoT and DevOps. He currently specialises in Rust and AVoIP.
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.