OCCT — OS Compliance Check Tool

User Manual (Prototype)

Document version: 1.1 (Prototype)
Product version: OCCT Prototype (November 2025)
Authors: OCCT Capstone Team
Client/Mentor: Dr Imran Makhdoom (UTS)

Table of Contents

Document Control

Version	Date	Author	Summary of Changes
1.0	03 Nov 2025	OCCT Team	Initial user manual for prototype
1.1	06 Nov 2025	OCCT Team	Include screenshot integrated descriptions

1. Introduction

1.1 What is OCCT

OCCT (OS Compliance Check Tool) is a full-stack prototype that automates Windows OS security auditing. It checks configurations against security frameworks (Common Criteria) and presents dashboards with evidence and remediation hints.

1.2 Key Benefits

- Automated checks of high-impact settings (e.g., firewall defaults, local admin membership).
- Visibility into privileged accounts and account activity tied to security events.
- Evidence-driven findings with framework mapping and remediation hints.

1.3 In-Scope vs Out-of-Scope

In scope (prototype):

- Windows 11 local host checks, Live Mode collectors, and Sample Mode with seeded data
- YAML-driven controls library with mappings to CC
- Dashboard, controls, and evidence views

Out of scope (prototype):

- Fleet-wide remote scanning/agent deployment
- Auto-enforcement/patching
- Email/SIEM integrations
- Group policy/OS cross-compatibility

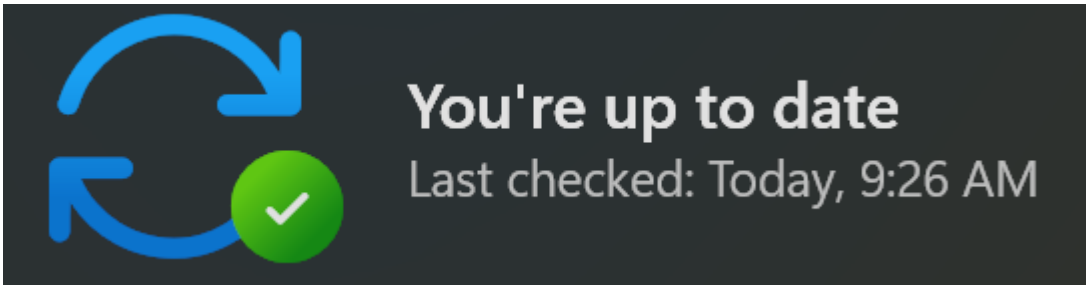
2. System Requirements

- **Operating System**: Windows 11, run as local administrator for full checks.
- **Software & Runtime:** PowerShell 5.1+, Python 3.9+, modern browser (e.g. Chrome).
- **Permissions**: Elevated PowerShell for querying firewall, local groups, and event logs.

3. Installation & Setup

3.1 Prepare the Windows Host

1. Ensure Windows is up to date
2. Confirm local admin access



3.2 Install OCCT Backend & Frontend

1. Use this Download Link to download a zip file containing the latest version of the application

2. Unzip the file and store it in the preferred location
3. Open a PowerShell terminal with Administrator privileges and find the location of the folder (occt-tool)

```
# In PowerShell

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned -Force
python -m venv env
.\env\Scripts\activate

pip install -r requirements.txt

python -m backend.app
```

The above should start the application on localhost: http://127.0.0.1:5000

```
(env) PS C:\Users\     \occt-tool> python -m backend.app
[detections] start_live_poller_if_enabled called
[detections] live poller started (ids=(4625, 4728, 4732, 4624), every 15s, lookback=5m, dedupe=0s)
[detections] live poller started (ids=(4625, 4728, 4732, 4624), every 15s, lookback=5m, threshold=5)
[detections] +0 events, +0 alerts (published 0; sent_to 0 clients; clients_now=0 bus_id=1887699704848 pid=25256),
 bookmark=897469
[OK] Ingested 25 audit, 7 events, 2 detections into C:\Users\     \occt-tool\backend\instance\occt.db
[auto-ingest] backend.ingest_samples completed
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server ins
tead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

4. Architecture Overview

4.1 Components

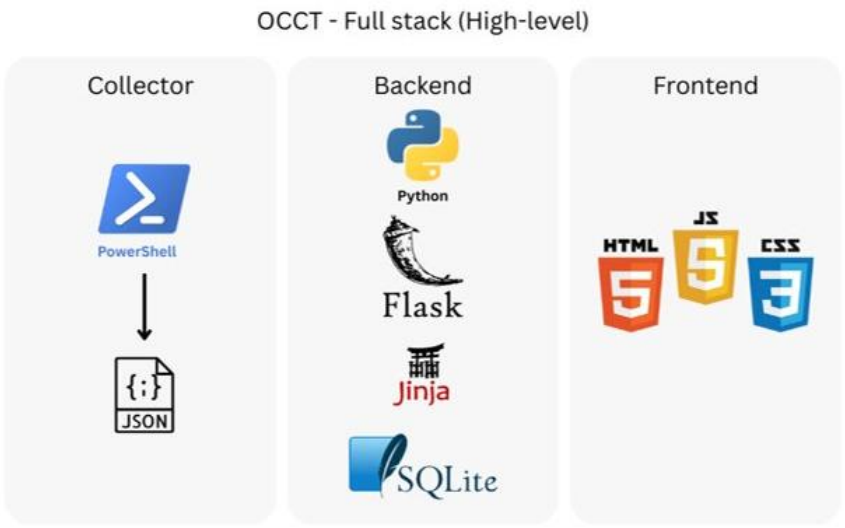
• **Collectors** (PowerShell): Query Windows configuration, groups, firewall, events.

• **Backend** (Python/Flask + SQLAlchemy): APIs, evidence processing, SQLite persistence.

• **Controls** Library (YAML): Control IDs, titles, mappings, severity, remediation.

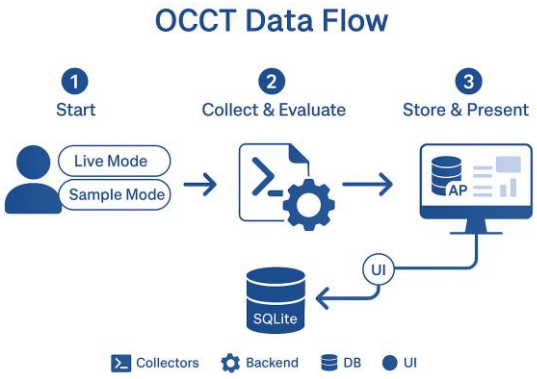
• **Frontend** (Flask/JS/Jinja): Dashboard, controls pages, account activity, evidence views.

High-level architecture overview:



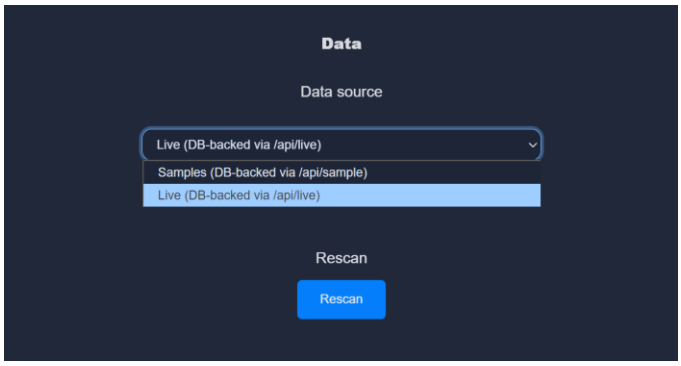
4.2 Data Flow

1. User triggers Scan (Live) or opens Sample Mode
2. Collectors gather facts/events → Backend evaluates against controls.yml
3. Results stored in SQLite → UI renders dashboards and evidence sent by API



4.3 Modes (Live vs Sample)

- **Live Mode**: Execute checks on the local Windows host.
- **Sample Mode**: Explore preloaded evidence without touching the host.

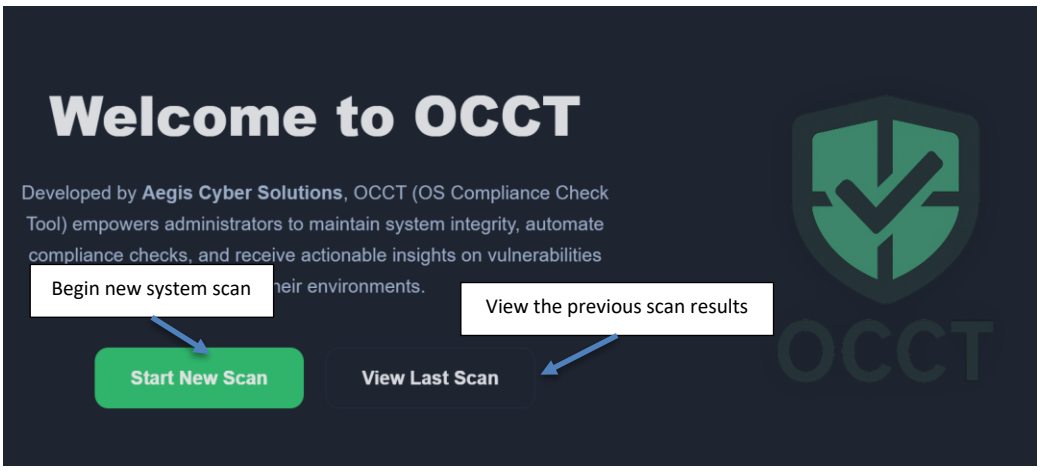


5. Getting Started (Quick Start)

5.1 Launching OCCT

```
.\env\Scripts\activate
python -m backend.app
```

After logging in you will be greeted with a login page with a prefilled username and password for proof of concept. This is followed by the landing page:

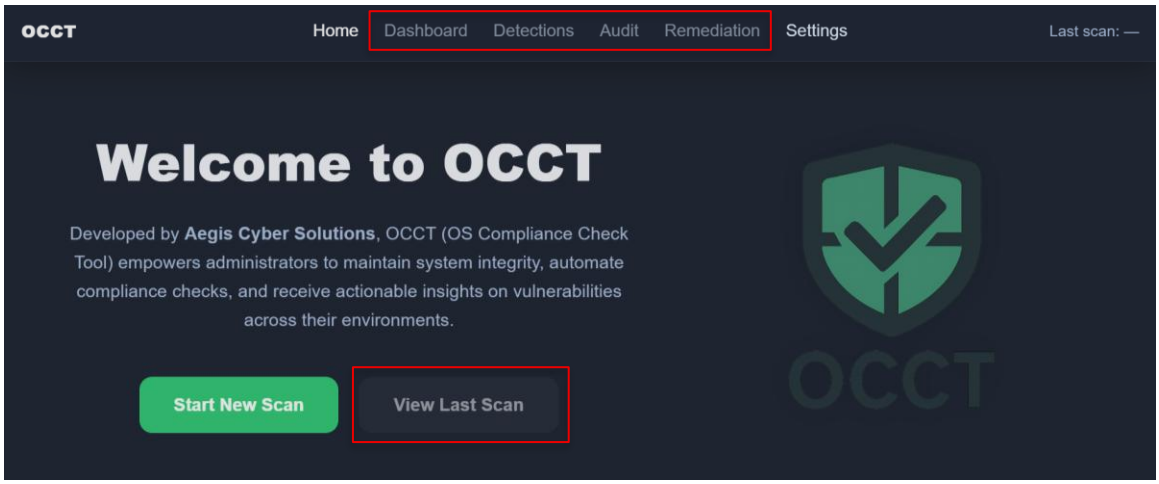


5.2 Running Your First Scan

- Login
- Click Start New Scan

- Review Dashboard → Audit → Remediation

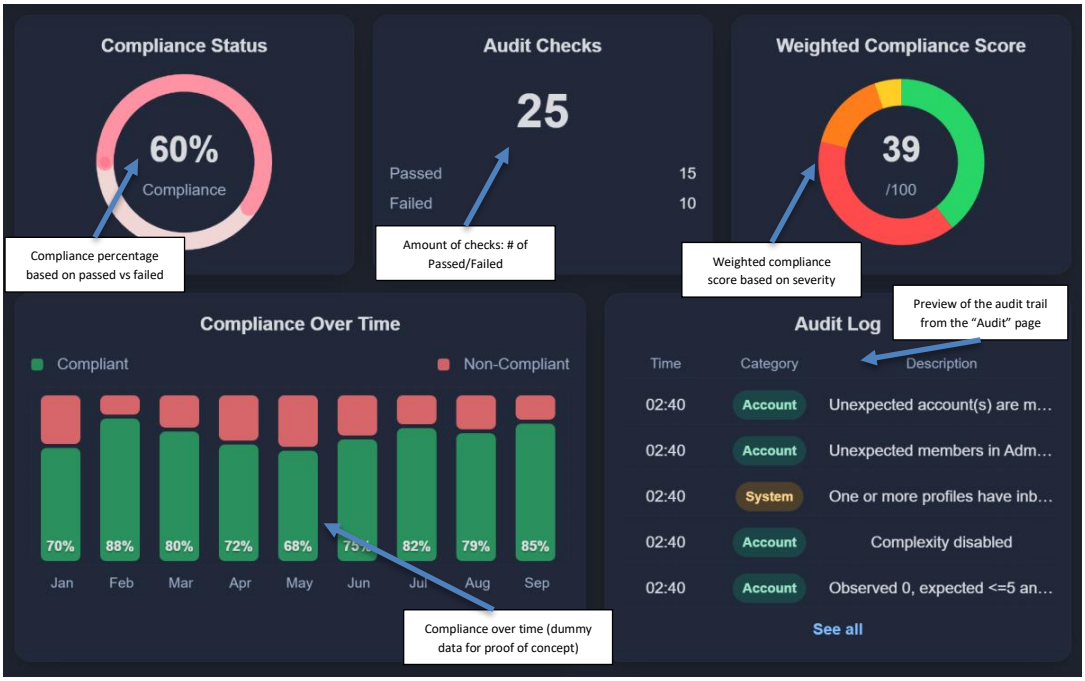
Buttons and data will be greyed out/unavailable until the first scan is run:



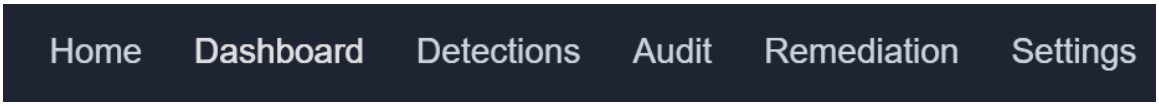
6. Using OCCT

6.1 Dashboard (Overview)

- Compliance Summary: pass/fail counts and severity distribution.
- Recent Findings: latest non-compliant or noteworthy events.
- Host & Mode: host label and Live/Sample indicator.

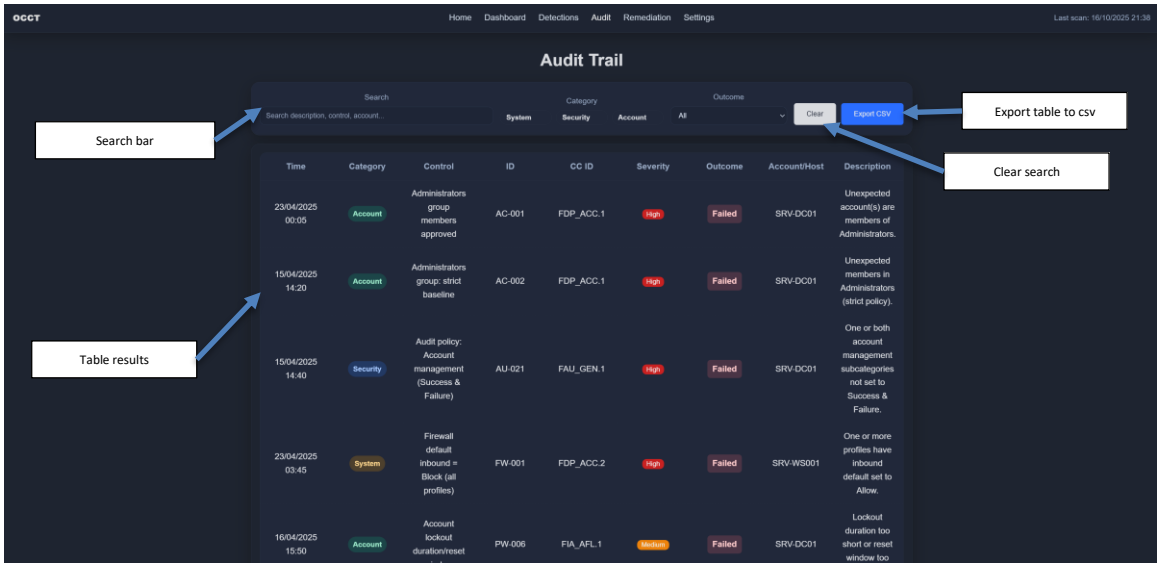


Navigate to the other pages using the navigation bar at the top of each page:

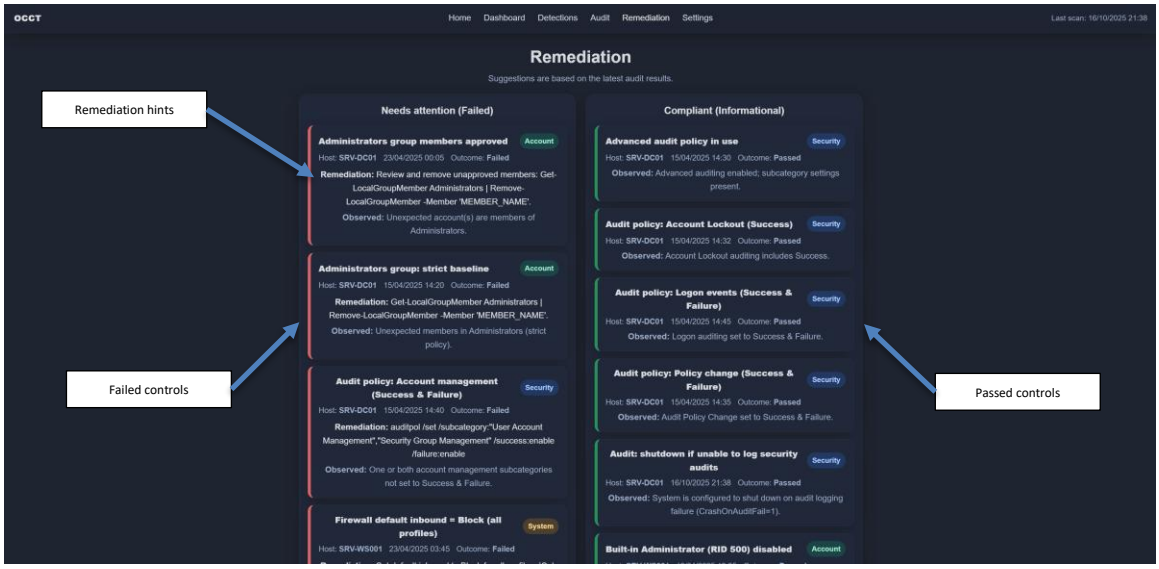


6.2 Controls & Compliance Views

Browse controls by category on the Audit page (Firewall, Account, etc.). Each control row shows Time, Category, Control, ID, CC ID, Severity, Outcome, Account/Host, Description:

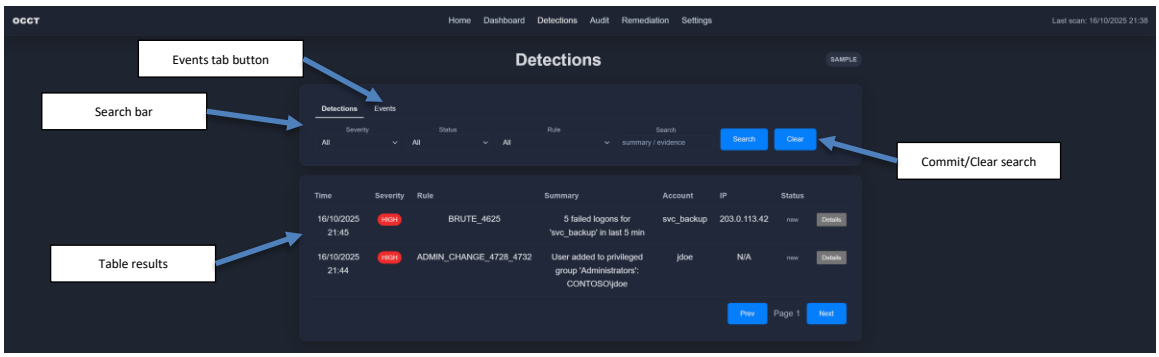


The Remediation page is a counterpart to the above, displaying all controls listed with their remediation hint if failed = true:

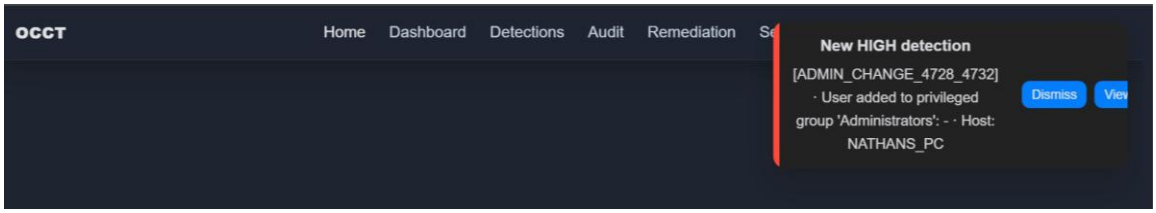


6.3 Account Activity & Admin Group Monitoring

View recent account events and Administrators group changes (adds/removes). Windows Security Event IDs like 4732/4728 indicate changes to local admin group membership.

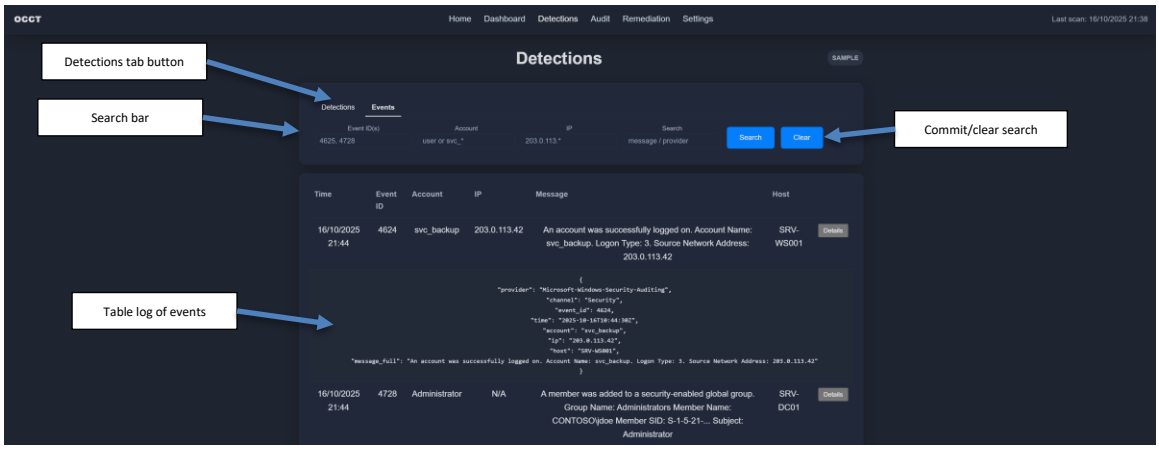


NOTE: When a detection event occurs, an alert will pop up in the top right corner on any page of the site:



6.4 Event & Evidence Views

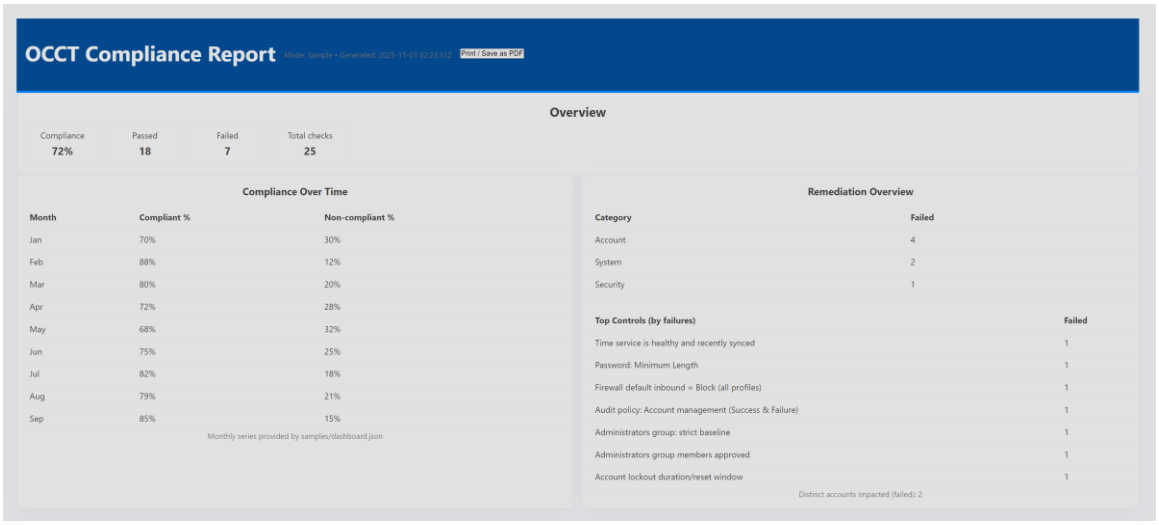
Inspect collected facts (registry/policies) and events (Security/System). Use filters (time, category, control) to locate relevant evidence.



6.5 Reports & Exports

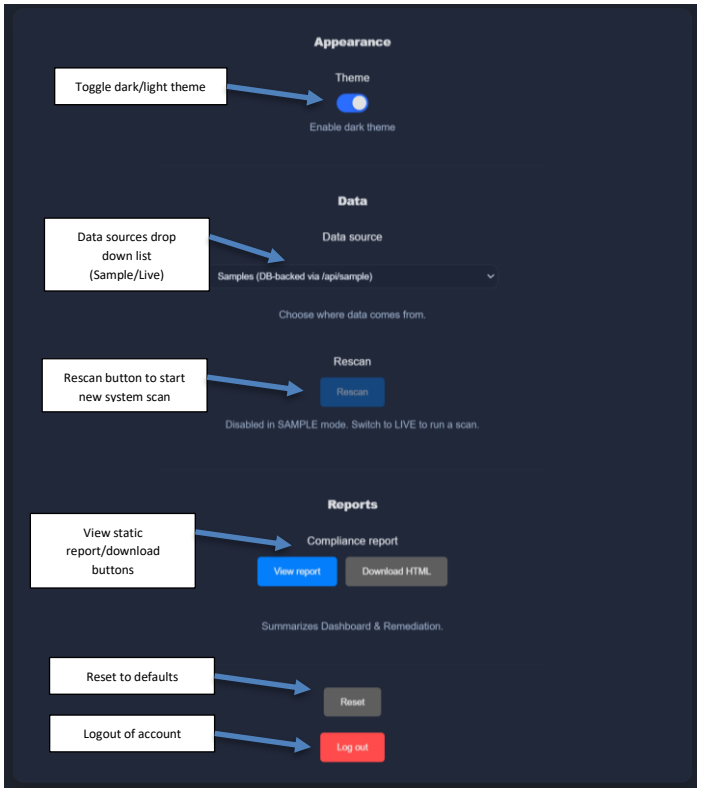
Export evidence lists for executive summaries in the following ways:

1. Export Audit Table to CSV on Audit Page
2. External Report Overview on Settings Page:



6.6 Settings

- Toggle Dark/Light Mode
- Switch Mode (Live/Sample)
- Rescan (Live)
- View Report (Opens external html formatted report)
- Download HTML (Downloads the report in HTML format)
- Reset (Resets the settings to default)
- Logout (Logs out of the application):



7. Controls Library & Framework Mapping

7.1 Control Card Template

- Control ID, Title, Category, Severity
- Framework Mapping (CC)
- Pass Condition
- Evidence fields and Remediation

```yaml
- id: PW-001
  title: "Password: Minimum Length"
  category: Account
  severity: medium
  cc_sfr: FMT_MTD.1
  when: "win.password.min_length >= 14"
  pass: "Observed {{win.password.min_length}}, expected >=14"
  fail: "Observed {{win.password.min_length}}, expected >=14"
  remediation: "Increase 'Minimum password length' to 14 or more: Local Security Policy → Account Policies → Password Policy."
```

7.2 Example Controls (Prototype)

FW-001 — Firewall default inbound = Block (all profiles)

- **Category**: Firewall | Severity: High
- **Observed**: One or more profiles have inbound default set to Allow.
- **Remediation** (PowerShell):

```
Get-NetFirewallProfile | Select-Object Name, DefaultInboundAction
Set-NetFirewallProfile -Profile Domain,Private,Public -DefaultInboundAction
Block
```

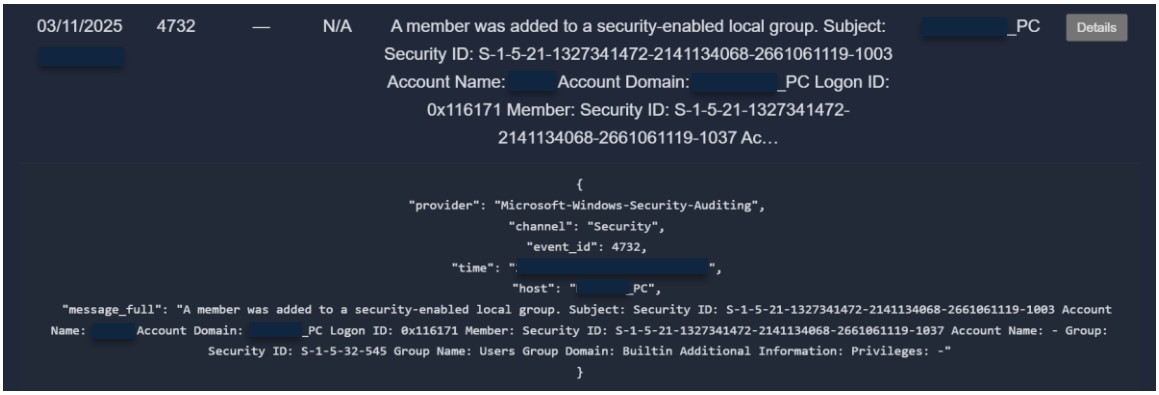
AC-002 — Complete access control (Local Administrators membership hygiene)

- **Category**: Account | Severity: High
- **Detection**: Unexpected members in Administrators
- **Remediation** (PowerShell):

```
Get-LocalGroupMember -Group "Administrators"
Remove-LocalGroupMember -Group "Administrators" -Member {username}  #
replace with actual account
```

8. Data Sources & Evidence

- **Windows Configuration**: firewall profiles, local groups, password policy
- **Windows Event Logs**: e.g., Security 4732/4728 (admin group changes), 4720 (account creation), 4625 (logon)
- **OCCT Facts & Events**: stored in SQLite with timestamps, category, control, outcome, account, description, source, host
- **Controls Library** (controls.yml): metadata, mapping, and detection logic.



9. Troubleshooting

Symptom	Likely Cause	Fix
"Running scripts is disabled on this system."	PowerShell execution policy	In the same PowerShell as Admin shell: Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
Controls page shows No Evidence	Scan not run (Live)	In Live Mode, return to Home or Settings, and click Run Scan
Live Poller/Events not working	Insufficient privileges to query local settings	Run OCCT with Administrator privileges
SQLite "database is locked"	Parallel runs or viewer open	Stop extra instances; close viewers; retry
Remediation hint shows placeholders literally	Placeholder not replaced	E.g. Replace {username} with real account name
Failed to install greenlet after "pip install -r requirements.txt"	Windows ARM machines are not compatible with the original requirements.txt	Use command: pip install -r requirements.winarm.txt

10. Security, Privacy & Known Limitations

- **Local Only**: Prototype runs locally; no data leaves the host.
- **Permissions**: Some checks require admin rights.
- **Scope**: Prototype controls cover key areas; not exhaustive CC coverage.
- **Integrations**: Email/SIEM integrations are not part of the prototype.