National College of Ireland

Configuration Manual

MSc Research Project
MSc Cybersecurity

Don Sunil
Student ID: x23317230

School of Computing
National College of Ireland

Supervisor: Mark Monaghan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Don Sunil

Student ID: x23317230

Programme: MSc Cybersecurity **Year:** 2024 - 2025

Module: MSc Research/Practicum

Lecturer: Mark Monaghan

Submission Due Date: 15/09/2025

Project Title: Security Analysis & Mitigation Strategies for Multi-Factor Authenticator Apps

Word Count: 860 **Page Count:** 5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Don Sunil

Date: 15/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□

You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on	□

computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Don Sunil
X23317230

Project Title: Security Analysis & Mitigation Strategies for Multi-Factor Authenticator Apps
Implementation Type: Lightweight MVP (No Emulator, Synthetic Data, Script-Based Mitigation)
Platform: Windows 10/11 + WSL (Ubuntu 22.04)
Storage Path: `D:\MFA-Security-Study`

1. Environment Setup

1.1 Hardware Requirements

- Host System: Windows 10 or 11
- Minimum 4 GB RAM (8 GB recommended)
- At least 5 GB free on `D:\` drive

1.2 Software Components

- Windows Subsystem for Linux (WSL) with Ubuntu 22.04
- Docker (with systemd enabled)
- Python 3.11 or above
- mitmproxy 10.x
- MobSF Docker image
- Required Python and system packages

2. Folder Structure

Create the following folder tree on the `D:` drive:

```
D:\MFA-Security-Study\
├── evidence\
│   ├── mitm\
│   ├── screenshots\
│   └── mobsf\
├── exports_raw\
├── exports_encrypted\
├── scripts\
├── tools\
└── MFA-Security-Study_MVP.zip
```

You may create these via Windows Explorer or WSL.

3. WSL Tool Installation

3.1 Open Ubuntu (WSL)

```
sudo apt update && sudo apt install -y \
  python3-pip \
  openssl \
  adb \
  docker.io \
  jq \
  wget \
  unzip \
  git \
  hexdump
```

3.2 Install Python Libraries

```
pip install mitmproxy pynacl argon2-cffi
```

3.3 Start Docker and Pull MobSF

```
sudo systemctl start docker
docker pull opensecurity/mobile-security-framework-mobsf:latest
```

4. Static Analysis Using MobSF

4.1 Launch MobSF

```
cd ~/MFA-Security-Study
mkdir -p tools/mobsf_data

docker run -d --name mobsf -p 8000:8000 \
  -v "$PWD/tools/mobsf_data:/home/mobsf/.MobSF" \
  opensecurity/mobile-security-framework-mobsf:latest
```

4.2 Analyze APK

1. Go to `http://localhost:8000`
2. Login with:
   - Username: `mobsf`
   - Password: `mobsf`
3. Upload the APK (e.g., `GoogleAuthenticator.apk`) from `tools/`
4. Download the **PDF report** and save it to:
   `evidence/mobsf/GA_QUICKSCAN.pdf`

5. Simulate and Encrypt Backup Files

5.1 Create Dummy TOTP Backup Files

```
cd ~/MFA-Security-Study/exports_raw
```

```
nano GA-1.json  # Paste Google Authenticator dummy JSON
nano MSA-1.json # Paste Microsoft Authenticator dummy JSON
nano Authy-1.json # Paste Authy dummy JSON
```

Each file should include fields like:

```
{
  "issuer": "TestApp",
  "account": "user@example.com",
  "secret": "JBSWY3DPEHPK3PXP",
  "algorithm": "SHA1",
  "digits": 6,
  "period": 30
}
```

5.2 Create **`secure_backup.py`** Script

```
cd ~/MFA-Security-Study/scripts
nano secure_backup.py
```

Paste the full encryption script using:

- Argon2id KDF (64 MiB, 2 threads)
- XChaCha20-Poly1305 AEAD

Make it executable:

```
chmod +x secure_backup.py
```

5.3 Run Encryption on All **`.json`** Files

```
cd ~/MFA-Security-Study

for f in exports_raw/*.json; do
  scripts/secure_backup.py "$f" "Strong Passphrase" | tee -a
exports_encrypted/SHA256.txt
done
```

Check `.sec` files and SHA-256 hashes.

6. Simulated Flow Capture and Leak Detection

6.1 Create **`flows.txt`** for Simulation

```
cd ~/MFA-Security-Study/evidence/mitm
nano flows.txt
```

Paste example TOTP leakage lines such as:

```
otpauth://totp/TestBank:user1@test.com?secret=JBSWY3DPEHPK3PXP
"secret":"HXDMVJECJJWSRB3HWIZR4IFUGFTMXBOZ"
{"account":"user3@social.com","secret":"GEZDGNBVGY3TQOJQGEZDGNBVGY3TQOJQ"}
```

6.2 Create **`parse_flows.sh`** Script

```
cd ~/MFA-Security-Study/scripts
nano parse_flows.sh
```

Paste the following:

```
#!/usr/bin/env bash
set -e
fl="../evidence/mitm/flows.txt"
out="../evidence/mitm/hits.txt"

grep -E -i "otpauth://|[A-Z2-7]{16,}|\"otp\"|\"secret\"" "$fl" \
  | tee "$out"
```

Make it executable:

```
chmod +x parse_flows.sh
```

6.3 Run Detection Script

```
cd ~/MFA-Security-Study/scripts
./parse_flows.sh
```

Check `hits.txt` for exposed secrets.

7. Final Documentation

7.1 **`results.csv`**

```
app,secret_observed,notes,enc_time_ms
Google Authenticator,Yes,otpauth URI seen in flows.txt,143
Microsoft Authenticator,Yes,base32 secret in JSON,162
Authy,Yes,full secret JSON in flow capture,238
```

Save in root folder as `results.csv`.

7.2 **`README.md`**

Include:

- Environment details
- Steps to reproduce
- Encryption parameters
- Leak detection methodology
- Ethical considerations
- Limitations

8. Final Packaging

From the root project directory:

```
cd ~/MFA-Security-Study
zip -r MFA-Security-Study_MVP.zip \
```

```
README.md \
results.csv \
evidence/ \
exports_raw/ \
exports_encrypted/ \
scripts/
```

You now have a complete submission-ready ZIP:

```
D:\MFA-Security-Study\MFA-Security-Study_MVP.zip
```

What This MVP Demonstrates

Capability	Outcome
Static Analysis	MobSF analysis of APK (PDF attached)
Backup Risk Replication	Simulated `.json` files mimic real ones
Leakage Detection	Simulated mitmproxy output parsed
Secure Backup Mitigation	Argon2id + XChaCha20 encryption
Performance Timing	Encryption time under 250 ms per file
Evidence Pack	All artifacts logged and archived

Ethical Controls

- All accounts and secrets are synthetic
- No actual apps were accessed during testing
- Certificate authorities were not permanently installed
- No ToS violations or privacy risks involved