IT Management & Audits

Practical Lab Manual

IT Security Assessment Platform

Practical P15 — CAPSTONE

Learning Domain

Comprehensive IT Security Assessment

Course Learning Outcomes

CLO15: Conduct end-to-end IT security assessments integrating infrastructure, application, and compliance testing

Unit

Unit IX: Capstone — Integrated Security Assessment

Time Allocation: 3 hours

Learning Mode: Hands-on (80%) + Theory (20%)

Difficulty Level: Advanced

IT Security Assessment Platform

Practical P15 — Capstone

Quick Reference

Practical Code	P15
Practical Name	IT Security Assessment Platform (Capstone)
Slot	T/P-15
Duration	3 hours
CLO Mapping	CLO15
Unit	Unit IX: Capstone — Integrated Security Assessment
Delivery Mode	Hands-on Lab
Target Audience	Advanced Level
India Integration	HIGH
Screenshot Count	5 Required

Prerequisites

- Completion of P01–P14 recommended (especially P04, P11, P14)
- Strong understanding of network security fundamentals (ports, protocols, scanning)
- Familiarity with web security concepts (HTTP headers, SSL/TLS, OWASP Top 10)
- Knowledge of compliance frameworks (ISO 27001, PCI-DSS, NIST CSF)
- Understanding of risk assessment methodologies and risk matrices
- Python 3.8+ installed with experience creating virtual environments

Tools Required

Tool	Version	Free	Notes
Python	3.8+	✓	Required
Flask	Latest	✓	Web dashboard framework
requests	Latest	✓	HTTP scanning library
PyYAML	Latest	✓	Configuration parsing
Click	Latest	✓	CLI framework
Rich	Latest	✓	Terminal formatting
Jinja2	Latest	✓	Report template engine
pip	Latest	✓	Included with Python
Web Browser	Latest	✓	Chrome or Firefox

Learning Objectives

- ✓ Conduct end-to-end IT security assessments integrating infrastructure, application, and compliance testing
- ✓ Perform network scanning to identify open ports, services, and potential vulnerabilities
- ✓ Execute web security assessments covering HTTP headers, SSL/TLS, cookies, and CORS policies
- ✓ Evaluate organizational compliance against ISO 27001, PCI-DSS, and NIST CSF frameworks
- ✓ Calculate risk scores using a weighted 5x5 risk matrix across multiple security categories
- ✓ Generate executive, technical, and remediation reports for different stakeholder audiences
- ✓ Operate a web-based security dashboard for real-time assessment monitoring
- ✓ Orchestrate automated assessment pipelines from scanning through reporting
- ✓ Apply Indian regulatory requirements (RBI, SEBI, CERT-In, DPDP Act) to security assessments

What You Will Learn

By the end of this capstone practical, you will:

1. Set up a comprehensive security assessment platform with scanning, compliance, risk, and reporting modules
2. Perform TCP port scanning with service detection and banner grabbing using Python sockets
3. Analyze web application security posture including headers, SSL/TLS, cookies, and CORS
4. Run compliance checks against ISO 27001, PCI-DSS, and NIST CSF control frameworks
5. Calculate organizational risk scores using weighted multi-category risk analysis
6. Generate three types of professional reports: executive summary, technical detail, and remediation roadmap
7. Launch and navigate a Flask-based security dashboard with interactive visualizations
8. Execute a fully orchestrated assessment pipeline that automates the entire workflow

Real-World Application

Comprehensive security assessments are the cornerstone of organizational cybersecurity maturity. In India, the Reserve Bank of India (RBI) mandates that all regulated entities conduct regular security assessments covering infrastructure, applications, and compliance. SEBI requires market intermediaries to undergo VAPT through CERT-In empanelled auditors. The Digital Personal Data Protection (DPDP) Act, 2023 imposes security obligations on data fiduciaries. Major consulting firms — Deloitte, PwC, EY, and KPMG — conduct integrated security assessments for Indian banks, combining network scanning, web application testing, compliance audits, and risk quantification into unified engagement reports. Organizations like NPCI (National Payments Corporation of India) require their member banks to demonstrate security assessment coverage across all layers before granting access to critical payment infrastructure like UPI and IMPS.

Hands-On Procedure

Capstone Integration Notice: This practical integrates concepts from all previous practicals (P01–P14). It brings together network scanning (P03), web security analysis (P04), compliance checking (P11), and risk assessment (P14) into a unified security assessment platform. All scanning and testing must be performed **strictly on localhost** (127.0.0.1). Scanning any external system without explicit written authorization is **illegal** under the Information Technology Act, 2000 (Sections 43 and 66). By proceeding, you agree to use these tools responsibly and ethically.

Part A: Platform Setup and Configuration

Step 1: Clone the Security Assessment Platform Repository

Objective: Download and explore the integrated security assessment platform project structure.
Instructions:

1. Open your terminal (bash on Linux/Mac, PowerShell on Windows)
2. Navigate to your working directory
3. Clone the security assessment platform repository
4. Explore the comprehensive project structure across all modules

Code/Command:

```
# Clone the repository
git clone <repository-url> security-assessment-platform
cd security-assessment-platform

# View the top-level project structure
ls -la
# Expected: src/, dashboard/, config/, results/, requirements.txt

# Explore the scanner modules
ls src/scanner/
# Expected: network_scanner.py, web_scanner.py, dns_scanner.py

# Explore the compliance modules
ls src/compliance/
# Expected: iso27001.py, pci_dss.py, nist_csf.py

# Explore the risk analysis modules
ls src/risk/
# Expected: risk_engine.py, risk_matrix.py

# Explore the reporting modules
```

```
ls src/reporting/
# Expected: executive_report.py, technical_report.py,
#           remediation_roadmap.py, templates/

# Explore the web dashboard
ls dashboard/
# Expected: app.py, templates/, static/

# View the CLI entry point and orchestrator
ls src/cli.py src/orchestrator.py

# Explore configuration files
ls config/
# Expected: targets.yaml, thresholds.yaml
```

Clone and Explore the Platform Repository

Expected Output

Project directory contains the following modules:
`src/scanner/` — Network, web, and DNS scanning engines
`src/compliance/` — ISO 27001, PCI-DSS, NIST CSF checkers
`src/risk/` — Risk calculation engine and 5x5 matrix
`src/reporting/` — Executive, technical, and roadmap report generators
`src/reporting/templates/` — Jinja2 HTML report templates
`dashboard/` — Flask web dashboard application
`dashboard/templates/` — Dashboard HTML templates
`dashboard/static/` — CSS, JavaScript, and assets
`src/cli.py` — Click-based CLI with all commands
`src/orchestrator.py` — Automated end-to-end pipeline
`config/` — YAML configuration files for targets and thresholds
`results/` — Output directory for scan results and reports

This capstone platform is organized in a modular architecture. Each module (scanner, compliance, risk, reporting) operates independently but is orchestrated together through the CLI and the orchestrator. This mirrors how enterprise security tools like Qualys, Tenable, and Rapid7 are structured.

Step 2: Create Virtual Environment, Install Dependencies, and Configure Targets

Objective: Set up the Python environment, install all required packages, configure scan targets, and verify the CLI.
Instructions:

1. Create and activate a Python virtual environment
2. Install all project dependencies from `requirements.txt`

3. Configure `targets.yaml` with localhost as the primary target
4. Verify all CLI commands are accessible

Code/Command:

```
# Create virtual environment
python -m venv venv

# Activate (Linux/Mac)
source venv/bin/activate

# Activate (Windows PowerShell)
.\venv\Scripts\Activate.ps1

# Install dependencies
pip install -r requirements.txt
# Installs: Flask, requests, PyYAML, Click, Rich, Jinja2,
#           and other dependencies

# Configure targets.yaml with localhost
cat config/targets.yaml
# Expected content:
# targets:
#   primary:
#     host: 127.0.0.1
#     ports: [80, 443, 8080, 5000, 22, 21, 3306, 5432]
#     web_url: http://localhost:5000
#   scan_options:
#     timeout: 5
#     threads: 10
#     verbose: true

# Verify CLI - list all available commands
python src/cli.py --help
# Expected commands:
#   scan-network      Run network infrastructure scan
#   scan-web          Run web security assessment
#   check-compliance  Run compliance framework checks
#   calculate-risk    Calculate organizational risk score
#   generate-report   Generate assessment reports
#   start-dashboard   Launch the web dashboard
```

Environment Setup and Configuration

Expected Output

```
Usage:  cli.py [OPTIONS] COMMAND [ARGS]...

IT Security Assessment Platform - Capstone CLI

Options:
-help  Show this message and exit.

Commands:
scan-network      Run network infrastructure scan
scan-web          Run web security assessment
check-compliance  Run compliance framework checks
calculate-risk    Calculate organizational risk score
generate-report   Generate assessment reports
start-dashboard   Launch the web dashboard
```

If Rich fails to install: (1) Ensure Python 3.8+ is installed: `python -version`, (2) Upgrade pip: `python -m pip install -upgrade pip`, (3) Install Rich separately: `pip install rich`, (4) On older systems, try: `pip install rich==12.0.0`.

Part B: Infrastructure Scanning

Step 3: Run Network Infrastructure Scan

Objective: Perform TCP port scanning with service detection and banner grabbing to map the target infrastructure.
Instructions:

1. Run the network scan command against localhost using the targets configuration
2. Observe the TCP port scanning process using Python sockets
3. Review detected open ports and identified services
4. Note the service banner grabbing results and OS fingerprinting hints

Code/Command:

```
# Run network scan with configuration file
python src/cli.py scan-network --config config/targets.yaml

# The scanner performs:
# 1. TCP connect scan on configured port range
# 2. Service banner grabbing on open ports
# 3. OS fingerprinting based on TCP/IP stack behavior
# 4. Service version detection from banner strings
# 5. Results stored for risk calculation
```

Network Infrastructure Scan

Expected Output

```
======================================================
NETWORK INFRASTRUCTURE SCAN
Target:  127.0.0.1
======================================================
Scanning 8 ports on 127.0.0.1...

PORT      STATE    SERVICE        VERSION
---                            ------
22/tcp    open     ssh            OpenSSH 8.9
80/tcp    closed   http           --
443/tcp   closed   https          --
3306/tcp  open     mysql          MySQL 8.0.32
5000/tcp  open     http           Python/Flask
5432/tcp  closed   postgresql     --
8080/tcp  closed   http-proxy     --

Open Ports:  3/8
OS Hint:  Linux (TTL=64)
Scan Duration:  4.2 seconds
======================================================
[MEDIUM] Port 22 open - SSH accessible
[INFO]   Port 3306 open - Database exposed
[INFO]   Port 5000 open - Web service running
======================================================
```

Port scanning even on localhost may trigger security software alerts. If your antivirus or firewall blocks the scan, temporarily add an exception for the Python process. The number of open ports on your system may differ from the expected output above — this is normal and depends on your local configuration.

Step 4: Run Web Security Assessment

Objective: Analyze web application security posture including HTTP headers, SSL/TLS configuration, cookie security, and CORS policies.
Instructions:

1. Run the web security scan against the local Flask application
2. Review HTTP security headers analysis (present vs. missing)
3. Examine SSL/TLS configuration findings
4. Check cookie security flags and CORS policy results
5. Note each finding's severity classification

Code/Command:

```
# Run web security scan
python src/cli.py scan-web --target http://localhost:5000

# The scanner checks:
# 1. HTTP security headers (CSP, HSTS, X-Frame-Options, etc.)
# 2. SSL/TLS configuration (certificate, protocol versions)
# 3. Cookie security flags (Secure, HttpOnly, SameSite)
# 4. CORS policy (Access-Control-Allow-Origin)
# 5. Server information disclosure
# 6. HTTP methods allowed (OPTIONS check)
```

Web Security Assessment

Expected Output

```
======================================================
WEB SECURITY ASSESSMENT
Target:  http://localhost:5000
======================================================

[SECURITY HEADERS]
Content-Security-Policy:        MISSING [HIGH]
Strict-Transport-Security:      MISSING [HIGH]
X-Frame-Options:                MISSING [HIGH]
X-Content-Type-Options:         MISSING [MEDIUM]
Referrer-Policy:                MISSING [MEDIUM]
Permissions-Policy:             MISSING [LOW]
Headers Score:  0/6 (0%)

[SSL/TLS ANALYSIS]
HTTPS Enabled:                  NO [HIGH]
Note:  Running on plain HTTP

[COOKIE SECURITY]
session cookie:
Secure Flag:                   MISSING [HIGH]
HttpOnly Flag:                 PRESENT [OK]
SameSite Attribute:            MISSING [MEDIUM]

[CORS POLICY]
Access-Control-Allow-Origin:    * (Wildcard) [HIGH]
CORS Misconfiguration Detected

[SERVER DISCLOSURE]
Server Header:  Werkzeug/2.x.x Python/3.x [LOW]
======================================================
Total Findings:  11
Critical:  0 | High:  5 | Medium:  3 | Low:  3
======================================================
```

In production environments, every web application should have all six security headers configured, enforce HTTPS with valid certificates, set Secure/HttpOnly/SameSite on all cookies, and restrict CORS to specific trusted origins. A wildcard CORS policy (*) is one of the most common and dangerous misconfigurations.

Screenshot 1

What to paste: Terminal output showing the network infrastructure scan results, including the port scan table with open ports, detected services, service versions, and OS fingerprinting hint from Step 3.

Paste your screenshot here

Screenshot 2

What to paste: Terminal output showing the web security assessment results from Step 4, including HTTP security headers analysis, SSL/TLS status, cookie security flags, and CORS policy findings with severity ratings.

Paste your screenshot here

Part C: Compliance Assessment

Step 5: Run ISO 27001 Compliance Check

Objective: Evaluate organizational security posture against a subset of 20 key ISO 27001 controls.
Instructions:

1. Run the compliance check command with the ISO 27001 framework
2. Review each control's status: Compliant, Non-Compliant, or Not Tested
3. Note the overall compliance percentage and gap analysis
4. Identify which control domains have the most gaps

Code/Command:

```
# Run ISO 27001 compliance assessment
python src/cli.py check-compliance --framework iso27001

# The checker evaluates 20 key controls from ISO 27001:
# A.5  - Information Security Policies
# A.6  - Organization of Information Security
# A.7  - Human Resource Security
# A.8  - Asset Management
# A.9  - Access Control
# A.10 - Cryptography
# A.11 - Physical Security
# A.12 - Operations Security
# A.13 - Communications Security
# A.14 - System Acquisition & Development
# A.16 - Incident Management
# A.18 - Compliance
```

ISO 27001 Compliance Check

Expected Output

```
=====================================================
ISO 27001 COMPLIANCE ASSESSMENT
=====================================================

CONTROL   DESCRIPTION                    STATUS
----   ------                     ---
A.5.1.1  Info Security Policy            COMPLIANT
A.5.1.2  Review of Policies              NON-COMPLIANT
A.6.1.1  Security Roles/Responsibilities COMPLIANT
A.6.1.2  Segregation of Duties           NON-COMPLIANT
A.8.1.1  Inventory of Assets             COMPLIANT
A.8.2.1  Classification of Info          NON-COMPLIANT
A.9.1.1  Access Control Policy           COMPLIANT
A.9.2.3  Privileged Access Management    NON-COMPLIANT
A.9.4.1  Information Access Restriction  COMPLIANT
A.10.1.1 Cryptographic Controls          NON-COMPLIANT
A.10.1.2 Key Management                  NOT TESTED
A.12.1.1 Documented Procedures           COMPLIANT
A.12.2.1 Malware Controls                 COMPLIANT
A.12.4.1 Event Logging                    NON-COMPLIANT
A.12.6.1 Vulnerability Management        NON-COMPLIANT
A.13.1.1 Network Controls                 COMPLIANT
A.14.1.2 Securing App Services           NON-COMPLIANT
A.14.2.1 Secure Development Policy       NOT TESTED
A.16.1.1 Incident Management             NON-COMPLIANT
A.18.1.1 Applicable Legislation          COMPLIANT

=====================================================
Compliant:  9/20 (45%)  |  Non-Compliant:  9/20 (45%)
Not Tested:  2/20 (10%)
Overall Status:  BELOW THRESHOLD (Target:  80%)
=====================================================
```

Step 6: Run PCI-DSS and NIST CSF Compliance Checks

Objective: Assess compliance against PCI-DSS and NIST CSF frameworks and compare coverage across all three frameworks.
Instructions:

1. Run the PCI-DSS compliance check
2. Run the NIST CSF compliance check
3. Compare compliance percentages across ISO 27001, PCI-DSS, and NIST CSF
4. Identify overlapping controls and unique gaps in each framework

Code/Command:

```
# Run PCI-DSS compliance check
python src/cli.py check-compliance --framework pci-dss

# Run NIST CSF compliance check
python src/cli.py check-compliance --framework nist-csf

# PCI-DSS checks 12 core requirements:
# Req 1-2:  Network Security
# Req 3-4:  Data Protection
# Req 5-6:  Vulnerability Management
# Req 7-9:  Access Control
# Req 10:   Monitoring & Logging
# Req 11:   Security Testing
# Req 12:   Security Policies

# NIST CSF checks 5 core functions:
# Identify (ID), Protect (PR), Detect (DE),
# Respond (RS), Recover (RC)
```

PCI-DSS and NIST CSF Compliance Checks

Expected Output

```
======================================================
PCI-DSS COMPLIANCE ASSESSMENT
======================================================
Req 1:  Firewall Configuration       COMPLIANT
Req 2:  Default Passwords            NON-COMPLIANT
Req 3:  Stored Cardholder Data       NON-COMPLIANT
Req 4:  Encryption in Transit        NON-COMPLIANT
Req 5:  Anti-Malware                  COMPLIANT
Req 6:  Secure Development            NON-COMPLIANT
Req 7:  Access Restriction            COMPLIANT
Req 8:  Authentication                COMPLIANT
Req 9:  Physical Access               NOT TESTED
Req 10:  Logging and Monitoring        NON-COMPLIANT
Req 11:  Security Testing              NON-COMPLIANT
Req 12:  Security Policy               COMPLIANT
Compliant:  5/12 (42%)
======================================================

======================================================
NIST CSF COMPLIANCE ASSESSMENT
======================================================
ID.AM   Asset Management             COMPLIANT
ID.RA   Risk Assessment              NON-COMPLIANT
PR.AC   Access Control               COMPLIANT
PR.DS   Data Security                NON-COMPLIANT
PR.IP   Protective Processes         NON-COMPLIANT
PR.MA   Maintenance                  COMPLIANT
DE.AE   Anomaly Detection            NON-COMPLIANT
DE.CM   Continuous Monitoring        NON-COMPLIANT
RS.RP   Response Planning            NON-COMPLIANT
RS.CO   Response Communications      COMPLIANT
RC.RP   Recovery Planning            NON-COMPLIANT
RC.IM   Recovery Improvements        NOT TESTED
Compliant:  4/12 (33%)
======================================================

CROSS-FRAMEWORK COMPARISON:
ISO 27001:  45% | PCI-DSS: 42% | NIST CSF: 33%
Weakest Area:  Detection & Response capabilities
======================================================
```

In practice, organizations often map controls across multiple frameworks to avoid duplication. For example, ISO 27001 A.9 (Access Control) maps directly to PCI-DSS Requirements 7–8 and NIST CSF PR.AC. A unified control framework reduces audit fatigue and ensures consistent implementation.

Screenshot 3

What to paste: Terminal output showing the ISO 27001 compliance check results from Step 5, including the full control status table with Compliant, Non-Compliant, and Not Tested entries and the overall compliance percentage.

Paste your screenshot here

Part D: Risk Analysis

Step 7: Calculate Organizational Risk Score

Objective: Compute a weighted organizational risk score across six security categories using a 5x5 risk matrix.
Instructions:

1. Run the risk calculation command to aggregate all assessment data
2. Review the six risk categories: Network, Application, Data, Access, Compliance, Operational
3. Examine the 5x5 risk matrix with color-coded likelihood vs. impact ratings
4. Note the overall weighted risk score and risk rating classification
5. Identify the highest-risk categories requiring immediate attention

Code/Command:

```
# Calculate organizational risk score
python src/cli.py calculate-risk

# The risk engine:
# 1. Aggregates findings from network and web scans
# 2. Incorporates compliance gap data
# 3. Applies weighted scoring across 6 categories
# 4. Generates a 5x5 risk matrix (likelihood x impact)
# 5. Calculates overall organizational risk rating

# Risk Categories and Weights:
# Network Security:    20%
# Application Security: 25%
# Data Protection:     20%
# Access Control:      15%
# Compliance:          10%
# Operational:         10%
```

Risk Score Calculation

Expected Output

```
======================================================
ORGANIZATIONAL RISK ASSESSMENT
======================================================

RISK CATEGORY SCORES (1-25 scale):
Category              Score   Rating    Weight
----------   ---   ---    ---
Network Security      12      MEDIUM    20%
Application Security  20      HIGH      25%
Data Protection       16      HIGH      20%
Access Control        9       MEDIUM    15%
Compliance            15      HIGH      10%
Operational           8       LOW       10%

5x5 RISK MATRIX (Likelihood x Impact):
| Negligible| Minor | Moderate| Major|Critical
Almost Cert.|    5     |  10   |   15    |  20  |  25
Likely      |    4     |   8   |  [12]   |  16  |  20
Possible    |    3     |   6   |    9    | [12] |  15
Unlikely    |    2     |   4   |    6    |   8  |  10
Rare        |    1     |   2   |    3    |   4  |   5

Color Key:  [1-4] LOW  [5-9] MEDIUM
[10-15] HIGH  [16-25] CRITICAL

WEIGHTED RISK SCORE: 14.15 / 25 (56.6%)
OVERALL RISK RATING: HIGH
Highest Risk:  Application Security (20/25)
======================================================
```

A risk rating of HIGH indicates significant security gaps that require a structured remediation program. In Indian regulatory context, RBI would classify this as requiring a time-bound corrective action plan with Board-level oversight. SEBI-regulated entities with this risk level would need to engage CERT-In empanelled auditors for immediate remediation verification.

Screenshot 4

What to paste: Terminal output showing the organizational risk assessment from Step 7, including the risk category scores table, the 5x5 risk matrix (likelihood vs. impact) with color-coded entries, and the overall weighted risk score and rating.

Paste your screenshot here

Part E: Reporting and Dashboard

Step 8: Generate All Three Report Types

Objective: Generate executive summary, technical detail, and remediation roadmap reports for different stakeholder audiences.
Instructions:

1. Generate the executive summary report (1-page overview for leadership)
2. Generate the technical report (detailed findings with evidence and CVSS-like scores)
3. Generate the remediation roadmap (prioritized actions: Quick Wins, Short-term, Long-term)
4. Open each report in a web browser and review its structure and content

Code/Command:

```
# Generate Executive Summary Report
python src/cli.py generate-report \
    --type executive \
    --output exec_report.html
# Contents: Overall risk score, top 5 critical findings,
#           key recommendations, compliance summary

# Generate Technical Report
python src/cli.py generate-report \
    --type technical \
    --output tech_report.html
# Contents: Detailed findings with evidence, CVSS-like
#           scores, affected assets, proof-of-concept data,
#           technical remediation steps

# Generate Remediation Roadmap
python src/cli.py generate-report \
    --type roadmap \
    --output roadmap.html
# Contents: Prioritized actions organized by timeline:
#   Quick Wins (0-30 days) - Low effort, high impact
#   Short-term (30-90 days) - Medium effort fixes
#   Long-term (90-180 days) - Strategic improvements

# Open reports in browser (Windows)
start exec_report.html
start tech_report.html
start roadmap.html

# Open reports in browser (Linux/Mac)
open exec_report.html
open tech_report.html
open roadmap.html
```

Generate Assessment Reports

Expected Output

```
[+] Executive report generated: exec_report.html
Risk Score: 14.15/25 (HIGH)
Top 5 Findings: Application Security (Critical),
Data Protection (High), Compliance Gaps (High),
Network Exposure (Medium), Missing Headers (Medium)
Recommendations: 5 strategic actions listed

[+] Technical report generated: tech_report.html
Total Findings: 23
Critical: 2 | High: 8 | Medium: 7 | Low: 6
Each finding includes CVSS score and evidence

[+] Remediation roadmap generated: roadmap.html
Quick Wins (0-30 days): 6 actions
Short-term (30-90 days): 8 actions
Long-term (90-180 days): 5 actions
Estimated effort: 320 person-hours total
```

Different reports serve different audiences. The **executive summary** is for CISOs and Board members who need a quick risk overview. The **technical report** is for security engineers who need detailed remediation instructions. The **remediation roadmap** is for project managers who need to plan and track remediation activities across teams.

Step 9: Launch Web Dashboard and Navigate Assessment Views

Objective: Start the Flask-based web dashboard and explore the interactive assessment views.

Instructions:

1. Start the web dashboard using the CLI command
2. Open `http://localhost:8080` in your web browser
3. Navigate the dashboard pages: Risk Overview, Findings List, Compliance Status, Remediation Tracker
4. Use filters on the Findings List to sort by severity and category
5. Review the compliance status breakdown by framework

Code/Command:

```
# Start the web dashboard
python src/cli.py start-dashboard

# Expected output:
```

```
# [+] Loading assessment data...
# [+] Dashboard starting on http://localhost:8080
# * Running on http://127.0.0.1:8080
# * Debug mode: off

# Open browser and navigate to:
# http://localhost:8080

# Dashboard pages:
# /               - Risk Overview (gauges, charts, summary)
# /findings       - Findings List (filterable, sortable)
# /compliance     - Compliance Status (by framework)
# /remediation    - Remediation Tracker (progress bars)
```

Launch Web Dashboard

Expected Output

Browser at `http://localhost:8080` shows:
▷ **Risk Overview:** Organizational risk gauge (14.15/25), risk category bar chart, trend indicators, top 5 critical findings summary
▷ **Findings List:** Sortable table of all 23 findings with columns for Severity, Category, Description, CVSS Score, and Status. Filter dropdowns for severity level and category
▷ **Compliance Status:** Three-panel view showing ISO 27001 (45%), PCI-DSS (42%), NIST CSF (33%) with per-control status indicators (green/red/gray)
▷ **Remediation Tracker:** Progress bars for Quick Wins (0% started), Short-term (0% started), Long-term (0% started) with individual action item checklists

Enterprise security platforms like Qualys, Tenable.io, and Rapid7 InsightVM provide similar dashboard views. The ability to filter findings by severity and track remediation progress is essential for security operations teams managing hundreds of findings across large organizations.

Step 10: Run Full Orchestrated Assessment Pipeline

Objective: Execute the complete automated assessment pipeline that performs all phases from scanning through report generation in a single orchestrated run.
Instructions:

1. Run the orchestrator with the configuration file and output directory
2. Observe the automated pipeline execution: scan → analyze → compliance → risk → report
3. Review the complete output in the `results/` directory
4. Verify that all reports, scan data, and risk assessments are generated

Code/Command:

```
# Run the complete orchestrated assessment
python src/orchestrator.py \
    --config config/targets.yaml \
    --output results/

# The orchestrator automates the full pipeline:
# Phase 1: Network Infrastructure Scan
# Phase 2: Web Security Assessment
# Phase 3: Compliance Checks (all frameworks)
# Phase 4: Risk Score Calculation
# Phase 5: Report Generation (all types)
# Phase 6: Results Aggregation and Export

# Review generated outputs
ls results/
# Expected:
#   network_scan.json
#   web_scan.json
#   compliance_iso27001.json
#   compliance_pci_dss.json
#   compliance_nist_csf.json
#   risk_assessment.json
#   exec_report.html
#   tech_report.html
#   roadmap.html
#   assessment_summary.json
```

Full Orchestrated Assessment

Expected Output

```
======================================================
ORCHESTRATED SECURITY ASSESSMENT
Configuration:  config/targets.yaml
======================================================

[Phase 1/6] Network Scan.............COMPLETE (4.2s)
3 open ports detected, 2 findings

[Phase 2/6] Web Assessment...........COMPLETE (6.8s)
11 findings (5 HIGH, 3 MEDIUM, 3 LOW)

[Phase 3/6] Compliance Checks.........COMPLETE (3.1s)
ISO 27001:  45% | PCI-DSS: 42% | NIST CSF: 33%

[Phase 4/6] Risk Calculation..........COMPLETE (1.2s)
Overall Risk:  14.15/25 (HIGH)

[Phase 5/6] Report Generation.........COMPLETE (2.4s)
3 reports generated (exec, tech, roadmap)

[Phase 6/6] Results Export............COMPLETE (0.8s)
All data exported to results/

======================================================
ASSESSMENT COMPLETE
Total Duration:  18.5 seconds
Output Directory:  results/
Files Generated:  10
Overall Risk Rating:  HIGH
======================================================
```

The orchestrator represents how real-world security assessment platforms operate. Tools like Qualys, Nessus, and OpenVAS follow similar automated pipelines. In Indian banking, RBI mandates that VAPT findings be integrated into a unified risk dashboard and reported to the Board's IT Strategy Committee quarterly.

Screenshot 5

What to paste: The web dashboard (`http://localhost:8080`) open in your browser, showing the Risk Overview page with the organizational risk gauge, risk category bar chart, and top findings summary, OR the orchestrated assessment pipeline output showing all six phases completing successfully.

Paste your screenshot here

Conceptual Background

Security Assessment Methodology

A comprehensive security assessment follows a structured methodology with five distinct phases:

1. **Planning and Scoping:** Define assessment objectives, target systems, rules of engagement, timeline, and deliverables. Obtain written authorization and establish communication channels.
2. **Discovery and Reconnaissance:** Identify target assets through network scanning, port enumeration, service detection, and DNS analysis. Build a comprehensive asset inventory.
3. **Vulnerability Assessment:** Test identified assets for security weaknesses including misconfigurations, missing patches, insecure protocols, and application-layer vulnerabilities.
4. **Risk Analysis:** Quantify and qualify discovered vulnerabilities based on likelihood of exploitation and potential business impact. Prioritize findings using risk matrices.
5. **Reporting and Remediation:** Document findings in appropriate formats for different audiences, provide actionable remediation guidance, and track remediation progress.

Defense in Depth

Defense in depth is a security strategy that employs multiple layers of protection so that if one layer fails, others remain to contain the threat:

Layer	Controls	Assessment Focus
Network	Firewalls, IDS/IPS, segmentation, VPN	Port scanning, network policy review
Host	OS hardening, patching, endpoint protection	Configuration audit, patch status
Application	Secure coding, input validation, WAF	Web vulnerability scanning, code review
Data	Encryption, DLP, access controls, backup	Data classification, encryption verification

Compliance Framework Integration

ISO 27001, PCI-DSS, and NIST CSF are complementary frameworks that overlap significantly but serve different purposes:

- **ISO 27001** provides a comprehensive Information Security Management System (ISMS) framework. It is certification-based and widely recognized internationally. Its 114 controls (Annex A) cover organizational, people, physical, and technological aspects.

- **PCI-DSS** is specific to organizations that process, store, or transmit payment card data. Its 12 requirements are prescriptive and technically detailed, with specific validation procedures.
- **NIST CSF** organizes security activities into five core functions (Identify, Protect, Detect, Respond, Recover) and is designed as a flexible, risk-based framework that can be adopted by any organization.

Overlap example: Access control requirements appear in ISO 27001 Annex A.9, PCI-DSS Requirements 7–8, and NIST CSF PR.AC. By implementing access control once using a unified approach, organizations achieve compliance across all three frameworks simultaneously.

Risk Management Process

Effective risk management follows a continuous cycle:

1. **Identify:** Catalog assets, threats, and vulnerabilities that could impact the organization
2. **Assess:** Evaluate likelihood of threat exploitation and potential business impact using qualitative or quantitative methods
3. **Treat:** Select risk treatment options — mitigate (reduce), transfer (insure), accept (acknowledge), or avoid (eliminate the activity)
4. **Monitor:** Continuously track risk levels, reassess after changes, and report to stakeholders

Risk Quantification

Risk can be quantified using several approaches:

- **Qualitative 5x5 Matrix:** Plots likelihood (1–5) against impact (1–5) to produce a risk score (1–25). Categories: Low (1–4), Medium (5–9), High (10–15), Critical (16–25). This is the most common approach for initial assessments.
- **Semi-Quantitative Scoring:** Assigns numerical weights to risk categories and calculates weighted averages. Provides more granularity than pure qualitative assessment while avoiding the data requirements of full quantitative analysis.
- **Residual Risk:** After applying controls, the remaining risk is called residual risk. It is calculated as: Residual Risk = Inherent Risk − Control Effectiveness. Organizations must ensure residual risk falls within their defined risk appetite.

Report Types for Different Audiences

Report Type	Audience	Content Focus	Length
Executive Summary	CISO, Board, CxO	Risk score, top findings, strategic recommendations	1–2 pages
Technical Report	Security engineers, developers	Detailed findings, evidence, CVSS scores, remediation steps	20–50+ pages
Remediation Roadmap	Project managers, operations	Prioritized actions, timelines, effort estimates, dependencies	5–10 pages

Remediation Prioritization

Effective remediation prioritization considers multiple factors:

- **Risk-Based:** Address highest-risk findings first (Critical → High → Medium → Low)
- **Effort-Based:** Consider implementation complexity and resource requirements
- **Quick Wins:** Low-effort, high-impact fixes that demonstrate immediate improvement (e.g., adding security headers, updating default passwords)
- **Dependencies:** Some fixes depend on others (e.g., implementing encryption requires key management infrastructure first)

India-Specific Regulatory Context

RBI Cybersecurity Framework

The Reserve Bank of India requires all regulated entities to:

- Establish a Board-approved Cybersecurity Policy reviewed annually
- Conduct VAPT at least annually and after major changes, using CERT-In empanelled auditors
- Maintain a Cyber Security Operations Centre (C-SOC) for real-time threat monitoring
- Report cyber incidents to RBI within 2–6 hours depending on severity
- Implement the Cyber Security Framework with controls mapped to international standards
- Conduct regular IT audits covering infrastructure, applications, and data security

SEBI VAPT Requirements

The Securities and Exchange Board of India mandates:

- All market intermediaries (stock exchanges, depositories, clearing corporations, brokers) must undergo VAPT
- VAPT must be conducted by CERT-In empanelled auditors
- Assessment reports must be submitted to SEBI within specified timelines
- Critical findings must be remediated before systems go live
- Annual comprehensive security audits covering all trading and settlement systems

CERT-In Empanelled Auditors

CERT-In maintains a panel of certified information security auditing organizations:

- Empanelment is granted after rigorous capability assessment
- Empanelled auditors are authorized to conduct security audits for government and regulated entities
- Organizations must engage empanelled auditors for critical infrastructure assessments
- CERT-In coordinates vulnerability disclosure and incident response nationally
- Mandatory 6-hour incident reporting requirement (CERT-In Direction, 2022)

Digital Personal Data Protection (DPDP) Act, 2023

Security obligations under DPDP Act:

- Data fiduciaries must implement "reasonable security safeguards" to protect personal data
- Breach notification to the Data Protection Board and affected individuals is mandatory
- Significant data fiduciaries must appoint a Data Protection Officer and conduct periodic data protection impact assessments
- Cross-border data transfer restrictions require security adequacy evaluations

India's National Cyber Security Strategy

India's cybersecurity strategy emphasizes:

- Building a secure and resilient cyberspace for citizens, businesses, and government
- Strengthening regulatory frameworks across sectors (banking, telecom, energy, healthcare)
- Developing indigenous cybersecurity capabilities and workforce
- Promoting public-private partnership for threat intelligence sharing
- Establishing sectoral CERTs for specialized incident response

Real-World: Security Assessments in Indian Banking

How Big 4 firms conduct security assessments for Indian banks:

1. **Engagement Setup:** Scope definition covering internet banking, mobile banking, UPI interfaces, core banking, ATM networks, and SWIFT connections
2. **Infrastructure Assessment:** Network architecture review, firewall rule analysis, segmentation verification, and external perimeter scanning
3. **Application Assessment:** OWASP-based testing of web and mobile applications, API security review, and business logic testing
4. **Compliance Mapping:** Controls mapped to RBI Cybersecurity Framework, ISO 27001, and PCI-DSS where applicable
5. **Risk Quantification:** Findings scored using CVSS and mapped to a risk matrix aligned with the bank's risk appetite
6. **Board Reporting:** Executive summary presented to the Board's IT Strategy Committee with remediation timelines
7. **Regulatory Submission:** Audit reports submitted to RBI as part of annual IS Audit compliance

NPCI security audit requirements for banks accessing UPI, IMPS, and RuPay networks include mandatory network security assessments, application penetration testing, and compliance verification against NPCI-specific security standards.

Assessment & Deliverables

Assessment Questions

Answer the following questions in your submission:

Q1. Describe the five phases of a security assessment methodology. For each phase, explain what activities are performed and what deliverables are produced. How does the platform you used in this lab map to each phase?

Q2. Explain the concept of defense in depth with its four layers (network, host, application, data). How did the network scan and web security scan in this lab assess different layers? Why is a single-layer defense insufficient?

Q3. Compare ISO 27001, PCI-DSS, and NIST CSF frameworks. What is each framework's primary focus, and how do they complement each other? Provide two examples of overlapping controls across the three frameworks.

Q4. Explain the 5x5 risk matrix used in this lab. How are likelihood and impact scores determined? What is the difference between inherent risk and residual risk, and why does it matter for remediation planning?

Q5. Describe the three report types generated in this lab (executive summary, technical report, remediation roadmap). Who is the target audience for each, and what key information does each contain? Why are different formats necessary?

Q6. What are the key requirements of the RBI Cybersecurity Framework for banks? How does SEBI's VAPT mandate differ from RBI's requirements? What role do CERT-In empanelled auditors play in both?

Q7. Explain remediation prioritization strategies (risk-based, effort-based, quick wins). Given the findings in this lab, propose a 90-day remediation plan with specific actions for each priority tier (immediate, 30-day, 90-day).

Q8. How does the Digital Personal Data Protection (DPDP) Act, 2023 create security obligations for organizations? What is the relationship between security assessments (like the one performed in this lab) and DPDP Act compliance? What are the consequences of a data breach under the DPDP Act?

Deliverables Checklist

Item	Description	Type	Status
Screenshot 1	Network scan results (ports, services)	Paste	☐
Screenshot 2	Web security findings (headers, SSL, cookies)	Paste	☐
Screenshot 3	ISO 27001 compliance status table	Paste	☐
Screenshot 4	Risk matrix visualization (5x5)	Paste	☐
Screenshot 5	Web dashboard or orchestrated pipeline	Paste	☐
Answers	Q1–Q8 written responses	Text	☐
Reports	exec_report.html, tech_report.html, roadmap.html	Files	☐
Results	Complete results/ directory with JSON data	Files	☐

Verification Checklist

Complete all items below before submitting:

- ☐ Repository cloned and project structure explored
- ☐ Virtual environment created and all dependencies installed
- ☐ `targets.yaml` configured with localhost as target
- ☐ CLI verified with all six commands accessible via `-help`
- ☐ Network scan completed with open ports and services identified
- ☐ Web security assessment completed with headers, SSL, cookies, and CORS findings
- ☐ ISO 27001 compliance check completed with control status table
- ☐ PCI-DSS and NIST CSF compliance checks completed with cross-framework comparison
- ☐ Risk score calculated with 5x5 matrix and weighted category scores
- ☐ Executive summary report generated and reviewed in browser
- ☐ Technical report generated with detailed findings and CVSS-like scores
- ☐ Remediation roadmap generated with prioritized Quick Wins / Short-term / Long-term actions
- ☐ Web dashboard launched and all four pages navigated (Risk, Findings, Compliance, Remediation)

- ☐ Full orchestrated assessment pipeline executed successfully
- ☐ All results exported to results/ directory
- ☐ All 5 required screenshots captured and pasted
- ☐ All 8 assessment questions answered

Grading Rubric

Criteria	Description	Points	Score
Setup	Repo cloned, venv created, config ready	5	___/5
Network Scan	Port scan with services and banners	10	___/10
Web Scan	Headers, SSL, cookies, CORS analysis	10	___/10
ISO 27001 Check	20 controls assessed with status	10	___/10
PCI-DSS/NIST Check	Cross-framework comparison	10	___/10
Risk Calculation	5x5 matrix, weighted score, rating	15	___/15
Reports	All 3 report types generated	15	___/15
Dashboard	Dashboard launched and navigated	10	___/10
Orchestration	Full pipeline executed successfully	10	___/10
Documentation	Screenshots + Q1–Q8 answers	5	___/5
	TOTAL	**100**	___/100

Appendix A: Security Assessment Phases Reference

Phase	Name	Key Activities	Deliverables
1	Planning & Scoping	Define objectives, scope, rules of engagement, timeline, authorization	Scope document, authorization letter, project plan
2	Discovery	Asset enumeration, port scanning, service detection, DNS analysis, network mapping	Asset inventory, network topology map, service catalog
3	Vulnerability Assessment	Configuration review, vulnerability scanning, web app testing, compliance checking	Vulnerability list, scan reports, compliance gaps
4	Risk Analysis	Likelihood assessment, impact analysis, risk scoring, risk matrix, prioritization	Risk register, risk matrix, priority rankings
5	Reporting	Executive summary, technical report, remediation roadmap, presentation, re-test plan	Final report package, remediation tracker

Appendix B: ISO 27001 Key Controls Summary

Control	Title	Description
A.5.1.1	Policies for InfoSec	Management direction for information security documented and approved
A.5.1.2	Review of Policies	Policies reviewed at planned intervals or on significant changes
A.6.1.1	Roles and Responsibilities	All security responsibilities defined and allocated
A.6.1.2	Segregation of Duties	Conflicting duties separated to reduce unauthorized modification
A.8.1.1	Inventory of Assets	Assets identified, inventoried, and owners assigned
A.8.2.1	Classification of Info	Information classified according to legal, value, and sensitivity
A.9.1.1	Access Control Policy	Policy established based on business and security requirements

Control	Title	Description
A.9.2.3	Privileged Access Mgmt	Allocation of privileged access rights restricted and controlled
A.9.4.1	Information Access	Access to information and application functions restricted per policy
A.10.1.1	Cryptographic Controls	Policy on use of cryptographic controls developed and implemented
A.10.1.2	Key Management	Policy on use, protection, and lifetime of cryptographic keys
A.12.1.1	Documented Procedures	Operating procedures documented, maintained, and available
A.12.2.1	Malware Controls	Detection, prevention, and recovery controls against malware
A.12.4.1	Event Logging	Event logs recording activities, exceptions, and security events
A.12.6.1	Vulnerability Mgmt	Information about technical vulnerabilities obtained and evaluated
A.13.1.1	Network Controls	Networks managed and controlled to protect information in systems
A.14.1.2	Securing App Services	Information in application services protected from fraud and misrouting
A.14.2.1	Secure Development	Rules for development of software and systems established
A.16.1.1	Incident Management	Responsibilities and procedures for incident management established
A.18.1.1	Applicable Legislation	All relevant legislative, regulatory, and contractual requirements identified

Appendix C: PCI-DSS Requirements Summary

Req	Title	Description
1	Firewall Configuration	Install and maintain firewall configuration to protect cardholder data
2	Default Passwords	Do not use vendor-supplied defaults for system passwords and security parameters

Req	Title	Description
3	Stored Cardholder Data	Protect stored cardholder data using encryption, truncation, masking, hashing
4	Encryption in Transit	Encrypt transmission of cardholder data across open, public networks
5	Anti-Malware	Protect all systems against malware and regularly update antivirus software
6	Secure Development	Develop and maintain secure systems and applications
7	Restrict Access	Restrict access to cardholder data by business need to know
8	Identify and Authenticate	Identify and authenticate access to system components
9	Physical Access	Restrict physical access to cardholder data and systems
10	Logging and Monitoring	Track and monitor all access to network resources and cardholder data
11	Security Testing	Regularly test security systems and processes (VAPT, IDS, file integrity)
12	Security Policy	Maintain a policy that addresses information security for all personnel

Appendix D: NIST CSF Functions and Categories

Function	Purpose	Category	Description
Identify (ID)	Understand risk	ID.AM	Asset Management — identify and manage assets
Identify (ID)	Understand risk	ID.RA	Risk Assessment — understand cybersecurity risk
Protect (PR)	Safeguard assets	PR.AC	Access Control — manage access permissions
Protect (PR)	Safeguard assets	PR.DS	Data Security — protect data integrity and confidentiality
Protect (PR)	Safeguard assets	PR.IP	Protective Processes — maintain security policies
Protect (PR)	Safeguard assets	PR.MA	Maintenance — perform maintenance and repairs
Detect (DE)	Identify events	DE.AE	Anomaly Detection — detect anomalous activity

Function	Purpose	Category	Description
Detect (DE)	Identify events	DE.CM	Continuous Monitoring — monitor for security events
Respond (RS)	Take action	RS.RP	Response Planning — execute response plans
Respond (RS)	Take action	RS.CO	Communications — coordinate response activities
Recover (RC)	Restore services	RC.RP	Recovery Planning — execute recovery plans
Recover (RC)	Restore services	RC.IM	Improvements — incorporate lessons learned

Appendix E: 5x5 Risk Matrix Template

Likelihood / Impact	Negligible (1)	Minor (2)	Moderate (3)	Major (4)	Critical (5)
Almost Certain (5)	5 (M)	10 (H)	15 (H)	20 (C)	25 (C)
Likely (4)	4 (L)	8 (M)	12 (H)	16 (C)	20 (C)
Possible (3)	3 (L)	6 (M)	9 (M)	12 (H)	15 (H)
Unlikely (2)	2 (L)	4 (L)	6 (M)	8 (M)	10 (H)
Rare (1)	1 (L)	2 (L)	3 (L)	4 (L)	5 (M)

Risk Rating Key: L = Low (1–4), M = Medium (5–9), H = High (10–15), C = Critical (16–25)

Treatment Guidance:

- **Critical (16–25):** Immediate action required. Escalate to senior management. Implement emergency controls.
- **High (10–15):** Priority remediation within 30 days. Assign dedicated resources.
- **Medium (5–9):** Scheduled remediation within 90 days. Include in regular maintenance.
- **Low (1–4):** Accept or address during next review cycle. Monitor for changes.

Appendix F: Report Structure Templates

Executive Summary Structure

1. Overall Risk Score and Rating (visual gauge or traffic light)
2. Assessment Scope and Methodology (1–2 sentences)
3. Top 5 Critical Findings (one-line summaries)

4. Compliance Status Summary (percentage by framework)
5. Key Recommendations (3–5 strategic actions)

Technical Report Structure

1. Assessment Overview (scope, methodology, tools used, timeline)
2. Finding Details (for each finding):
   - Title, Severity, CVSS Score
   - Affected Asset(s) and Component(s)
   - Description and Evidence (screenshots, code, logs)
   - Business Impact Analysis
   - Remediation Steps (specific, actionable)
   - References (CVE, CWE, OWASP)
3. Summary Statistics (findings by severity, category, asset)
4. Appendices (scan data, tool configurations)

Remediation Roadmap Structure

1. Quick Wins (0–30 days): Low effort, immediate risk reduction
2. Short-term Actions (30–90 days): Medium effort, significant improvement
3. Long-term Initiatives (90–180 days): Strategic improvements, architectural changes
4. Each action includes: Owner, Effort estimate, Dependencies, Success criteria

Appendix G: Indian Regulatory Compliance Matrix

Requirement Area	**RBI Framework**	**SEBI Mandate**	**CERT-In Direction**	**DPDP Act**
VAPT	Annual + post-change	Annual, empanelled auditors	Empanelled auditor list	Reasonable safeguards
Incident Reporting	2–6 hours to RBI	Immediate to SEBI	6 hours to CERT-In	To Board + individuals
Access Control	RBI CSF mandated	SEBI Circular	—	Data access controls
Encryption	Data at rest + transit	Payment data protection	—	Encryption recommended

Requirement Area	RBI Framework	SEBI Mandate	CERT-In Direction	DPDP Act
Logging	SOC mandatory	Audit trail required	180-day log retention	Audit logs for access
Board Oversight	IT Strategy Committee	Board-level reporting	—	DPO appointment
Data Protection	RBI data localization	Data localization for market data	—	Cross-border restrictions
Audit Frequency	Annual IS Audit	Annual + event-driven	As per empanelment	Periodic DPIA

Appendix H: Troubleshooting Guide

Problem: The network scan reports all ports as closed or filtered.
Solutions:

1. Verify that the target services (Flask app, database) are running before scanning
2. Check if your firewall or antivirus is blocking Python socket connections
3. Try scanning with a reduced port list: edit `config/targets.yaml` to include only port 5000
4. Run the Flask app first: `python dashboard/app.py` to ensure port 5000 is open
5. On Windows, run the terminal as Administrator if socket access is restricted

Problem: The compliance checker cannot evaluate controls and marks everything as Not Tested.
Solutions:

1. Ensure the network and web scans have been run first — compliance checks depend on scan data
2. Check that `config/thresholds.yaml` exists and has valid threshold values
3. Verify that scan result files exist in the expected output directory
4. Run scans in order: network → web → compliance, or use the orchestrator
5. Check the console for Python import errors or missing module messages

Problem: The Flask dashboard does not start, crashes on launch, or shows an empty page.
Solutions:

1. Ensure Flask is installed: `pip install flask`
2. Check if port 8080 is already in use: `netstat -an | grep 8080` (Linux/Mac) or `netstat -an | findstr 8080` (Windows)
3. Verify that the `dashboard/templates/` directory contains HTML template files
4. Ensure assessment data exists — run at least one scan before launching the dashboard
5. Check for Jinja2 template errors in the terminal output
6. Try a different port: modify the dashboard configuration or run with `-port 9090`

Appendix I: Resources and Tools Reference

Official Documentation

→ ISO 27001 Standard: `https://www.iso.org/isoiec-27001-information-security.html`

→ PCI-DSS v4.0: `https://www.pcisecuritystandards.org/`

→ NIST CSF: `https://www.nist.gov/cyberframework`

→ CERT-In: `https://www.cert-in.org.in/`

→ RBI Cyber Security Framework: `https://www.rbi.org.in`

→ SEBI Cybersecurity Circulars: `https://www.sebi.gov.in`

→ DPDP Act, 2023: `https://www.meity.gov.in`

→ OWASP: `https://owasp.org/`

→ CVSS Calculator: `https://www.first.org/cvss/calculator/3.1`

Tools Used in This Practical

Tool	Purpose	Cost
Python 3.8+	Programming language runtime	Free
Flask	Web dashboard framework	Free
requests	HTTP library for web scanning	Free
PyYAML	Configuration file parsing	Free
Click	CLI framework for command structure	Free
Rich	Terminal formatting and tables	Free
Jinja2	HTML report template engine	Free
pip	Python package manager	Free
Web Browser	Dashboard and report viewing	Free

Learning Resources

- "Information Security Management Handbook" — Tipton and Krause
- "The Web Application Hacker's Handbook" — Stuttard and Pinto
- "NIST Special Publication 800-30: Risk Assessment" — NIST
- "ISO 27001 Implementation Guide" — IT Governance Publishing
- "PCI-DSS Quick Reference Guide" — PCI Security Standards Council
- OWASP Testing Guide v4 — OWASP Foundation
- PortSwigger Web Security Academy (free online labs)
- SANS Reading Room (free whitepapers on security assessment)

—END OF LAB MANUAL—

Document Version: 1.0

IT Management & Audits – Practical Lab Series