NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY

SPECIALIZATION IN CYBER SECURITY

Cybersecurity Risk Evaluation Report (CISS)

Vulnerability Assessment & Reverse Engineering CT-371

Team Members:	**Roll Numbers:**
Hafsa Ali	CR-22006
Dania Fazal	CR-22007
Alishba Liaquat	CR-22019
Alizeh Mohsin	CR-22023
Mahnoor Shamim	CR-22049

Supervisor: Miss Saadia Arshad

1. *Executive Summary:*

This report presents a self-developed Cybersecurity Intelligent Scoring System (CISS), a web-based application that evaluates networked systems' security posture using weighted metrics. It simulates a cybersecurity audit model where users assess and quantify risks across 15 domains. The project supports awareness and helps organizations or individuals benchmark their readiness against cyber threats.

2. *Project Objective:*

- Develop a lightweight, intuitive risk scoring tool.
- Allow manual input of key cybersecurity indicators.
- Generate a composite security risk score based on weighted metrics.
- Present results in a clear, interpretable manner for stakeholders.

3. *System Overview:*

3.1 Architecture

- **Front-End:** HTML5, CSS (index.html)
- **Back-End:** Python Flask (app.py)
- **Communication:** RESTful API using JSON POST
- **Evaluation Model:** Weighted risk scoring formula

3.2 Technology Stack

Component	Technology
Web Server	Flask (Python)
Client	HTML/CSS
Data Flow	JavaScript/JSON

Media	MP4 Screen Demo

4. Evaluation Criteria:

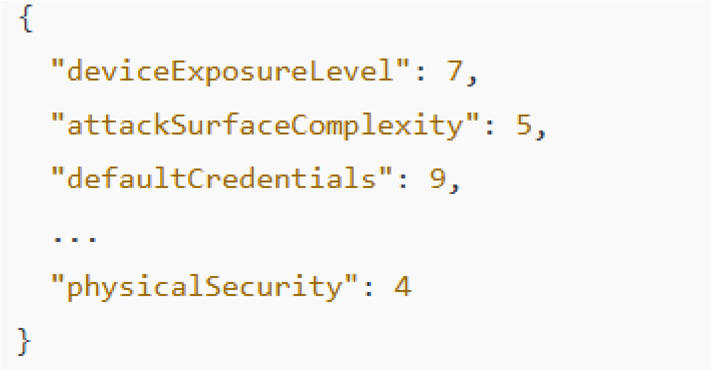
The system uses 15 predefined cybersecurity controls/attributes. Each is assigned a specific weight based on its contribution to overall risk.

ID	Metric Name	Description	Weight
M1	Device Exposure Level	System accessibility and external visibility	0.10
M2	Attack Surface Complexity	Complexity of available services and access points	0.08
M3	Use of Default Credentials	Risk from unchanged factory-set logins	0.10
M4	Firmware Update Mechanism	Availability and automation of firmware updates	0.08
M5	Vulnerability Disclosure Timeline	Responsiveness to disclosed flaws	0.06
M6	Code Integrity	Presence of code signing and anti-tampering measures	0.05
M7	Access Control Granularity	Detail and strictness of access policies	0.07
M8	Encryption Usage	Use of secure cryptographic practices	0.10
M9	Device Identity Verification	Unique and verifiable identity of devices	0.06
M10	Secure Channel Usage	Use of HTTPS, VPNs, encrypted tunnels	0.08
M11	Logging & Monitoring	Availability of activity logs and alerts	0.06
M12	Supply Chain Security	Risk evaluation in procurement and vendor management	0.05
M13	Threat Intelligence Integration	Use of external threat feeds or alerts	0.10
M14	Backup and Recovery Readiness	Existence of disaster recovery or business continuity plans	0.05
M15	Physical Security	Controls against physical tampering or theft	0.06

5. System Workflow:

1. **User Input:**
   a. End-user accesses a local webpage and fills in ratings (scale 0–10) for each security metric.
2. **Data Submission:**
   a. Ratings are packaged into a JSON payload and POSTed to the Flask server.
3. **Score Calculation:**
   a. Server parses the values, applies weights, and computes the final score.
4. **Output Delivery:**
   a. Score returned as a percentage or normalized number indicating risk level.

6. Sample JSON Input Format:



```
{
  "deviceExposureLevel": 7,
  "attackSurfaceComplexity": 5,
  "defaultCredentials": 9,
  ...
  "physicalSecurity": 4
}
```

7. Result Interpretation:

Score Range	Risk Category	Recommendation
0–3	High Risk	Immediate remediation necessary
4–6	Medium Risk	Prioritize vulnerabilities for mitigation

7–10	Low Risk	Maintain posture, periodic audits

8. Testing & Demonstration:

A video demo (`bandicam_2025-05-04_15-29-25-514.mp4`) showcases:

- Launching the local server
- Submitting form data
- Viewing real-time score response

9. Use Cases:

- **University Projects**: Teaching cyber risk awareness
- **Enterprise Self-Audits**: Quick internal assessments
- **IoT Vendor Evaluation**: Score third-party devices before adoption
- **Incident Response**: Snapshot posture before and after events

10. IoT devices scoring:

13.1 Reference:

In late 2016, a widespread cyberattack was carried out using a large botnet composed of compromised Internet of Things (IoT) devices, primarily IP cameras and home routers. These devices were infected by the Mirai malware, which systematically scanned the internet for vulnerable IoT devices using factory default usernames and passwords. Once compromised, the devices became part of a coordinated Distributed Denial of Service (DDoS) attack against critical internet infrastructure.

The affected devices typically:

Operated with default credentials and open network ports (e.g., Telnet),

Lacked secure firmware update mechanisms,

Did not support signed firmware or any form of data encryption,

Were built with minimal or no hardware security features, *Had poor access control granularity, and*

Were manufactured without stringent supply chain security oversight.

This case highlighted several systemic weaknesses in IoT security and exemplified how large-scale exploitation can result from inadequate security practices at both the firmware and network levels.

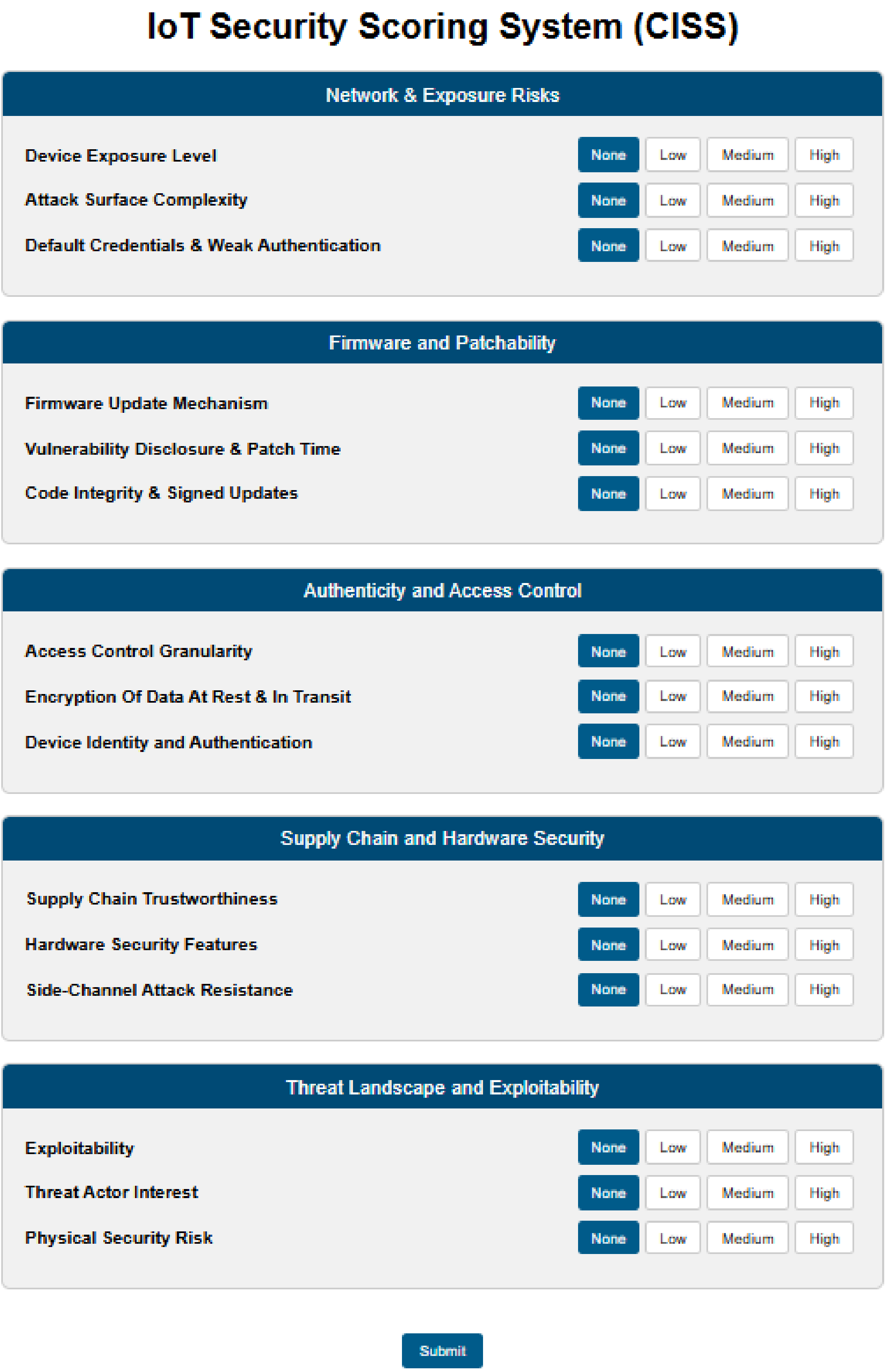
Sources:

Antonakakis et al., "Understanding the Mirai Botnet", arXiv:2007.13410

[*https://arxiv.org/abs/2007.13410*](https://arxiv.org/abs/2007.13410)

The Verge, "A massive DDoS attack is causing outages across the internet"

[*https://www.theverge.com/2016/10/21/13362354/ddos-attack-dyn-dns-internet-outage*](https://www.theverge.com/2016/10/21/13362354/ddos-attack-dyn-dns-internet-outage)



IoT Security Scoring System (CISS)

Network & Exposure Risks

Device Exposure Level	[✓ None]	Low	Medium	High
Attack Surface Complexity	[✓ None]	Low	Medium	High
Default Credentials & Weak Authentication	[✓ None]	Low	Medium	High

Firmware and Patchability

Firmware Update Mechanism	[✓ None]	Low	Medium	High
Vulnerability Disclosure & Patch Time	[✓ None]	Low	Medium	High
Code Integrity & Signed Updates	[✓ None]	Low	Medium	High

Authenticity and Access Control

Access Control Granularity	[✓ None]	Low	Medium	High
Encryption Of Data At Rest & In Transit	[✓ None]	Low	Medium	High
Device Identity and Authentication	[✓ None]	Low	Medium	High

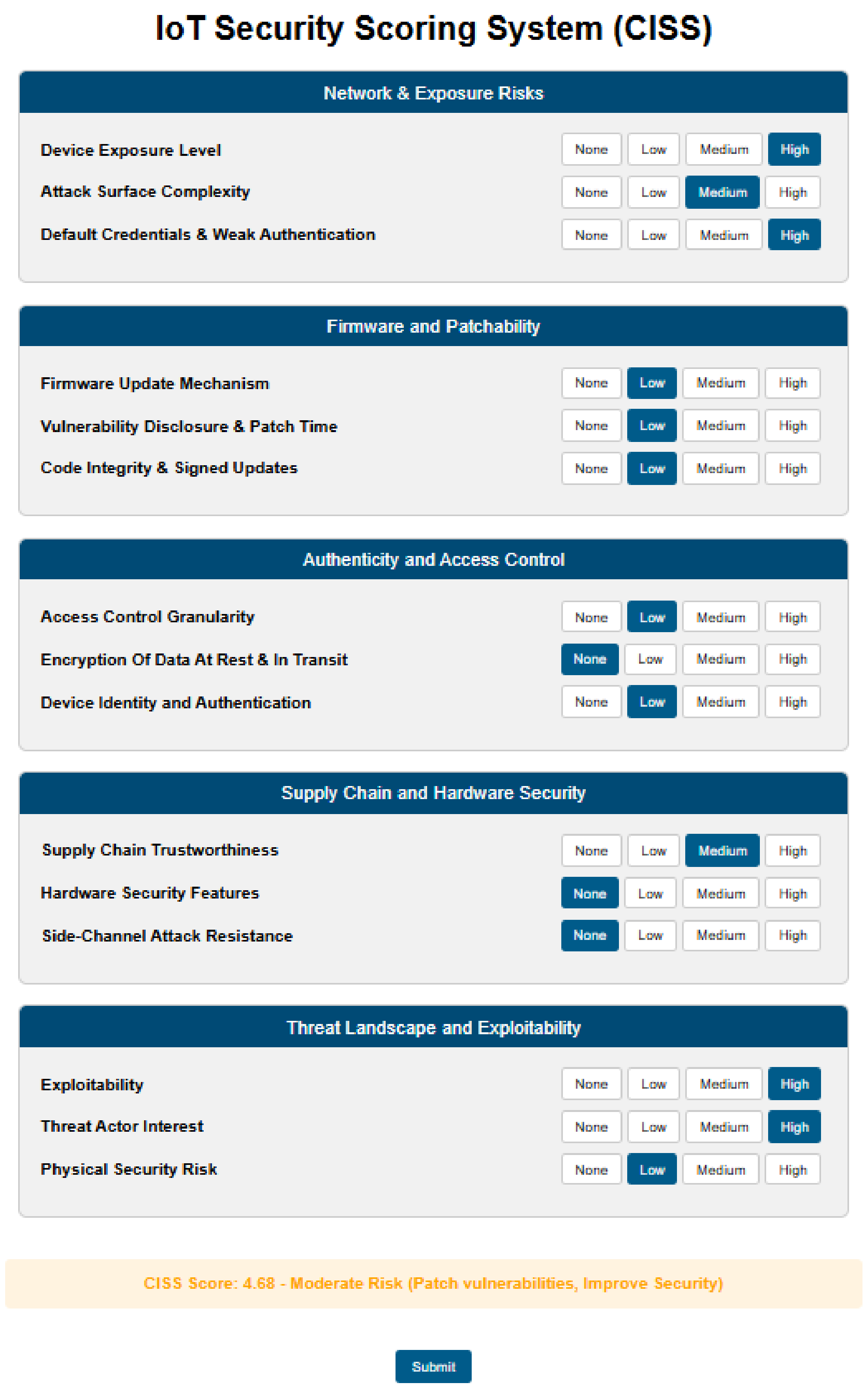
Supply Chain and Hardware Security

Supply Chain Trustworthiness	[✓ None]	Low	Medium	High
Hardware Security Features	[✓ None]	Low	Medium	High
Side-Channel Attack Resistance	[✓ None]	Low	Medium	High

Threat Landscape and Exploitability

Exploitability	[✓ None]	Low	Medium	High
Threat Actor Interest	[✓ None]	Low	Medium	High
Physical Security Risk	[✓ None]	Low	Medium	High

ar



IoT Security Scoring System (CISS)

Network & Exposure Risks

Device Exposure Level	None	Low	Medium	[✓ High]
Attack Surface Complexity	None	Low	[✓ Medium]	High
Default Credentials & Weak Authentication	None	Low	Medium	[✓ High]

Firmware and Patchability

Firmware Update Mechanism	None	[✓ Low]	Medium	High
Vulnerability Disclosure & Patch Time	None	[✓ Low]	Medium	High
Code Integrity & Signed Updates	None	[✓ Low]	Medium	High

Authenticity and Access Control

Access Control Granularity	None	[✓ Low]	Medium	High
Encryption Of Data At Rest & In Transit	[✓ None]	Low	Medium	High
Device Identity and Authentication	None	[✓ Low]	Medium	High

Supply Chain and Hardware Security

Supply Chain Trustworthiness	None	Low	[✓ Medium]	High
Hardware Security Features	[✓ None]	Low	Medium	High
Side-Channel Attack Resistance	[✓ None]	Low	Medium	High

Threat Landscape and Exploitability

Exploitability	None	Low	Medium	[✓ High]
Threat Actor Interest	None	Low	Medium	[✓ High]
Physical Security Risk	None	[✓ Low]	Medium	High

CISS Score: 4.68 - Moderate Risk (Patch vulnerabilities, Improve Security)

11. Strengths:

- Simple yet informative tool
- Easy customization of weights
- Works offline (no dependency on cloud)
- Transparent logic (open and editable)

12. Limitations:

- Manual entry may introduce bias
- Not integrated with actual vulnerability scanners
- No user authentication or historical tracking
- Fixed weight model, lacks adaptiveness

13. Future Enhancements

- Add user accounts and login
- Enable automated scanning via Nmap APIs
- Include visual dashboards (charts, graphs)
- Integrate with CVE databases for live threat updates
- Support exportable PDF/Excel reports

14. Conclusion:

The CISS project effectively demonstrates how qualitative cybersecurity metrics can be translated into a quantitative model. This tool is a proof-of-concept that highlights the importance of structured security evaluation and presents a foundation for building more advanced, integrated platforms in the future.