ML-Protected Web Application Against SQL Injection & XSS

A Real-World Machine Learning + Web Security CTF Challenge GitHub Repository: https://github.com/showkothosen/ML-Protected-CTF-WebApp
Live Demo: Coming soon on Render / Railway

Project Overview

This is a **fully functional, intentionally vulnerable Flask web application** designed as a **realistic Capture The Flag (CTF) challenge** that showcases the **practical power of Machine Learning in detecting and responding to** the two most dangerous web vulnerabilities in the **OWASP Top 10**:

1. **A03:2021 – Injection (SQL Injection)**
2. **A07:2021 – Cross-Site Scripting (XSS)**

Every single user input (login, register, search, ping, upload, profile) is **monitored in real time** by two **highly accurate pre-trained**

- **SQLi Detector → 100% accuracy** [pre-trained Decision Tree model]
- **XSS Detector → 99.99% accuracy**[pre-trained Logistic Regression model]

The app **does NOT block** attacks (so players can solve it), but instead **detects, logs, flashes warnings, and even rewards successful attacks with flags** — simulating a next-gen **ML-enhanced Web Application Firewall (WAF)**.

OWASP Top 10 Context

Vulnerability	OWASP Rank (2021)	Real-World Risk	ML Defense in This Project
SQL Injection	A03:2021	Full DB takeover, credential dump	100% real-time detection + flag reveal
Cross-Site Scripting	A07:2021	Session hijacking, defacement, keylogging	99.99% detection + JS alert flag

Project Structure

```
ML-Protected-CTF-WebApp/
│
├── app.py                  # Core Flask app + ML detection engine
├── ctf.db                  # Auto-generated SQLite DB
├── models/
│   ├── sqli_dt.joblib        # SQLi Decision Tree model
│   ├── sqli_vec.joblib       # SQLi TF-IDF vectorizer
│   ├── xss_dt.joblib         # XSS Logistic Regression model
│   └── xss_vec.joblib        # XSS TF-IDF vectorizer
│
├── templates/
│   ├── base.html
│   ├── login.html
│   ├── register.html
│   ├── home.html
│   ├── profile.html
│   └── alerts.html
├── static/
│   └── style.css
│
├── uploads/uploads           # Auto-created upload folder
└── requirements.txt
```

Key Features & CTF Gameplay

Feature	Normal Users	Attackers / CTF Players	ML Response
Login / Register	Works normally	SQLi / XSS in credentials	Detected + **SQLi flag revealed**
Search Bar	Normal search	SQLi / XSS payloads	Detected + warning
Ping Tool	Ping any IP	Command injection attempts	Sanitized + ML scan
File Upload	Upload files	Malicious filenames (`shell.php`, `<script>`)	Detected + logged
Profile View (`?id=`)	View profiles	IDOR / Enumeration (`id=313`)	Detected + **IDOR flag revealed**

Flags (All Revealed on Successful Detection!)

Flag	How to Get It	Trigger
`flag{SqL_inject10n_1s_fun}`	Trigger **any SQLi** (e.g., `admin'--`)	ML detects → flashes flag instantly
`flag{XSS_1s_tr1cky_but_fun}`	Trigger **any XSS** (e.g., `<script>alert(1)</script>`)	ML detects → **JavaScript alert** with flag
`flag{IDOR_Found_1n_th3_m4tr1x}`	Visit `/profile?id=313`	Direct access → flag shown in profile

Machine Learning Models Performance

Model	Algorithm	Dataset Size	Accuracy	Precision	Recall	F1-Score
SQLi Detector	Decision Tree	~25k (synthetic + real)	**100%**	1.00	1.00	1.00
XSS Detector	Decision Tree	~40k (Kaggle + augmented)	**99.99%**	0.9998	1.00	0.9999

Local Installation & Running (100% Tested)

```
git clone https://github.com/showkothosen/ML-Protected-CTF-WebApp.git

cd ML-Protected-CTF-WebApp

python -m venv venv

source venv/bin/activate    # Linux/Mac

# venv\Scripts\activate     # Windows

pip install scikit-learn==1.2.2 numpy==1.26.4 pandas==2.2.3 joblib==1.5.2 Flask gunicorn

python app.py
```

Open → *http://127.0.0.1:5000*

Tools & Technologies Used

Purpose	Tool Used
Backend & Routing	Flask (Python)
Machine Learning Models	scikit-learn 1.2.2
Vectorization	TF-IDF (CountVectorizer alternative)
Model Persistence	joblib
Database	SQLite (ctf.db)
Frontend Templates	Jinja2 + HTML/CSS
Development Assistance	**Grok (xAI)**, ChatGPT 4o
Documentation & Debugging	Grok + VS Code + GitHub Copilot
Testing Payloads	Burp Suite Community, Custom Lists

Future Roadmap

- Add LFI / SSTI / RCE ML detectors
- Real-time dashboard with Chart.js
- Auto-model retraining endpoint
- Docker + CI/CD pipeline
- Deploy as public CTF on TryHackMe / HackTheBox
- Research paper submission (IEEE / Springer)

Why This Project Stands Out

This is **not just another vulnerable app**. It is a **working prototype of tomorrow's cybersecurity**: **Machine Learning that doesn't just detect — it learns, responds, and even rewards ethical hacking.**

Perfect for:

- University final year projects
- Cybersecurity + ML portfolio
- Job interviews (FAANG, Palo Alto, CrowdStrike, etc.)
- Research in Adversarial ML & WAF Evasion

About the Author

Showkot Hosen Final Year B.Sc. Engineering Electronics & Telecommunication Engineering Chittagong University of Engineering & Technology (CUET)

Credentials:

- CISCO Ethical Hacking Badge Holder
- ISC2 Certified in Cybersecurity (CC) Candidate
- TryHackMe Top 1% Global Rank
- Active HTB & PicoCTF Solver
- Passionate about **ML-Powered Cyber Defense**

Vision: Pursue a **fully-funded M.Sc./Ph.D. in Artificial Intelligence for Cybersecurity** to build the next generation of intelligent, adaptive defense systems.

"In the age of AI, the best firewall thinks."

Star this repo if you believe in the future of **ML-driven cybersecurity**!

Pull requests welcome · Issues encouraged · Let's make the web safer together.

Contact:

📧: shrahat56@gmail.com
LinkedIn: https://www.linkedin.com/in/showkot-hosen10
Kaggle: https://www.kaggle.com/showkothosen
TryHackMe: https://tryhackme.com/p/Showkot313