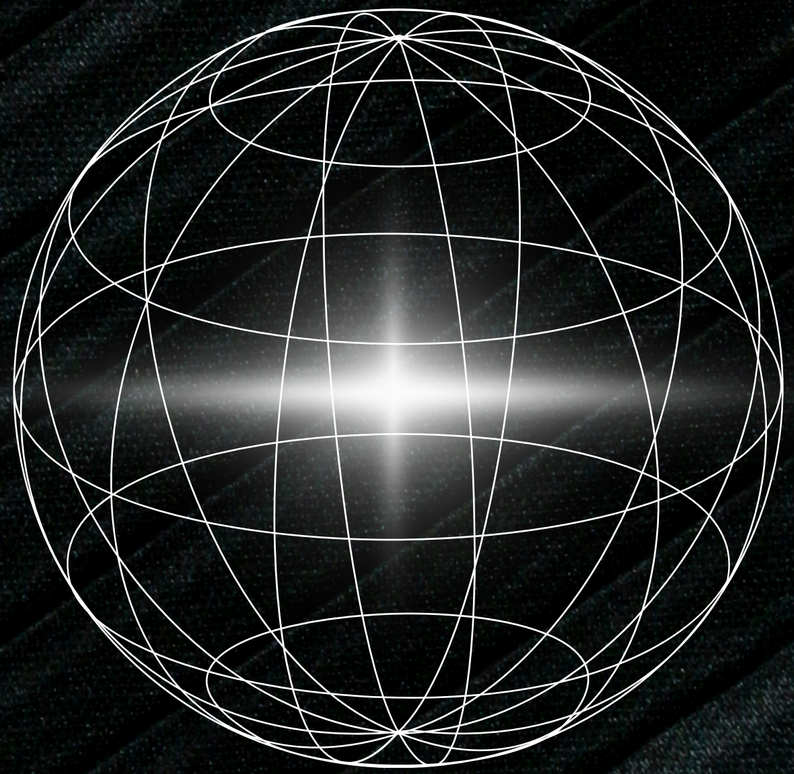
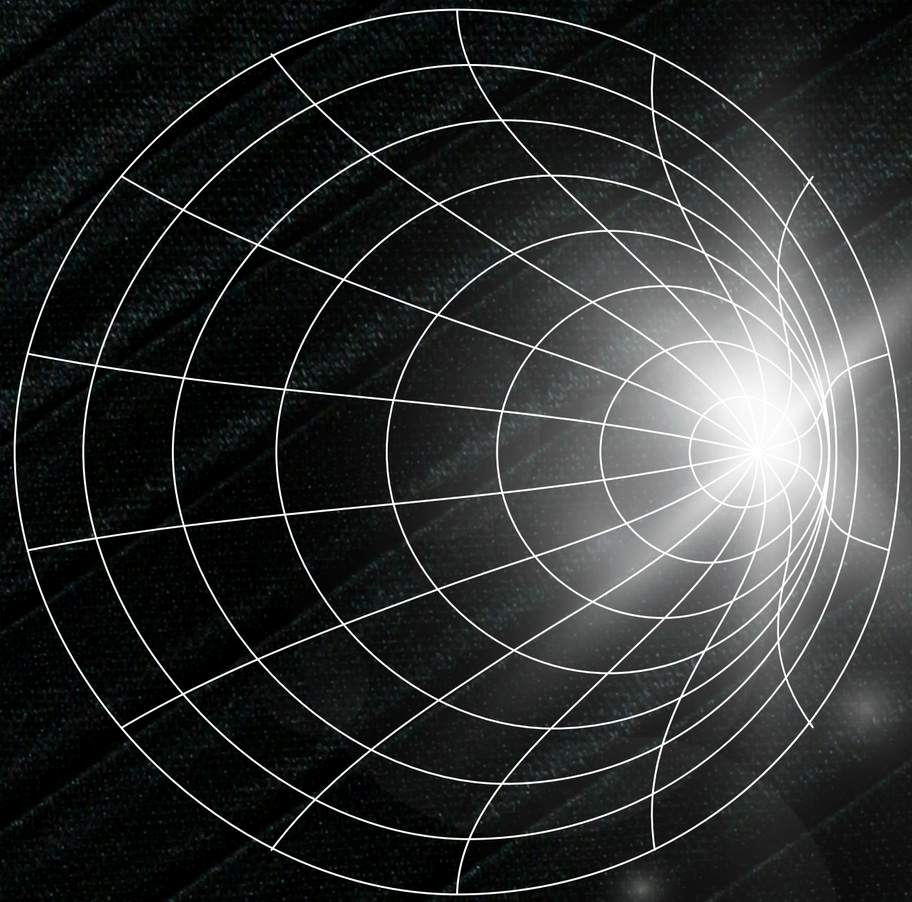
# Building a Secure DGKart Application

"Focusing on the OWASP Top 10 Vulnerabilities"



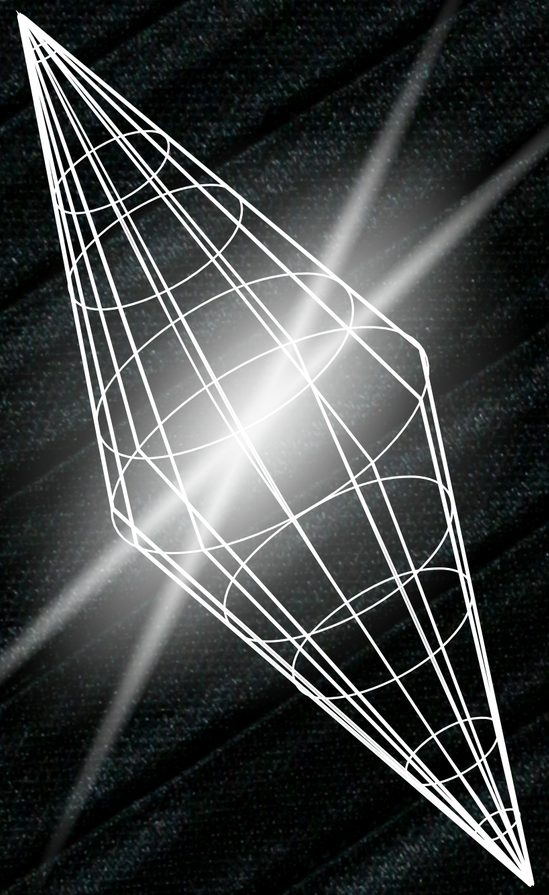
by Nischal Subedi

# Contents



01 - Introduction
02 - Campaign
03 - Statistics
04 - Documentation
05 - Conclusions

# Introduction to DGKart Application



- Overview of DGKart:
  - A shopping cart application.
  - Importance of security in e-commerce platforms.
- Goal of the Presentation:
  - Discuss security strategies for mitigating the top 10 security risks identified by OWASP.

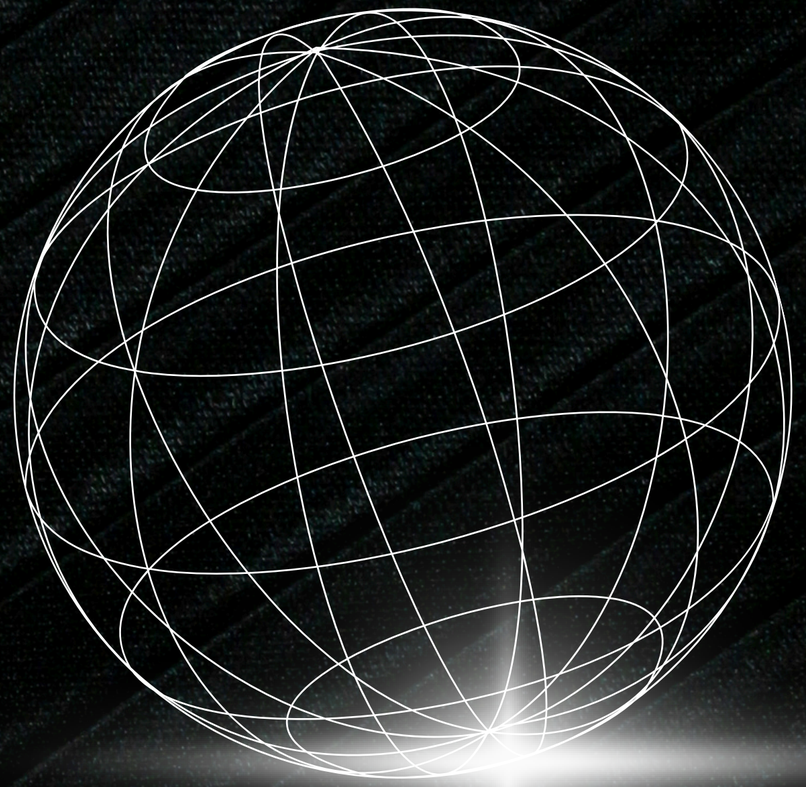
# Introduction to OWASP Top 10

What is OWASP?

- Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve software security.

- OWASP Top 10:
  - Industry-standard for web application security, highlighting the most critical risks.

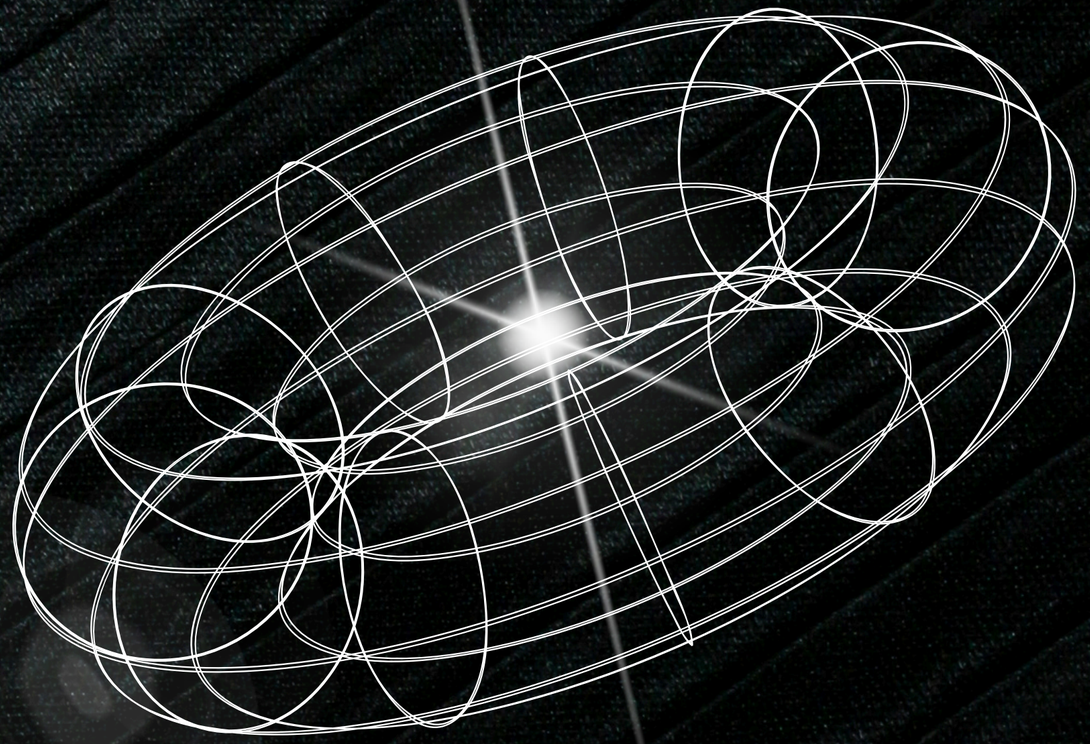
# OWASP Top 10 Overview



**Quick overview of the top 10 vulnerabilities:**

1. **Injection**
2. **Broken Authentication**
3. **Sensitive Data Exposure**
4. **XML External Entities (XXE)**
5. **Broken Access Control**
6. **Security Misconfiguration**
7. **Cross-Site Scripting (XSS)**
8. **Insecure Deserialization**
9. **Using Components with Known Vulnerabilities**
10. **Insufficient Logging and Monitoring**

# 03 - Statistics



"We improve our online presence"

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

# Injection (Example in DGKart)

- **Definition:**
  - **Occurs when untrusted data is sent to an interpreter.**
- **Example in DGKart:**
  - **SQL injection in product search queries.**
- **Prevention:**
  - **Use parameterized queries and prepared statements.**

# Broken Authentication

- **Definition:**
  - **Allows attackers to compromise user accounts.**
- **Example in DGKart:**
  - **Weak password policies.**
- **Prevention:**
  - **Enforce strong password rules.**
  - **Implement multi-factor authentication (MFA).**

# Sensitive Data Exposure

- **Definition:**
  - **Failure to protect sensitive data such as credit card info.**
- **Example in DGKart:**
  - **Plain text storage of payment details.**
- **Prevention:**
  - **Encrypt sensitive data in storage and transit.**
  - **Use HTTPS everywhere.**

# Security Misconfiguration

- **Definition:**
  - **Misconfigurations in security settings.**
- **Example in DGKart:**
  - **Default admin credentials left unchanged.**
- **Prevention:**
  - **Regularly update and patch software.**
  - **Disable unnecessary features.**

# Cross-Site Scripting (XSS)

- **Definition:**
  - **Attackers inject malicious scripts into web pages.**
- **Example in DGKart:**
  - **Reflected XSS in product reviews.**
- **Prevention:**
  - **Validate and escape user inputs.**

# Insecure Deserialization

- **Definition:**
  - **Allows data to be tampered with during deserialization.**
- **Example in DGKart:**
  - **Tampering with serialized objects like order information.**
- **Prevention:**
  - **Avoid using deserialization for user inputs.**

# Using Components with Known Vulnerabilities

- **Definition:**
  - **Utilizing outdated or vulnerable third-party components.**
- **Example in DGKart:**
  - **Use of outdated payment gateway libraries.**
- **Prevention:**
  - **Regularly update all libraries and dependencies.**

# Insufficient Logging and Monitoring

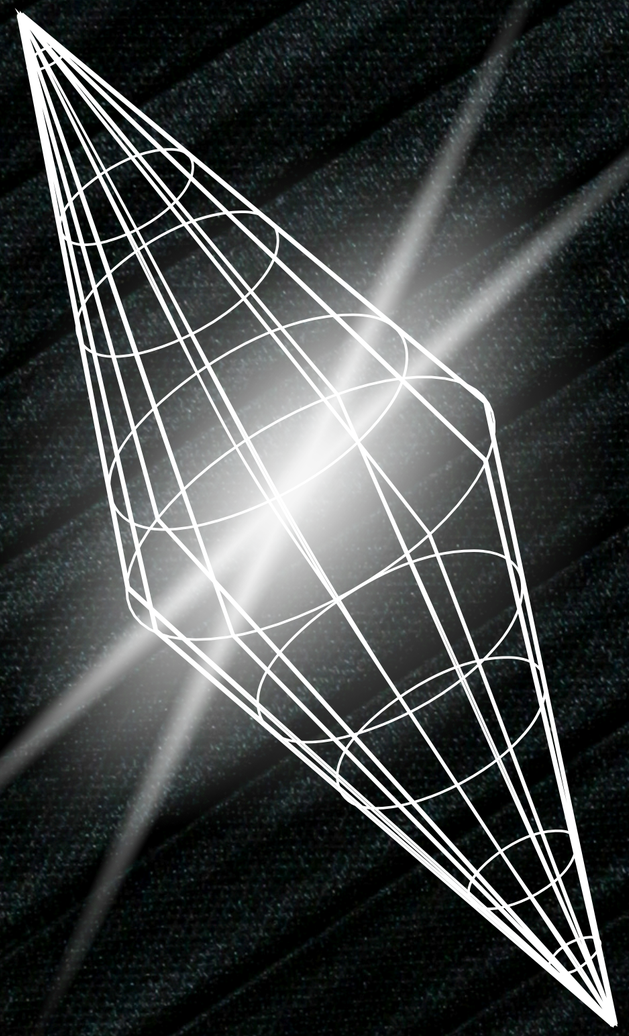
- **Definition:**
  - **Failure to detect or respond to security incidents.**
- **Example in DGKart:**
  - **No logging for failed login attempts.**
- **Prevention:**
  - **Implement comprehensive logging and alert systems.**

# Final Recommendations

- **Security Practices for DGKart:**
  - **Regular security audits.**
  - **Penetration testing.**
  - **Continuous monitoring for vulnerabilities.**
  - **Employee training on security best practices.**

# 05 - Conclusions

- **Security is essential for DGKart's success**. By implementing OWASP Top 10 strategies, we protect both users and the business.
- **Proactive security measures** such as regular audits, secure coding practices, and constant monitoring help prevent potential breaches.
- **User trust is built on security**. Ensuring data protection enhances customer confidence and loyalty.
- **Security is a continuous process**. Staying updated with evolving threats and best practices is key to maintaining a secure platform.
  - .



# Thanks