Vulnerability and Exploit Report

secure.uscarteams.org

**Report Date**: July 28, 2025

**Domain Evaluated**: https://secure.uscarteams.org/secure/vroom1\_files/

**IP Address**: 64.9.213.34

**Server**: Apache/2.2.22 (Ubuntu)

**Certificate CN**: ewcap.uscarteams.org

**Certificate SANs**: ewcap.uscarteams.org, secure.uscarteams.org, www.uscarteams.org

# Executive Summary

This report highlights **multiple high-risk vulnerabilities** associated with the above domain:

1. **Public Access to Sensitive File Repository**: Full recursive access is enabled to all files and folders under /secure/vroom1\_files/, bypassing any authentication or authorization checks.
2. **Outdated Apache and TLS Configurations**: The server supports deprecated TLS protocols and weak/insecure ciphers, allowing downgrade and MITM attacks.
3. **Misaligned SSL Certificate Chain**: The certificate is not issued by a widely trusted public CA (e.g., DigiCert, GeoTrust), and OCSP stapling is not enabled.
4. **Weak Diffie-Hellman (DH) Parameters**: The server negotiates DH key exchanges using small key sizes that modern clients reject, which required explicit weakening of the client’s security settings to establish a connection.

These vulnerabilities expose the organization to legal, regulatory, and reputational risks, especially if any of the exposed data is sensitive, proprietary, or non-public.

## 1. Public File Exposure (Confirmed Exploit)

* **Vulnerability**: Directory listing is enabled on a public Apache server under https://secure.uscarteams.org/secure/vroom1\_files/.
* **Impact**: Any unauthenticated user can:
  + Traverse all folders and subfolders recursively
  + Download all files directly without restriction
* **Proof of Exploit**:
  + A recursive Python script (attached separately) successfully recreated the entire directory and file structure under a local MIGRATION/ folder.
  + No login, token, or API key was required.
* **Exploit Risk**:
  + Unauthorized access to intellectual property, meeting notes, research, or other proprietary content.
  + Risk of data scraping or exfiltration using automated tools.

## 2. SSL/TLS Vulnerabilities

### Weak SSL/TLS Protocols

* The server supports deprecated protocols:
  + TLSv1.0 and TLSv1.1 — **deprecated since 2020** by major vendors
  + Allows cipher suites vulnerable to attacks like **BEAST** and **SWEET32**

|  |  |  |
| --- | --- | --- |
| Protocol Version | Status | Risk |
| TLSv1.0 | Enabled (Insecure) | ❌ |
| TLSv1.1 | Enabled (Insecure) | ❌ |
| TLSv1.2 | Enabled | ✅ |

### Insecure Ciphers Detected

* **Insecure (3DES-based) cipher suites in use**:
  + TLS\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
  + TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA
* These ciphers are vulnerable to known cryptographic attacks such as **SWEET32**, and should no longer be used per current NIST and OWASP guidance.

### Weak Diffie-Hellman Parameters

* The server negotiates DH parameters that are too small (likely 1024-bit or less).
* Modern Python clients reject connections with:

SSLError(1, '[SSL: DH\_KEY\_TOO\_SMALL]')

* Exploited by manually weakening client security via:

ctx.set\_ciphers("DEFAULT:@SECLEVEL=1")

## 3. Certificate Chain and Validation

* **Issuer**: R11, chained to ISRG Root X1
* **OCSP Stapling**: ❌ Not enabled
* **CRL**: ✅ Responds with “Good”
* **Validation Warnings**:
  + While the certificate is technically valid, it is **not issued by DigiCert, GeoTrust, Thawte, or RapidSSL**—raising compatibility and trust chain concerns for corporate networks.
  + **OCSP validation cannot be completed**, reducing resilience against revoked certificates.
* **Concerns:**
  + While the certificate is technically valid and matches the domain, OCSP stapling is not enabled, and the certificate is not issued by a widely trusted enterprise CA (e.g., DigiCert, GeoTrust, Thawte).
  + This may affect compatibility with security-conscious systems and corporate clients.

# Recommendations

## File Access

* Immediately disable directory listing under /secure/vroom1\_files/:

Options -Indexes

* Apply authentication and authorization requirements to access this folder.
* Scan and audit all exposed files for sensitive content.

## SSL/TLS Hardening

* **Disable TLSv1.0 and TLSv1.1** via ssl.conf or virtual host settings:

SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

* **Remove insecure ciphers**, including all 3DES suites:

SSLCipherSuite HIGH:!aNULL:!MD5:!3DES

SSLHonorCipherOrder on

* Generate and use **strong 2048-bit or 3072-bit DH parameters**:

openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048

* Then reference this in Apache config:

SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"

## Certificate and OCSP

* Enable OCSP Stapling in Apache:

SSLUseStapling on

SSLStaplingResponderTimeout 5

SSLStaplingReturnResponderErrors off

* Consider renewing your certificate through a publicly recognized CA if you plan to support broader compatibility.

## Exploit Script (Summary)

python get\_files.py  # Recursively downloads and reconstructs all content

An automated script (available on request) was able to fully exploit the site using the following steps:

* Crawl all directory indexes recursively.
* Download every file via direct HTTPS GET requests.
* Reconstruct the data in a local folder (MIGRATION/), preserving the full structure.

**No authentication or session mechanism was required.**

This constitutes a full **data exposure event** from a security classification standpoint.

## Risk Assessment

|  |  |  |
| --- | --- | --- |
| Area | Risk Level | Potential Consequences |
| Data Exposure | Critical | Unauthorized access to internal/proprietary data |
| TLS/SSL Configuration | High | Vulnerable to downgrade or MITM attacks |
| Legal / Regulatory | High | Potential violations of GDPR, CCPA, or industry-specific data handling rules |
| Reputational Risk | Medium | Loss of trust with partners or public if data is found |