
# *OSM + ANAGRAFE WEB GUINEA BISSAU*
*(OSMGB)*

# SECURITY ASSESSMENT FINDINGS REPORT

*Data: 18/02/2020*
*Progetto: OSMGB*
*Versione: 1.0*

Pagina lasciata intenzionalmente vuota

| Versione | Descrizione | Revisore |
|---|---|---|
| 1.0 | Prima versione del documento | Pimen Flavian Dei |

# Sommario

# Disclaimer

Un Penetration Testing è considerata un'istantanea in un certo momento. Le scoperte e le raccomandazioni si riferiscono alle informazioni raccolte durante i test e non a modifiche effettuate al di fuori dal periodo di test.

# Introduzione

## *Informazioni generali*

**Inizio dei test:** 24/01/2020

**Fine dei test:** 15/02/2020

**Scope:**

- http://ntchangue3.altervista.org

**Tipo di consulenza:** White Box Penetration Testing

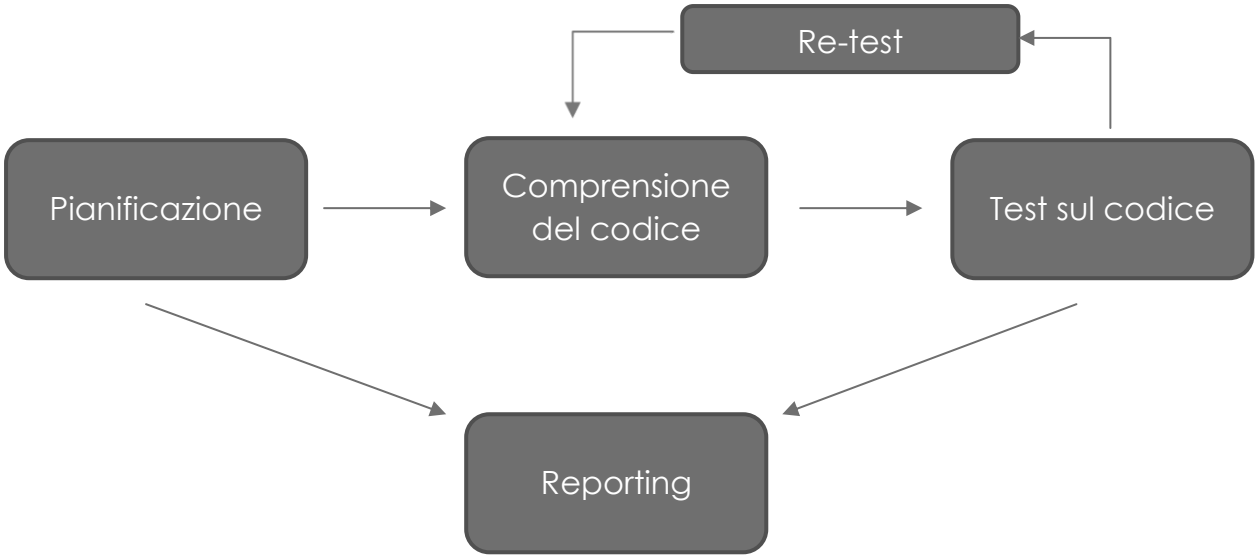
## *Panoramica*

Nel giorno 14/01/2020 il professore **Carlone Alfonso** ha richiesto una *White Box Penetration* sul progetto **NTchangue** della classe 5° A dell'istituto tecnico *Amedeo Avogadro*. I test sono incentrati sulla sicurezza dell'applicazione, attraverso strumenti automatici e test manuali.

Al presente è stata fornito il pieno accesso al panello di amministrazione (presente sul sito di altervista) con la possibilità di analizzare il codice sorgente del sito.

Le fasi relative a questo Penetration testing sono le seguenti:

- Pianificazione
- Comprensione del codice
- Test sul codice
- Reporting




# Severità delle vulnerabilità

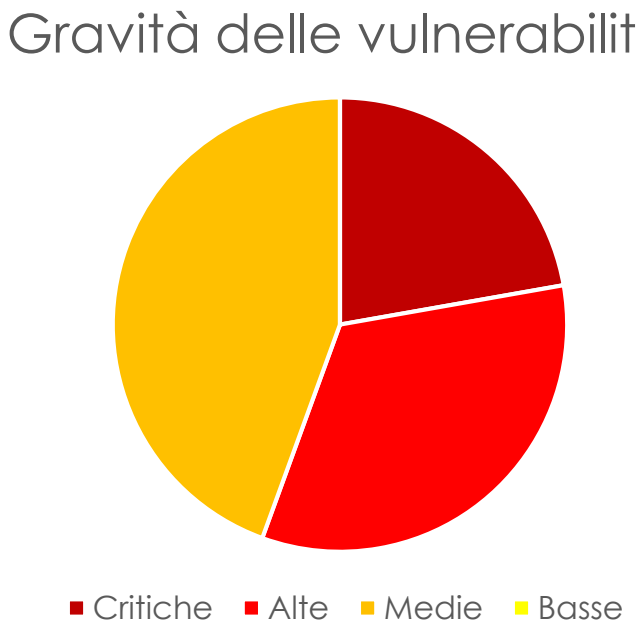
La seguente tabella definisce i livelli di sicurezza.

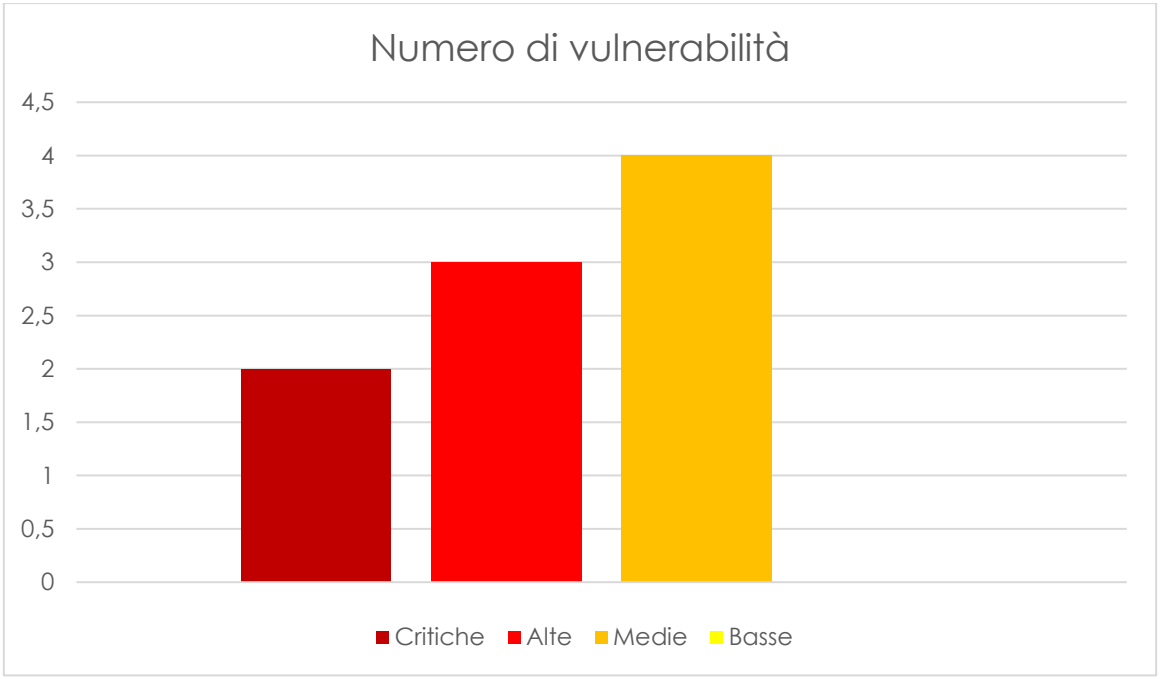
| Severità | Descrizione |
|---|---|
| Critica | È altamente consigliato creare un piano di azione e risolvere il **immediatamente** la vulnerabilità. |
| Alta | È altamente consigliato creare un piano di azione e risolvere il **prima possibile** la vulnerabilità. |
| Media | È altamente consigliato creare un piano di azione e risolvere la vulnerabilità dopo aver risolto le vulnerabilità **Alte.** |
| Bassa | È altamente consigliato creare un piano di azione e risolvere la vulnerabilità alla **prossima versione** del codice. |

# Executive Summary

I test hanno rivelato un quadro generale **critico**, cosa normale per un progetto con pochi mesi di vita e con membri della squadra senza la conoscenza dei vettori di attacco. È consigliato seguire tutte le raccomandazioni presenti nell'ultima sezione del documento e nella descrizione delle singole vulnerabilità.

Osservando i grafici qui sotto si può notare che c'è bisogno di un'azione immediata per risolvere le vulnerabilità **Critiche** e **Alte** il prima possibile.






# Vulnerabilità

## Weak Password Requirements

### *CWE*
CWE-521

### *Gravità*
Critica

### *Descrizione*
L'applicazione non richiede che l'utente debba avere password robuste cosi rendendo più semplice per un attaccante compromettere l'account dell'utente

### *Mitigazione*
Incoraggiare l'uso di password robuste. Una password Policy dovrebbe contenere i seguenti attributi:

- Lunghezza minima
- Richiedere caratteri  misti (alpha, numerici, speciali, maiuscole e minuscole)
- Non deve contenere il nome dell'utente
- Scadenza

- Non permettere di usare la stessa password più volte

## Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

### *CWE*
CWE-89

### *Gravità*
Critica

### *Descrizione*
L'applicazione è costituita interamente o in parte di un commando SQL usando input estremamente influente da un altro componente, però non neutralizza o neutralizza incorrettamente elementi speciali che possono modificare il commando SQL progettato quando viene mandato.

### *Mitigazione*
Se disponibile, utilizzare meccanismi strutturati che impongono automaticamente la separazione tra dati e codice. Questi meccanismi possono essere in grado di fornire automaticamente la codifica e la convalida pertinenti, invece di affidarsi allo sviluppatore per fornire questa funzionalità in ogni punto in cui viene generato l'output.

Processare le query SQL usando Statement preparate, query parametrizzate, o procedure immagazzinate.

### *Link utili*
- https://cwe.mitre.org/data/definitions/89.html
- https://owasp.org/www-project-cheat-sheets/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet
- https://owasp.org/www-community/attacks/SQL_Injection

## Cleartext Storage of Sensitive Information

### *CWE*
CWE-312

### *Gravità*
Alta

### *Descrizione*
L'applicazione immagazzina file contenenti informazioni in chiaro che possono essere accessibili da una persona malintenzionata.

### *Mitigazione*
Non salvare mai file di configurazione o contenenti informazioni sensibili esposti sul server web.

### *Link utili*
- https://cwe.mitre.org/data/definitions/529.html

## Use of a One-Way Hash without a Salt

### *CWE*
CWE-759

### *Gravità*
Alta

### *Descrizione*
L'applicazione usa un hash one-way su un input, come per esempio una password, però non usa un sale come parte dell'input.

### *Mitigazione*
Usare un sale per criptare le password all'interno del database.

Il sale deve essere randomico e diverso per ogni password

### *Link utili*
- https://cwe.mitre.org/data/definitions/759.html
- https://www.php.net/manual/en/faq.passwords.php
- https://www.php.net/manual/en/function.password-hash.php

## Missing Authentication for Critical Function

### *CWE*
CWE-306

### *Gravità*
Alta

### *Descrizione*
L'applicazione non esegue nessuna autenticazione che richiede l'identità dell'utente.

### *Mitigazione*
Richiede sempre l'autenticazione per accedere a parti del sito critiche, come per esempio la modifica del database o la visualizzazione di dati sensibili di persone.

### *Link utili*
- https://cwe.mitre.org/data/definitions/306.html
- https://www.php.net/manual/en/features.http-auth.php

## Exposure of Backup File to an Unauthorized Control Sphere

### *CWE*
CWE-530

### *Gravità*
Media

### *Descrizione*
Un file di backup è immagazzinato in una directory che è accessibile da persone fuori dalla sfera di controllo.

### *Mitigazione*
Non salvare i file di backup su un server web accessibile dall'esterno. Una soluzione può essere salvare i file di backup su uno storage condiviso o un server interno non accessibile da persone non autorizzate.

# Cleartext Transmission of Sensitive Information

## *CWE*
CWE-319

## *Gravità*
Media

## *Descrizione*
L'applicazione trasmette dati sensibili in chiaro in un canale di comunicazione che può essere intercettato (sniffing) da una persona non autorizzata.

## *Mitigazione*
Utilizzare un canale di comunicazione criptato come per esempio HTTPS (*SSL/TLS*).

## *Link utili*
- https://cwe.mitre.org/data/definitions/319.html
- http://it.help.altervista.org/w/Accesso_al_sito_tramite_connessione_sicura_HTTPS

## Insufficient Logging

### *CWE*
CWE-778

### *Gravità*
Media

### *Descrizione*
Quando un evento critico accade, l'applicazione non registra l'evento, cosi impedendo una futura investigazione.

### *Mitigazione*
Salvare tutti gli eventi sospetti in un luogo sicuro e notificare gli amministratori.

### *Link utili*
- https://cwe.mitre.org/data/definitions/778.html
- https://www.php.net/manual/en/function.error-log.php

## Cross Site Request Forgery (CSRF)

### *CWE*
CWE-352

### *Gravità*
Media

### *Descrizione*
L'applicazione non verifica se una richiesta è stata mandata intenzionalmente dall'utente.

### *Mitigazione*
Usare un token CSRF per determinare sempre se la richiesta è stata mandata volontariamente dall'utente.

### *Link utili*
- https://cwe.mitre.org/data/definitions/352.html
- https://wiki.php.net/rfc/automatic_csrf_protection
- https://it.wikipedia.org/wiki/Cross-site_request_forgery

# Piano di mitigazione

In questa sezione verrà mostrato un semplice piano di azione per mitigare la maggior parte delle vulnerabilità riscontrate durante i test. Bisogna tenere presente che questo non è un piano completo e che il team di sviluppo ne dovrà creare uno il prima possibile.

| | |
|---|---|
| SQL injections | • Sanitizare l'input<br>• Usare query preparate |
| CSRF | • Usare i CSRF token |
| Informazioni sensibili | • Implementare HTTPS<br>• Non salvare i backup sul server esposto<br>• Togliere tutti i file contenenti informazioni sul database o altre informazioni sensibili |
| Password | • Obbligare gli utenti ad usare password robuste<br>• Implementare il sale nelle password |
| Altro | • Implementare il logging degli eventi<br>• Obbligare l'autenticazioni per sezioni del sito potenzialmente sensibili |