



# Penetration Testing & Ethical Hacking

## Tipi e Metodologie di Testing

### Parte 3

Arcangelo Castiglione  
arcastiglione@unisa.it

# Metodologie di Testing

## OWASP – Mobile Application Security (MAS)

---

- Fornisce uno standard di sicurezza per le App mobile (**MASVS**) ed una guida completa su come valutarle rispetto a tale standard (**MASTG**)
  - OWASP Mobile Application Security Verification Standard (**MASVS**)
  - OWASP Mobile Application Security Testing Guide (**MASTG**)
  - OWASP Mobile Application Security Checklist (**MAS Checklist**)
- **OWASP MASVS e OWASP MASTG** definiscono
  - I processi, le tecniche e gli strumenti da utilizzare durante la valutazione di sicurezza di una App mobile
  - Una serie di casi di test che consentono ai pentester di fornire risultati riproducibili, coerenti e completi a valle di una valutazione della sicurezza
- <https://mas.owasp.org/>

# Metodologie di Testing

## OWASP – MAS – Verification Standard

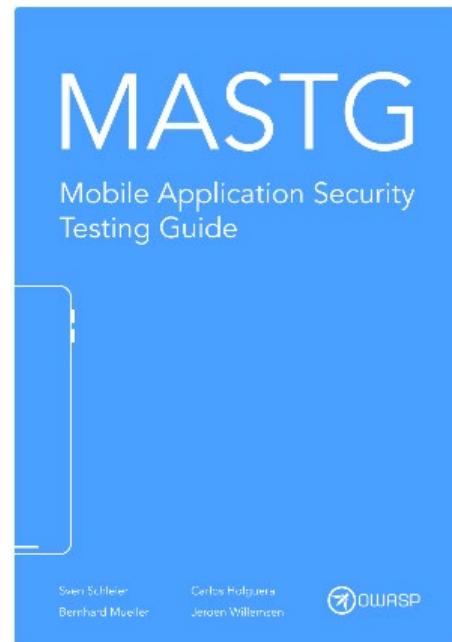
- OWASP Mobile Application Security Verification Standard (**MASVS**) è lo standard di settore per la sicurezza delle App mobile
- Può essere utilizzato da
  - Progettisti e sviluppatori di software mobile per sviluppare applicazioni sicure
  - Pentester per garantire la completezza e la coerenza dei risultati dei test di sicurezza



# Metodologie di Testing

## OWASP – MAS – Testing Guide

- La **OWASP Mobile Application Security Testing Guide (MASTG)** è un manuale completo per i test di sicurezza delle App mobile ed il reverse engineering
- Descrive i processi tecnici (**Casi di Test**) per la verifica dei controlli elencati nell'OWASP MASVS



# Metodologie di Testing

## OWASP – MAS – Checklist

- La **OWASP Mobile Application Security Checklist** permette di verificare i casi di test definiti nella **OWASP MASTG** per ciascuno dei controlli richiesti dal **OWASP MASVS**

The screenshot shows the OWASP MASVS interface for the Storage category. It displays a table with columns for MASVS-ID, Platform, Description, L1, L2, R, and Status. The table includes two rows for MASVS-STORAGE-1 and MASVS-STORAGE-2, with specific details for Android and iOS platforms.

MASVS-ID	Platform	Description	L1	L2	R	Status
<a href="#">MASVS-STORAGE-1</a>		The app securely stores sensitive data.				
	android	<a href="#">Testing the Device-Access-Security Policy</a>	Pass	Pass	Pass	Fail ▾
	android	<a href="#">Testing Local Storage for Sensitive Data</a>	Pass	Pass	Pass	N/A ▾
	ios	<a href="#">Testing Local Data Storage</a>	Pass	Pass	Pass	N/A ▾
<a href="#">MASVS-STORAGE-2</a>		The app prevents leakage of sensitive data.				
	android	<a href="#">Testing Logs for Sensitive Data</a>	Pass	Pass	Pass	Fail ▾
	android	<a href="#">Determining Whether the Keyboard Cache Is Disabled for Text Input Fields</a>	Pass	Pass	Pass	N/A ▾
	android	<a href="#">Testing Backups for Sensitive Data</a>	Pass	Pass	Pass	N/A ▾

# Metodologie di Testing

## OWASP – Top 10 Project

---

- Mostra i 10 principali rischi per la sicurezza delle Web App
- Per ciascun rischio mostra
  - I principali scenari di attacco
  - Generici metodi di attacco, indipendenti dalla tecnologia utilizzata
  - Il suo impatto tecnico ed aziendale
  - Come tale rischio potrebbe essere prevenuto
  - Riferimenti a fonti esterne utili per meglio comprendere il rischio
- Si concentra sulle macro-aree dei problemi di sicurezza delle Web App
  - Piuttosto che affrontarne i dettagli

<https://owasp.org/www-project-top-ten/>



# Metodologie di Testing

## OWASP – Top 10 Project

---

### OWASP Top Ten

[Main](#)

Translation Efforts

Sponsors

Data 2025

#### Important note:

#### OWASP Top Ten 2025

Current project status as of September 2024:

- We are planning to announce the release of the **OWASP Top 10:2025** in the first half of 2025.
- **Data Collection (Now - December 2024):** Please donate your application penetration testing statistics.

[Stay Tuned!](#)

**Ultimo aggiornamento al 2021, ma dovrebbero esserci presto novità...**

# Metodologie di Testing

OWASP – Top 10 Project

---

2021

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

· A03:2021-Injection

A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

· A07:2021-Identification and Authentication Failures

A08:2021-Software and Data Integrity Failures

· A09:2021-Security Logging and Monitoring Failures\*

A10:2021-Server-Side Request Forgery (SSRF)\*

\* From the Survey



# Metodologie di Testing

OWASP – Top 10 Project

---

2021

- A01:2021-Broken Access Control
- A02:2021-Cryptographic Failures
- A03:2021-Injection
- A04:2021-Insecure Design
- A05:2021-Security Misconfiguration
- A06:2021-Vulnerable and Outdated Components
- A07:2021-Identification and Authentication Failures
- A08:2021-Software and Data Integrity Failures
- A09:2021-Security Logging and Monitoring Failures\*
- A10:2021-Server-Side Request Forgery (SSRF)\*

\* From the Survey

# Metodologie di Testing

## OWASP – Top 10 Project

A01:2021 – Broken Access Control



### Factors

CWEs Mapped	Max Incidence Rate	Avg Incidence Rate	Avg Weighted Exploit	Avg Weighted Impact	Max Coverage	Avg Coverage
34	55.97%	3.81%	6.92	5.93	94.55%	47.72%

### Overview

Moving up from the fifth position, 94% of applications were tested for some form of broken access control with the average incidence rate of 3.81%, and has the most occurrences in the contributed dataset with over 318k. Notable Common Weakness Enumerations (CWEs) included are *CWE-200: Exposure of Sensitive Information to an Unauthorized Actor*, *CWE-201: Exposure of Sensitive Information Through Sent Data*, and *CWE-352: Cross-Site Request Forgery*.

# Metodologie di Testing

## OWASP – Top 10 Project

---

### Description

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data or performing a business function outside the user's limits. Common access control vulnerabilities include:

- Violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone.
- Bypassing access control checks by modifying the URL (parameter tampering or force browsing), internal application state, or the HTML page, or by using an attack tool modifying API requests.
- Permitting viewing or editing someone else's account, by providing its unique identifier (insecure direct object references)
- Accessing API with missing access controls for POST, PUT and DELETE.
- Elevation of privilege. Acting as a user without being logged in or acting as an admin when logged in as a user.
- Metadata manipulation, such as replaying or tampering with a JSON Web Token (JWT) access control token, or a cookie or hidden field manipulated to elevate privileges or abusing JWT invalidation.
- CORS misconfiguration allows API access from unauthorized/untrusted origins.
- Force browsing to authenticated pages as an unauthenticated user or to privileged pages as a standard user.

# Metodologie di Testing

## OWASP – Top 10 Project

---

### How to Prevent

Access control is only effective in trusted server-side code or server-less API, where the attacker cannot modify the access control check or metadata.

- Except for public resources, deny by default.
- Implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage.
- Model access controls should enforce record ownership rather than accepting that the user can create, read, update, or delete any record.
- Unique application business limit requirements should be enforced by domain models.
- Disable web server directory listing and ensure file metadata (e.g., .git) and backup files are not present within web roots.
- Log access control failures, alert admins when appropriate (e.g., repeated failures).
- Rate limit API and controller access to minimize the harm from automated attack tooling.
- Stateful session identifiers should be invalidated on the server after logout. Stateless JWT tokens should rather be short-lived so that the window of opportunity for an attacker is minimized. For longer lived JWTs it's highly recommended to follow the OAuth standards to revoke access.

Developers and QA staff should include functional access control unit and integration tests.

# Metodologie di Testing

## OWASP – Top 10 Project

### Example Attack Scenarios

**Scenario #1:** The application uses unverified data in a SQL call that is accessing account information:

```
pstmt.setString(1, request.getParameter("acct"));
ResultSet results = pstmt.executeQuery();
```

An attacker simply modifies the browser's 'acct' parameter to send whatever account number they want. If not correctly verified, the attacker can access any user's account.

```
https://example.com/app/accountInfo?acct=notmyacct
```

**Scenario #2:** An attacker simply forces browses to target URLs. Admin rights are required for access to the admin page.

```
https://example.com/app/getappInfo
https://example.com/app/admin_getappInfo
```

If an unauthenticated user can access either page, it's a flaw. If a non-admin can access the admin page, this is a flaw.

# Metodologie di Testing

## OWASP – Top 10 Project

### References

- OWASP Proactive Controls: Enforce Access Controls
- OWASP Application Security Verification Standard: V4 Access Control
- OWASP Testing Guide: Authorization Testing
- OWASP Cheat Sheet: Access Control
- OWASP Cheat Sheet: Authorization
- PortSwigger: Exploiting CORS misconfiguration
- OAuth: Revoking Access

### List of Mapped CWEs

CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

CWE-23 Relative Path Traversal

CWE-35 Path Traversal: '.../...//'

# Metodologie di Testing

## OWASP Internet of Things Project – Aspetti Chiave

---

- **Sicurezza nell'IoT:** Il progetto si concentra sull'identificazione e la mitigazione delle vulnerabilità nei dispositivi IoT
- **OWASP IoT Top 10:** Fornisce un elenco delle dieci principali minacce alla sicurezza IoT
- **Mappatura delle Minacce:** Confronta le vulnerabilità nel tempo per analizzarne l'evoluzione
- **Risorse e Strumenti:** Offre linee guida, best practice e strumenti per migliorare la sicurezza dei sistemi IoT
- **OWASP Firmware Testing:** Fornisce metodologie e strumenti per analizzare la sicurezza dei firmware nei dispositivi IoT
- **Community:** Incoraggia la partecipazione attiva per sviluppare soluzioni e migliorare la sicurezza IoT

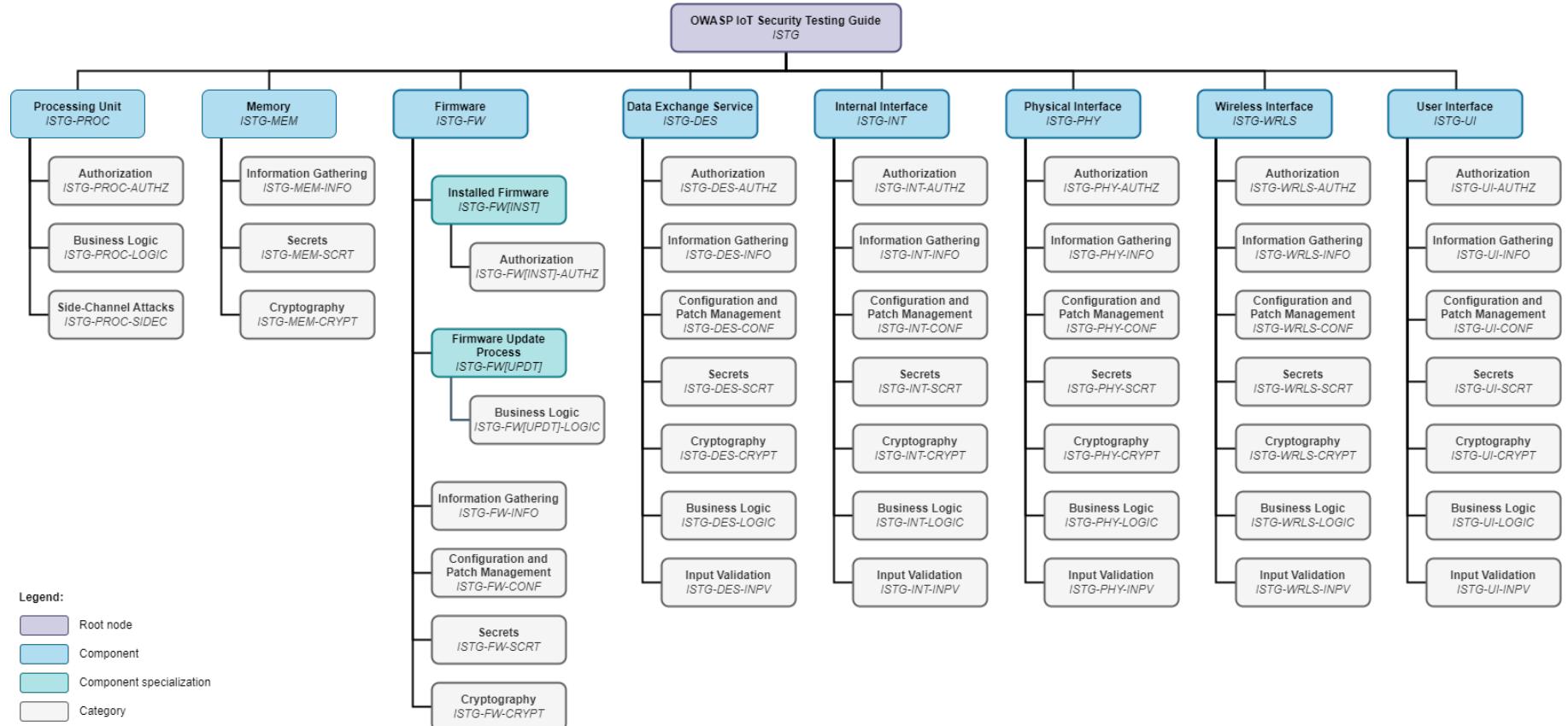
<https://owasp.org/www-project-internet-of-things/>





# Metodologie di Testing

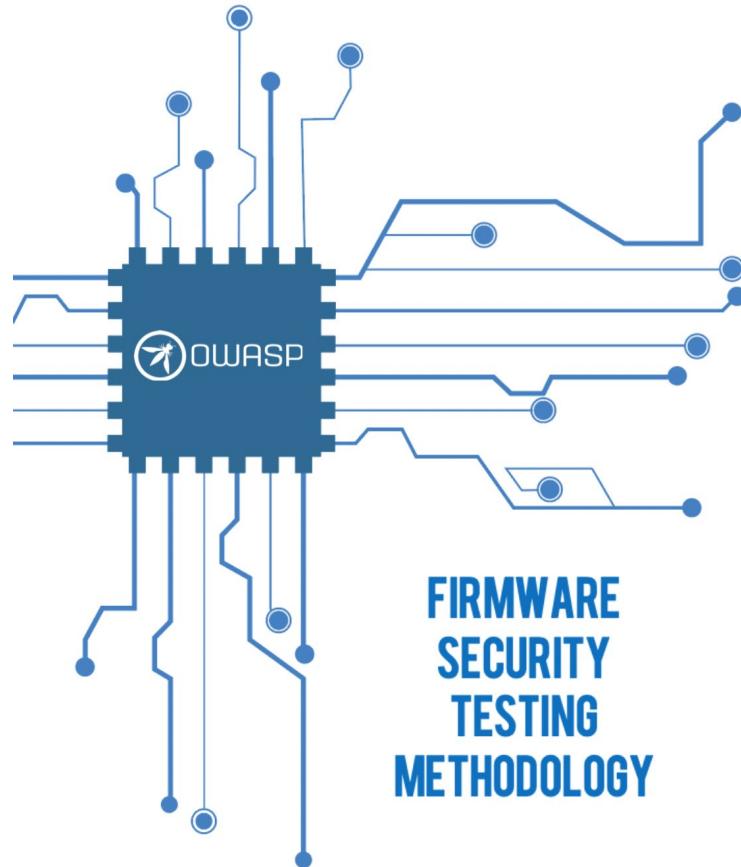
## OWASP IoT Security Testing Guide (ISTG)



<https://owasp.org/owasp-istg/>

# Metodologie di Testing

OWASP – Firmware Security Testing Methodology



<https://github.com/scriptingxss/owasp-fstm>



# Metodologie di Testing

## OWASP – Firmware Security Testing Methodology

- FSTM è composto da nove fasi che consentono la valutazione della sicurezza dei firmware

Stage	Description
1. Information gathering and reconnaissance	Acquire all relative technical and documentation details pertaining to the target device's firmware
2. Obtaining firmware	Attain firmware using one or more of the proposed methods listed
3. Analyzing firmware	Examine the target firmware's characteristics
4. Extracting the filesystem	Carve filesystem contents from the target firmware
5. Analyzing filesystem contents	Statically analyze extracted filesystem configuration files and binaries for vulnerabilities
6. Emulating firmware	Emulate firmware files and components
7. Dynamic analysis	Perform dynamic security testing against firmware and application interfaces
8. Runtime analysis	Analyze compiled binaries during device runtime
9. Binary Exploitation	Exploit identified vulnerabilities discovered in previous stages to attain root and/or code execution

# Metodologie di Testing

## OWASP – Firmware Security Testing Methodology

---

- **EmbedOS:** Macchina virtuale Ubuntu-based, preconfigurata con gli strumenti per l'analisi dei firmware utilizzati dalla Firmware Security Testing Methodology
  - <https://github.com/scriptingxss/EmbedOS>
  
- **EmbedOS** può essere scaricato tramite l'URL seguente:
  - <https://tinyurl.com/EmbedOS-2020>

# Metodologie di Testing

## OWASP – Principali Vantaggi

---

- Valutare le Web App rispetto ai 10 principali rischi di sicurezza garantisce che vengano
  - Evitati o mitigati gli attacchi derivanti dalle vulnerabilità più comuni
  - Mantenute la confidenzialità, l'integrità e la disponibilità (*triade CIA*) della Web App
  - Maggiori dettagli in seguito...



# Metodologie di Testing

OWASP – Principali Vantaggi

---

- Incoraggia pratiche di programmazione sicura, integrando la valutazione della sicurezza in ogni fase dello sviluppo di un sistema
  - Garantisce che il sistema messo in produzione sia (presumibilmente) robusto, privo di errori e sicuro
- È ampiamente accettato a livello globale
  - I primi 10 rischi sono di solito allineati con altri standard di valutazione della sicurezza delle Web App
  - Permette di ottenere contemporaneamente la conformità rispetto a più di uno standard

# Metodologie di Testing

## NIST Special Publication (SP) 800-115

---

- Fornisce linee guida per valutare la sicurezza degli asset di tutte le dimensioni e settori
  - Ogni organizzazione può utilizzare volontariamente tali le linee guida per migliorare la sicurezza del proprio asset
  - È obbligatorio che tutte le agenzie federali rispettino tali linee guida
  - Anche gli appaltatori (e subappaltatori) che lavorano con/per le agenzie federali devono rispettare tali linee guida, rischiando altrimenti di perdere il contratto
- La NIST Special Publication (SP) 800-115 fornisce in particolare
  - Linee guida tecniche per condurre attività di *penetration testing* e *vulnerability assessment*
  - Supporto nella pianificazione ed esecuzione di tali attività
- <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf>

# Metodologie di Testing

NIST Special Publication (SP) 800-115

---

- Alcuni punti salienti del NIST SP 800-115 includono
  - **Pianificazione:** Fornisce indicazioni sulle attività di pianificazione, come la definizione degli obiettivi, l'individuazione delle regole di ingaggio, l'identificazione dei ruoli e delle responsabilità del team di pentesting e lo sviluppo di piani di testing
  - **Scoperta:** Definisce tecniche per la raccolta di informazioni (*Information Gathering*), identificazione di porte/servizi, rilevamento di vulnerabilità, etc
  - **Attacco:** Definisce metodi per sfruttare le vulnerabilità, ottenere accessi non autorizzati, aumentare i privilegi, condurre attacchi *DoS* e «spostarsi» attraverso la rete (*pivoting*)
  - **Reporting:** Delinea gli elementi chiave che dovrebbero essere inclusi in un *Penetration Testing Report*, come vulnerabilità, impatto ed azioni correttive
  - **Valutazione delle Competenze:** Fornisce elementi per valutare le capacità del/dei pentester
  - **Considerazioni Legali:** Fornisce considerazioni su questioni legali e restrizioni che potrebbero applicarsi agli incarichi di penetration testing

# Metodologie di Testing

NIST Special Publication (SP) 800-115

---

- NIST SP 800-115 è suddiviso in varie sezioni che coprono diversi aspetti dei test di sicurezza
  - *Security Testing and Examination Overview*
  - *Review Techniques*
  - *Target Identification and Analysis Techniques*
  - *Target Vulnerability Validation Techniques*
  - *Security Assessment Planning*
  - *Security Assessment Execution*
  - *Post-Testing Activities*

# Metodologie di Testing

## SP 800-115 – Security Testing and Examination Overview

---

- Stabilisce che una valutazione di sicurezza dovrebbe comprendere almeno le seguenti fasi
  1. *Planning*
  2. *Execution*
  3. *Post-Execution*
- Vengono inoltre definite 3 tipologie di valutazione per un asset
  - *Testing*: Confrontare il comportamento reale con il comportamento atteso
  - *Examination*: Controllare, ispezionare, revisionare, osservare, studiare ed analizzare un *asset* per migliorarne la comprensione
  - *Interviewing*: Discutere con il personale dell'*asset* (in gruppi o individualmente) per ottenere chiarimenti

# Metodologie di Testing

## SP 800-115 – Review Techniques

---

- Vengono affrontati diversi aspetti e tecniche riguardanti la
  - Revisione della documentazione
  - Revisione dei log
  - Revisione delle regole
  - Revisione delle configurazioni

# Metodologie di Testing

SP 800-115 – Target Identification and Analysis Techniques

---

- Vengono fornite indicazioni su come identificare porte, servizi e sistemi nella rete
  - Il passo successivo consiste nell'identificare eventuali loro vulnerabilità
  
- Le tecniche trattate in questa sezione sono
  - *Network Discovery*
  - *Network Port e Service Identification*
  - *Vulnerability Scanning*
  - *Wireless Scanning (passive ed active scanning, wireless device location tracking, bluetooth scanning)*

# Metodologie di Testing

SP 800-115 – Target Vulnerability Validation Techniques

---

- Vengono fornite indicazioni su come
  - Confermare l'esistenza di una vulnerabilità
  - Comprenderne l'impatto (rischio) se la vulnerabilità viene sfruttata
  
- Tale sezione copre sia le debolezze tecniche che quelle dovute alla mancanza di consapevolezza o formazione (i.e., «*vulnerabilità umane*»)
  - *Target Exploitation*
  - *Password Cracking*
  - *Social Engineering*
  - *Etc*

# Metodologie di Testing

## SP 800-115 – Security Assessment Planning

---

- Definisce come pianificare il processo di valutazione della sicurezza
  
- Fornendo indicazioni su come
  - Sviluppare una politica di valutazione della sicurezza
  - Dare priorità e pianificare le valutazioni
  - Gestire lo sviluppo del piano di valutazione
  - Selezionare e personalizzare le tecniche di valutazione della sicurezza
  - Gestire la logistica della valutazione (selezione dei valutatori e delle loro competenze, ubicazione, strumenti, risorse, etc)
  - Affrontare gli aspetti legali

# Metodologie di Testing

## SP 800-115 – Security Assessment Execution

---

- L'esecuzione è ciò che segue la pianificazione
  
- È importante che i valutatori si attengano al piano di valutazione
  - Se è necessario «deviare» da tale piano, la situazione dovrebbe essere riesaminata per prendere nuove decisioni
  
- Questa sezione copre aspetti quali
  - Coordinamento delle risorse e delle attività coinvolte nel processo di testing
  - Valutazione ed analisi dei risultati ottenuti dal processo di testing
  - Trattamento dei dati relativi a tale processo (raccolta, archiviazione, trasmissione e distruzione)

# Metodologie di Testing

## SP 800-115 – Post-Testing Activities

---

- Riguarda ciò che accade dopo il processo di valutazione della sicurezza
- Le attività di post-testing mirano a raccogliere i risultati della sezione precedente ed a creare un piano per mitigare le vulnerabilità rilevate
  - I dati raccolti vengono «convertiti» in azioni da intraprendere
- Il NIST fornisce linee guida per le seguenti attività di post-testing
  - *Recommendation/Remediation* su come risolvere e/o mitigare le problematiche di sicurezza rilevate
  - *Reporting*

# Metodologie di Testing

## Penetration Testing Execution Standard (PTES)

- Penetration Testing Execution Standard (PTES)
  - [http://www.pentest-standard.org/index.php/Main\\_Page](http://www.pentest-standard.org/index.php/Main_Page)

**Main Page**

LOG IN

Navigation

- [Main page](#)
- [PTES Technical Guideline](#)
- [In the Media](#)
- [FAQ](#)

Search

Search The Penetration Ti

Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)

[main page](#) [view source](#) [history](#)

### High Level Organization of the Standard

The penetration testing execution standard consists of seven (7) main sections. These cover everything related to a penetration test - from the initial communication and reasoning behind a pentest, through the intelligence gathering and threat modeling phases where testers are working behind the scenes in order to get a better understanding of the tested organization, through vulnerability research, exploitation and post exploitation, where the technical security expertise of the testers come to play and combine with the business understanding of the engagement, and finally to the reporting, which captures the entire process, in a manner that makes sense to the customer and provides the most value to it.

This version can be considered a v1.0 as the core elements of the standard are solidified, and have been "road tested" for over a year through the industry. A v2.0 is in the works soon, and will provide more granular work in terms of "levels" - as in intensity levels at which each of the elements of a penetration test can be performed at. As no pentest is like another, and testing will range from the more mundane web application or network test, to a full-on red team engagement, said levels will enable an organization to define how much sophistication they expect their adversary to exhibit, and enable the tester to step up the intensity on those areas where the organization needs them the most. Some of the initial work on "levels" can be seen in the intelligence gathering section.

Following are the main sections defined by the standard as the basis for penetration testing execution:

- Pre-engagement Interactions
- Intelligence Gathering
- Threat Modeling
- Vulnerability Analysis
- Exploitation
- Post Exploitation
- Reporting

As the standard does not provide any technical guidelines as far as how to execute an actual pentest, we have also created a technical guide to accompany the standard itself. The technical guide can be reached via the link below:

- [Technical Guidelines](#)

For more information on what this standard is, please visit:

- [The Penetration Testing Execution Standard: FAQ](#)



# Metodologie di Testing

## Penetration Testing Execution Standard (PTES)

---

- Sviluppato e proposto da esperti di sicurezza per fornire una metodologia di penetration testing riproducibile ed intuitiva dal punto di vista logico, basata sostanzialmente sulle tipiche fasi effettuate dai black hat hacker
- Può essere utilizzato per eseguire un processo di penetration testing in un qualsiasi dominio applicativo
- Definisce l'intero processo di penetration testing in maniera organizzata e completa
  - Il penetration testing è composto da sette fasi
  - Fasi descritte in maniera dettagliata mediante mappe concettuali che esplicitano le azioni da compiere in ciascuna di esse
- Aiuta a garantire la coerenza tra le varie fasi del processo e tra tutte le interazioni che possono avvenire all'interno di esso
- Si tratta della metodologia generale di penetration testing più simile a quella che sarà studiata nel prosieguo del corso



# Metodologie di Testing

## Penetration Testing Execution Standard (PTES)

- Nel PTES il processo di penetration testing è composto da 7 fasi
  1. *Pre-Engagement*: Stabilisce le regole di ingaggio, l'ambito di valutazione, i meccanismi di comunicazione tra le parti e gli accordi legali
  2. *Intelligence Gathering*: Identifica e caratterizza la presenza online dell'asset, raccogliendo informazioni su nomi di dominio, blocchi di indirizzi IP, nomi/e-mail dei dipendenti e le tecnologie utilizzate
  3. *Threat Modeling*: Crea modelli per caratterizzare come gli aggressori potrebbero violare l'asset e causare danni ad esso. Utilizzato come linea guida per i test
  4. *Vulnerability Analysis*: Scopre ed analizza le vulnerabilità tecniche, come i punti deboli del sistema operativo, della rete e delle applicazioni, e ne valuta la gravità
  5. *Exploitation*: Tenta di accedere all'asset o di comprometterne il regolare funzionamento
  6. *Post-Exploitation*: Esfiltra dati dall'asset, mantiene l'accesso persistente all'asset, aumenta i privilegi all'interno dell'asset, passa ad altri sistemi (*pivoting*)
  7. *Reporting*: Documenta le vulnerabilità rilevate e quelle sfruttate, fornendo un'analisi dei risultati ottenuti e consigliando opportune strategie di mitigazione



# Metodologie di Testing

## PTES – Principali Vantaggi

---

- Framework molto accurato che copre sia aspetti tecnici che gestionali di un processo di penetration testing
- Fornisce istruzioni dettagliate su come eseguire molte delle attività necessarie per valutare accuratamente la sicurezza di un asset
- Creato da penetration tester che svolgono queste attività quotidianamente
- Riguarda sia le tecnologie più comunemente utilizzate che quelle non molto comuni
  - Anche se non copre quelle più recenti
- È facile da comprendere e può essere adattato a varie esigenze e contesti di testing



# Metodologie di Testing

## ISSAF – Caratteristiche

---

- **Information Systems Security Assessment Framework (ISSAF)**
  - Framework Open Source
  - Suddiviso in diversi **domini**, che permettono di affrontare la valutazione della sicurezza secondo un preciso ordine logico
  - Ciascuno dominio valuta una parte dell'asset da analizzare



# Metodologie di Testing

## ISSAF – Caratteristiche

---

- Il framework si focalizza su due aspetti del testing
  - **Tecnico**
    - Stabilisce l'insieme di regole e procedure da seguire
    - Crea un processo di valutazione della sicurezza adeguato
  - **Manageriale**
    - Definisce le migliori pratiche che dovrebbero essere seguite durante la gestione del processo di penetration testing



# Metodologie di Testing

## ISSAF – Caratteristiche

---

- Contiene numerosi criteri di valutazione tecnica per testare numerose tecnologie e processi
- Può essere integrato nel ciclo di vita dell'asset per soddisfare i suoi requisiti di sicurezza
- Affronta diversi aspetti della sicurezza
  - Valutazione dei rischi
  - Gestione delle risorse aziendali
  - Valutazione dei controlli di sicurezza
  - Sviluppo delle politiche di sicurezza
  - Best Practice



# Metodologie di Testing

## ISSAF – Caratteristiche

---

- Il framework può focalizzarsi su specifiche tecnologie
  - Router / Switch
  - Firewall
  - Intrusion Detection / Prevention System
  - Virtual Private Network
  - Sistemi Operativi
  - Web Application Server
  - Database
  - Etc
- **Problema:** mantenere aggiornato il framework rispetto all'introduzione di nuove tecnologie e processi



# Metodologie di Testing

## ISSAF – Principali Vantaggi

---

- Cerca di colmare il divario tra la visione tecnica e gestionale dei test di sicurezza
  - Implementando i controlli necessari per gestire entrambi gli aspetti
  
- Permette di
  - Esaminare la sicurezza di un asset
  - Proteggere l'asset valutando i controlli di sicurezza esistenti rispetto a vulnerabilità critiche
  - Comprendere i rischi esistenti in un asset e ridurli in modo proattivo
    - Identificando le vulnerabilità che possono influire sulla sicurezza dell'asset



# Metodologie di Testing

Web Application Security Consortium: Threat Classification (WASC-TC)

- **Web Application Security Consortium: Threat Classification (WASC-TC)**
- <http://www.webappsec.org/projects/threat/>

The screenshot shows a workspace interface for 'The Web Application Security Consortium'. The main content area displays the 'Threat Classification' page. The page title is 'Threat Classification' and the subtitle is 'The WASC Threat Classification v2.0'. A note states: 'The Threat Classification is an effort to classify the weaknesses, and attacks that can lead to the compromise of a website, its data, or its users.' Below this is a 'Description' section: 'The WASC Threat Classification is a cooperative effort to clarify and organize the threats to the security of a web site. The members of the Web Application Security Consortium have created this project to develop and promote industry standard terminology for describing these issues. Application developers, security professionals, software vendors, and compliance auditors will have the ability to access a consistent language and definitions for web security related issues.' There is also a 'Download' section with a link to 'PDF Version'. To the right, there is a sidebar titled 'SideBar' containing links to various WASC projects like 'Open Proxy Honeypots', 'Script Mapping', and 'Static Analysis Technologies Evaluation Criteria (NEW)'. Another sidebar lists 'WASC Project Leaders' including 'Robert Auger', 'Ryan Barnett', and 'Romain Gaucher'. The top navigation bar includes tabs for 'Wiki' and 'Pages & Files', and a search bar.

# Metodologie di Testing

Web Application Security Consortium: Threat Classification (WASC-TC)

---

- Standard Open Source per valutare la sicurezza delle Web App
  
- Simile allo standard OWASP
  - Classifica una serie di attacchi e vulnerabilità, ma li affronta in modo più approfondito

# Metodologie di Testing

Web Application Security Consortium: Threat Classification (WASC-TC)

---

- Lo standard definisce tre diverse «**view**», che permettono di valutare da diverse «prospettive» le principali minacce di sicurezza per le Web App
  - *Enumeration View*
  - *Development View*
  - *Taxonomy Cross-reference View*

# Metodologie di Testing

WASC-TC – Enumeration View

---

- Fornisce una lista (enumerazione) delle principali «debolezze» e dei principali attacchi per le Web App
  
- Debolezze ed attacchi sono discussi individualmente (non a livello di macro-aree), fornendo per ciascuno di essi
  - Definizione concisa
  - Tipologia
  - Esempi su varie piattaforme di programmazione

# Metodologie di Testing

WASC-TC – Development View

---

- Fornisce allo sviluppatore una visione più completa sulla sicurezza di un determinato asset
  
- Definisce le vulnerabilità a partire da un insieme di debolezze ed attacchi che possono verificarsi in una delle seguenti fasi del ciclo di vita di una Web App
  1. Progettazione
  2. Sviluppo
  3. Distribuzione

# Metodologie di Testing

WASC-TC – Development View

---

- **Vulnerabilità di Progettazione:** introdotte quando le problematiche di sicurezza della Web App non sono state tenute in considerazione durante la fase di raccolta dei requisiti
  
- **Vulnerabilità di Implementazione:** si verificano a causa di regole e pratiche di programmazione sbagliate o non sicure
  
- **Vulnerabilità di Distribuzione:** causate dell'errata configurazione della Web App, del Web server o di altri sistemi ad essi relativi

# Metodologie di Testing

WASC-TC – Taxonomy Cross-reference View

---

- Permette di «mappare» la terminologia usata da uno standard in quella usata da un altro standard
  - Talvolta per avere la conformità rispetto a più standard
  - Ciascuno standard definisce i propri criteri per valutare le Web App sotto diversi punti di vista e misura i rischi associati alle vulnerabilità
- Permette di valutare in maniera approfondita le Web App rispetto alle debolezze ed agli attacchi più comuni
- WASC-TC è accettato a livello industriale ed è utilizzato in molte soluzioni sia Open Source che commerciali

# Outline

---

- Terminologia
- Tipologie di Test di Sicurezza
- Tipi di Penetration Testing
- Metodologie di Testing
- **Framework Generale per il Penetration Testing (FGPT)**
- Penetration Testing Report

# Framework Generale per il Penetration Testing (FGPT)

---

- Kali Linux fornisce numerosi strumenti per condurre varie tipologie di test di sicurezza su un determinato asset
- L'uso di questi strumenti richiede un approccio strutturato
  - Framework secondo il quale tali strumenti possono operare
- Formalizzare il processo di penetration testing attraverso un approccio strutturato è estremamente importante
  - Sia dal punto di vista tecnico che gestionale



# Framework Generale per il Penetration Testing (FGPT)

---

- Il framework definisce i passi da seguire durante un processo di penetration testing per valutare la sicurezza di un asset in modo efficace
  - Fornisce una panoramica delle tipiche fasi che un pentester dovrebbe condurre
- Include sia le tecnologie più comunemente utilizzate che quelle meno note



# Framework Generale per il Penetration Testing (FGPT)

---

- È di facile apprendimento e può essere adattato a varie esigenze di testing
- Permette di realizzare sia approcci di tipo Black Box che White Box
- Ciascuno di questi approcci può essere «specializzato» in base all'asset da valutare
- Approccio generale che può essere
  - Combinato con una qualsiasi delle metodologie esistenti
  - Usato come linea guida tecnica ed operativa



# Framework Generale per il Penetration Testing (FGPT)

---

- Il FGPT è costituito dalle seguenti fasi, tipicamente sequenziali
  1. *Target Scoping*
  2. *Information Gathering*
  3. *Target Discovery*
  4. *Enumerating Target*
  5. *Vulnerability Mapping*
  6. *Social Engineering*
  7. *Target Exploitation*
  8. *Privilege Escalation*
  9. *Maintaining Access*
  10. *Documentation and Reporting*

*Target Post-Exploitation*



# Framework Generale per il Penetration Testing (FGPT)

---

- Un «qualsiasi sottoinsieme» di queste fasi può essere utilizzato sia in approcci di tipo Black Box che White Box
  
- Il pentester deve scegliere il migliore percorso di testing in base
  - Alle richieste del committente
  - Alla tipologia ed alla complessità dell'asset
  - Alle informazioni disponibili sull'asset prima dell'inizio del processo di penetration testing
  - Alle risorse che ha a disposizione (budget, tempo, personale, etc)



# FGPT

## Target Scoping

---

- Si occupa di comprendere l'ambito ed i «confini» del penetration testing
  
- Durante questa fase vengono tipicamente prese le seguenti decisioni
  - Cosa deve essere analizzato?
  - Come deve essere analizzato?
  - Quali condizioni devono essere applicate durante il processo di test?
  - Cosa limiterà l'esecuzione del processo di test?
  - Quanto in termini di risorse e tempo ci vorrà per completare il test?
  - Quali obiettivi tecnici/aziendali saranno raggiunti?



# FGPT

## Target Scoping

---

- Per condurre efficacemente un processo di penetration testing il pentester dovrebbe conoscere i seguenti fattori
  - Tecnologia che sta valutando
  - Funzionalità di base di tale tecnologia
  - Interazione di tale tecnologia con l'ambiente esterno
- La competenza e l'esperienza del pentester contribuiscono in maniera significativa al successo di un qualsiasi tipo di valutazione della sicurezza



# FGPT

## Information Gathering

---

- Il pentester per «conoscere meglio» il suo obiettivo (asset) consulta una serie di risorse pubblicamente disponibili
  - Forum
  - Bacheche
  - Albi
  - Articoli
  - Blog
  - Social Network
  - Siti Web
  - Etc



# FGPT

## Information Gathering

---

- Informazioni possono anche essere raccolte attraverso motori di ricerca
  - Google, Yahoo!, Microsoft Bing, Baidu, Yandex Search, etc
- Un pentester può utilizzare gli strumenti forniti da Kali Linux per raccogliere quante più informazioni possibili su un determinato asset
  - Informazioni di rete, Whois, Informazioni sul DNS e sugli spazi di indirizzamento
  - Indirizzi e-mail e numeri di telefono
  - Informazioni personali
  - Account utente
  - Etc



# FGPT

## Information Gathering

---

- Man mano che vengono raccolte ulteriori informazioni aumenta la probabilità di condurre con successo il processo di penetration testing
  
- Altra importante fonte di informazioni è il **Dark Web**
  - Tipicamente accessibile tramite TOR Browser
  - Contiene molte informazioni utili su vulnerabilità, exploit, etc
  - Ad es., la ricerca nel Dark Web può fornire una visione più esaustiva sulle vulnerabilità e le minacce per un determinato asset



# FGPT

## Target Discovery

---

- Permette di
  - Determinare gli host attivi all'interno dell'asset ed i sistemi operativi in esecuzione su tali host
  - Caratterizzare ciascun host in base al proprio ruolo all'interno dell'architettura di rete
- Fornisce una visione completa delle tecnologie e dei dispositivi interconnessi in un determinato asset
- Gli strumenti per il target discovery generalmente implementano tecniche di rilevamento attivo e passivo



# FGPT

## Enumerating Target

- Utilizza numerose tecniche per la scansione delle porte
- Rileva le «porte aperte» sui sistemi analizzati
  - Le porte rilevate come «aperte» possono essere enumerate in base ai servizi che esse erogano
- Utile per valutare la «visibilità» delle porte anche se l'host è protetto da firewall o *Intrusion Detection System (IDS)*



# FGPT

## Enumerating Target

- I servizi associati alle porte aperte verranno ulteriormente analizzati per rilevare le vulnerabilità dell'asset
- Questa fase rappresenta il primo passo per la ricerca delle vulnerabilità nelle componenti dell'asset analizzato



# FGPT

## Vulnerability Mapping

---

- Identifica ed analizza le vulnerabilità in base alle porte aperte ed ai servizi erogati dall'asset
  
- Fase che può essere condotta tramite due approcci
  - Strumenti automatici
  - Manualmente
  
- La combinazione dei due approcci permette al pentester di esaminare sia vulnerabilità note che sconosciute (*0-day*)



# FGPT

## Social Engineering

---

- Praticare l'«arte dell'inganno» può essere «molto utile» quando non vengono rilevati punti di accesso (vulnerabilità sfruttabili dal punto di vista informatico) nell'asset
- Rappresenta un'ulteriore opportunità da sfruttare per tentare di «violare» l'asset
  - Ingannando un utente attraverso l'esecuzione di codice dannoso che potrebbe consentire l'accesso all'asset stesso
- Può essere condotta come un'attività a sé stante



# FGPT

## Social Engineering

---

- Può essere attuato in varie forme, non solo digitali
  - Ad esempio, imitando il personale per entrare in un luogo fisico
- Ampia casistica di possibilità che potrebbero essere messe in atto per raggiungere l'obiettivo richiesto



# FGPT

## Social Engineering

---

- Condurre un attacco efficace potrebbe richiedere tempo
  - Necessario per comprendere la psicologia dell'obiettivo ed applicare la forma di «inganno» più adatta nei suoi confronti
- **N.B.** Fondamentale comprendere appieno le leggi nazionali ed internazionali in materia di social engineering prima di intraprendere questa fase
  - Che dovrebbe essere espressamente richiesta ed autorizzata dal committente



# FGPT

## Target Exploitation

---

- Dopo aver esaminato le vulnerabilità esistenti in un asset si cerca di «violarle» attraverso la rete, sfruttando opportuni vettori di attacco (*exploit remoti*)
- Un pentester potrebbe anche utilizzare *exploit client-side* per assumere il controllo di un determinato asset
  - Veicolati alla vittima tramite tecniche di ingegneria sociale
- Potrebbero essere necessarie ulteriori ricerche o modifiche agli exploit esistenti per farli funzionare correttamente



# FGPT

## Target Exploitation

---

### ➤ Questa fase

- Si concentra principalmente sul processo di «acquisizione» dell'asset analizzato
  - Per assumerne il controllo o per causare malfunzionamenti ad esso
  - È strettamente relata alle attività di *Post-Exploitation*
    - *Privilege Escalation*
    - *Maintaining Access*



# FGPT

## Privilege Escalation

---

- Una volta «acquisito» l'asset, un pentester potrebbe operare all'interno di esso
  - In base a determinati privilegi di accesso
  
- I privilegi potrebbero anche essere «aumentati» utilizzando opportuni strumenti
  - Che ad esempio permettono di ottenere i permessi di *super-user* (*root*) o di *amministratore* di sistema



# FGPT

## Privilege Escalation

---

- Lo scopo dell'attività di Privilege Escalation è quello di ottenere l'accesso all'asset disponendo dei massimi permessi possibili
- Questa attività può essere di portata *limitata* o *non limitata*, a seconda dello scopo del testing



# FGPT

## Privilege Escalation

---

- Dopo aver ottenuto l'accesso ad alcune componenti dell'asset, un pentester potrebbe
  - Acquisire ulteriori informazioni/permessi/visibilità sull'asset
    - Utilizzando *exploit locali*
    - Analizzando il traffico di rete (*Sniffing*)
    - Effettuando il «cracking» delle password di alcuni servizi
    - Sfruttando errate o improprie configurazioni dell'asset
    - Effettuando keylogging
    - Etc
  - Condurre ulteriori attacchi verso altre componenti dell'asset
    - *Pivoting*



# FGPT

## Maintaining Access

---

- Potrebbe essere necessario mantenere l'accesso all'asset per un determinato periodo di tempo (*persistenza*)
  - Ad esempio, per dimostrare l'accesso «non autorizzato» all'asset senza eseguire nuovamente l'intero processo di penetration testing
  
- Ciò consente di risparmiare tempo, costi e risorse per dimostrare l'accesso all'asset



# FGPT

## Maintaining Access

---

- Tipicamente, l'accesso persistente all'asset è mantenuto mediante software chiamati *backdoor*
  
- Questo tipo di accesso fornisce una visione chiara di come un attaccante potrebbe mantenere la propria persistenza all'interno dell'asset
  - Spesso, senza che ciò venga rilevato



# FGPT

## Documentation and Reporting

- Documentare, riportare e presentare le vulnerabilità rilevate e sfruttate
  - Penetration Testing Report
  - Rapporto di Scansione Dettagliato
  - Presentazione Digitale
- Fondamentale sia dal punto di vista etico che professionale
  - L'analisi delle vulnerabilità può permettere di risolverle o mitigarle



# FGPT

## Documentation and Reporting

---

- I report creati possono essere di diverso tipo
  - A seconda di chi dovrà utilizzarli per comprendere ed analizzare i punti deboli presenti nell'asset
  
- I report permettono anche di stabilire e confrontare la sicurezza dell'asset analizzato, prima e dopo il processo di penetration testing



# Outline

---

- Terminologia
- Tipologie di Test di Sicurezza
- Tipi di Penetration Testing
- Metodologie di Testing
- Framework Generale per il Penetration Testing (FGPT)
- Penetration Testing Report

# Penetration Testing Report

## Struttura – Cover Page

---

- Dovrebbe includere dettagli quali
  - Eventuali loghi delle entità (aziende, enti, etc) coinvolte nel processo di penetration testing
  - Titolo
  - Breve descrizione del processo effettuato



## Penetration Test Report

MegaCorp One

August 10<sup>th</sup>, 2013

# Penetration Testing Report

## Struttura – Table of Contents

- Indice che permette di leggere anche solo determinate parti del penetration testing report

**Table of Contents**

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report .....	4
Remediation Report .....	4
Findings Summary .....	5
Detailed Summary .....	5
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability .....	6
E3 – Stored Cross Site Scripting Vulnerability .....	8
E4 – Blind XSS Vulnerability .....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure .....	16
E8 – Administrative Login And Database Manipulation .....	17

# Penetration Testing Report

## Struttura – Executive Summary

### Table of Contents

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report.....	4
Remediation Report.....	4
Findings Summary.....	5
Detailed Summary .....	5
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability.....	6
E3 – Stored Cross Site Scripting Vulnerability.....	8
E4 – Blind XSS Vulnerability.....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure.....	16
E8 – Administrative Login And Database Manipulation .....	17

# Penetration Testing Report

## Struttura – Executive Summary

---

- Parte più importante del penetration testing report
  
- Rivolto alla parte gestionale dell'asset che ha commissionato il processo di penetration testing
  
- Scritto per rivolgersi ad un pubblico non tecnico
  - Deve essere facilmente comprensibile da esso

# Penetration Testing Report

## Struttura – Executive Summary

---

- Tipicamente la parte gestionale di un asset ha poco tempo a disposizione per leggere i report e non ha competenze tecniche
  - L'Executive Summary deve essere preciso e conciso
- L'Executive Summary dovrebbe iniziare con la definizione dello Scopo/Ambito del processo di penetration testing e del modo in cui tale processo è stato condotto
  - Lo Scopo (o ambito) deve essere definito in modo molto preciso

# Penetration Testing Report

## Struttura – Executive Summary

---

- In questa sezione andrebbero
  - Spiegati i risultati ottenuti dal processo di penetration testing e le eventuali scoperte
  - Discusse, in generale, le problematiche di sicurezza rilevate, le relative cause ed eventuali contromisure

# Penetration Testing Report

## Struttura – Executive Summary

---

- Andrebbe poi inserita la parte di analisi, che dovrebbe evidenziare
- Rischio complessivo per l'asset, determinato in base ai risultati ottenuti dal processo di penetration testing
- Diminuzione del rischio dopo aver affrontato le problematiche di sicurezza ed implementato le opportune contromisure

# Penetration Testing Report

## Struttura – Executive Summary – Esempio

### EXECUTIVE SUMMARY

**RHAinfoSec** conducted a full webapplication penetration test on **foonetworks**, the goal was to analyze the security posture of the Webapplications and suggest countermeasures for all the findings requiring remediation.

The Application Penetration test was conducted on foonetworks from January 2013 onwards. The target subdomains were also included in the scope of penetration test, which were not provided by default since it was a full black box penetration test.

As a result of the engagement we managed to find lots of high risk vulnerabilities which confirmed that the security posture of the application is very low and proper security countermeasures have not been implemented inside the environment.

This report contains detailed analysis about the vulnerabilities that we found during the engagement along with the report also contains a remediation report which would help you improve the overall security posture of your application. The report also contains a detailed explanation about every vulnerability found along with the detailed countermeasures to fix the vulnerability.

The overall risk of compromise was analyzed to be 70%. Addressing the security issues that present inside the report would significantly increase the overall risk of compromise.

# Penetration Testing Report

## Struttura – Executive Summary – Esempio

### EXECUTIVE SUMMARY

**RHAinfoSec** conducted a full webapplication penetration test on **foonetwoks**, the goal was to analyze the security posture of the Webapplications and suggest countermeasures for all the findings requiring remediation.

The Application Penetration test was conducted on foonetwoks from January 2013 onwards. The target subdomains were also included in the scope of penetration test, which were not provided by default since it was a full black box penetration test.

As a result of the engagement we managed to find lots of high risk vulnerabilities which confirmed that the security posture of the application is very bad and proper security countermeasures have not been implemented inside the environment.

This report contains detailed analysis about the vulnerabilities found during the engagement along with the report also contains a remediation section to help you improve the overall security posture of your application. The report also contains an explanation about every vulnerability found along with the details of how the vulnerability was exploited.

**Definizione dello scopo/ambito del testing e del modo in cui è stato realizzato**

The overall risk of compromise was analyzed to be 70%. Addressing the security issues that are present inside the report would significantly increase the overall risk of compromise.

# Penetration Testing Report

## Struttura – Executive Summary – Esempio

### EXECUTIVE SUMMARY

**RHAinfoSec** conducted a full webapplication penetration test on **foonetworks**, the goal was to analyze the security posture of the Webapplications and suggest countermeasures for all the findings requiring remediation.

The Application Penetration test was conducted on foonetworks from January 2013 onwards. The target subdomains were also included in the scope of penetration test, which were not provided by default since it was a full black box penetration test.

As a result of the engagement we managed to find lots of high risk vulnerabilities which confirmed that the security posture of the application is very low and proper security countermeasures have not been implemented inside the environment.

This report contains detailed analysis about the vulnerabilities that we found during the engagement along with the report also contains a remediation report which would help you improve the overall security posture of your application. The report also contains a detailed explanation about every vulnerability found along with the details of the vulnerability.

The overall risk of compromise was analyzed to be 70%. Additional vulnerabilities present inside the report would significantly increase the overall risk of compromise.

**Descrizione generale dei risultati del penetration testing e delle problematiche rilevate**

# Penetration Testing Report

## Struttura – Executive Summary – Esempio

### EXECUTIVE SUMMARY

RHAinfoSec conducted a full analysis of the application to analyze the security posture of the application and identify findings requiring remediation.

The Application Penetration test was conducted on the target subdomains.

The target subdomains were also included in the penetration test, which were not provided by default since it was a full black box penetration test.

As a result of the engagement we managed to find lots of high risk vulnerabilities which confirmed that the security posture of the application is very low and proper security countermeasures have not been implemented inside the environment.

#### ➤ Parte di analisi

- **Descrizione dei rischi in base ai risultati del testing**
- **In che modo il rischio diminuirà dopo aver implementato le appropriate contromisure**

This report contains detailed analysis about the vulnerabilities that we found during the engagement along with the report also contains a remediation report which would help you improve the overall security posture of your application. The report also contains a detailed explanation about every vulnerability found along with the detailed countermeasures to fix the vulnerability.

The overall risk of compromise was analyzed to be 70%. Addressing the security issues that present inside the report would significantly increase the overall risk of compromise.

# Penetration Testing Report

## Struttura – Engagement Highlights

### Table of Contents

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report.....	4
Remediation Report.....	4
Findings Summary.....	5
Detailed Summary .....	5
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability.....	6
E3 – Stored Cross Site Scripting Vulnerability.....	8
E4 – Blind XSS Vulnerability.....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure .....	16
E8 – Administrative Login And Database Manipulation .....	17



# Penetration Testing Report

## Struttura – Engagement Highlights

---

### ➤ Pre-Ingaggio

- Vengono discusse tra le parti coinvolte i requisiti legali e le «Regole di Ingaggio»

### ➤ Le Regole di Ingaggio definiscono

- Come deve essere condotto il processo di penetration testing
- Quale metodologia deve essere utilizzata
- Le date di inizio e fine del processo
- Gli obiettivi del processo
- Gli obblighi e le responsabilità delle parti coinvolte nel processo
- Etc

# Penetration Testing Report

## Struttura – Engagement Highlights

---

- Tutte le **Regole di Ingaggio** devono essere **concordate** tra le parti **prima dell'inizio** del processo di **penetration testing**
  
- Le **Regole di Ingaggio** dovrebbero definire almeno i seguenti aspetti
  - Accordo di «Non Divulgazione» (*Non-Disclosure Agreement - NDA*)
  - Portata del processo di penetration testing
    - Parti dell'asset che devono essere valutate e come devono esserlo
  - Tecniche consentite e non consentite
  - Strumenti consentiti e non consentiti

# Penetration Testing Report

## Struttura – Vulnerability Report

### Table of Contents

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report .....	4
Remediation Report.....	4
Findings Summary.....	5
Detailed Summary .....	5
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability.....	6
E3 – Stored Cross Site Scripting Vulnerability.....	8
E4 – Blind XSS Vulnerability.....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure .....	16
E8 – Administrative Login And Database Manipulation .....	17



# Penetration Testing Report

## Struttura – Vulnerability Report

---

### ➤ Descrizione

- Generale (non tecnica) delle vulnerabilità
- Del come tali vulnerabilità vanno ad impattare sulla sicurezza dell'asset

# Penetration Testing Report

## Struttura – Remediation Report

### Table of Contents

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report .....	4
<b>Remediation Report.....</b>	<b>4</b>
Findings Summary.....	5
Detailed Summary .....	5
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability.....	6
E3 – Stored Cross Site Scripting Vulnerability.....	8
E4 – Blind XSS Vulnerability.....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure .....	16
E8 – Administrative Login And Database Manipulation .....	17



# Penetration Testing Report

## Struttura – Remediation Report

---

- Raccomandazioni generali da implementare per migliorare la sicurezza dell'asset
- Rivolto a chi si occupa di stabilire dal punto di vista manageriale le politiche di sicurezza dell'asset
  - Deve essere molto preciso e di facile comprensione

# Penetration Testing Report

## Struttura – Remediation Report – Esempio

### REMEDIATION

The security control environment for foonetworks was found very poor, as a result of which there are certain security countermeasures we would like to suggest. With the goal of protecting the Web application's infrastructure, we would recommend you to perform the following actions.

- A perfect plan for fixing the Critical, High, Medium, low risk vulnerabilities should designed and implemented. The vulnerabilities should be fixed in the descending order of priority.
- Secure development life cycle (SDLC) for developing web applications shall be implemented.
- A Web Application Firewall shall be implemented to detect, filter and block all the malicious packets.
- Security Audits shall be performed on the regular basis.
- Early security checks should be performed in the development process.

# Penetration Testing Report

## Struttura – Findings Summary

### Table of Contents

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report.....	4
Remediation Report.....	4
<b>Findings Summary.....</b>	<b>5</b>
Detailed Summary .....	5
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability.....	6
E3 – Stored Cross Site Scripting Vulnerability.....	8
E4 – Blind XSS Vulnerability.....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure .....	16
E8 – Administrative Login And Database Manipulation .....	17



# Penetration Testing Report

## Struttura – Findings Summary

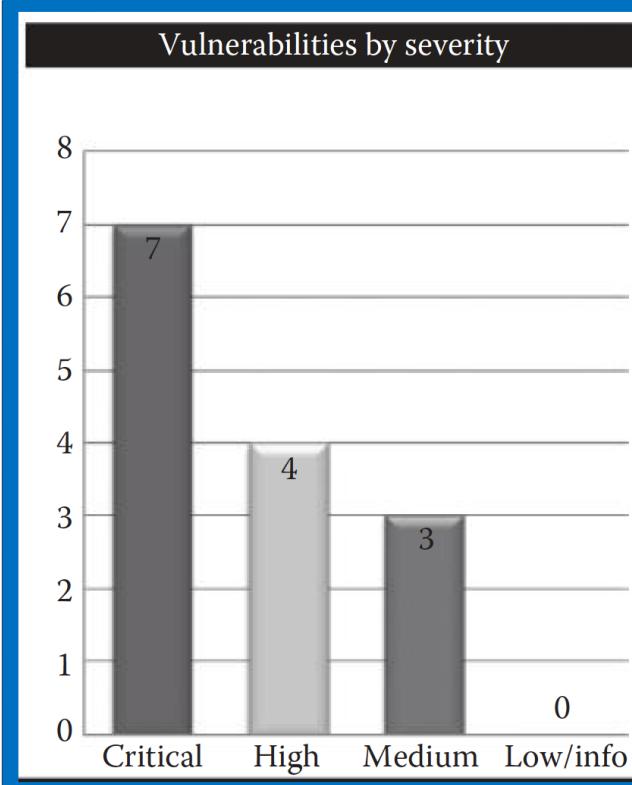
---

- In questa parte del report vengono presentati, usando un maggiore livello di dettaglio, i risultati ottenuti dal processo di penetration testing
  
- Utilizzo di grafici per permettere una migliore comprensione delle vulnerabilità rilevate
  
- I responsabili tecnici della sicurezza dell'asset potrebbero essere interessati a questa parte del report
  - Per poter applicare le adeguate contromisure tecniche

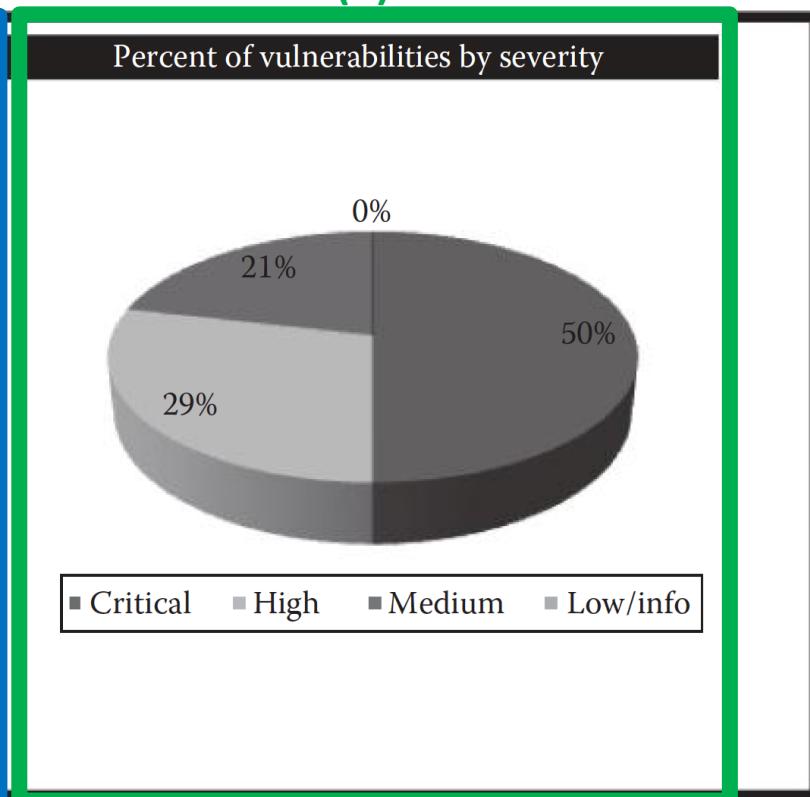
# Penetration Testing Report

## Struttura – Findings Summary – Elementi Tipici

(a)



(b)



Il primo grafico (a) mostra il numero di vulnerabilità sulla base della loro gravità (severity), il secondo (b) in base alla loro percentuale

# Penetration Testing Report

## Struttura – Findings Summary – Elementi Tipici

- **Vulnerabilities breakdown:** numero di vulnerabilità che sono state rilevate per ciascun host dell'asset

Vulnerabilities breakdown						
S #	IP Address	Hostname	Critical	High	Medium	Low/Info
1	192.254.236.66	Services.rafayhackingarticles.net	3	14	7	0
2	192.254.236.67	Tools.rafayhackingarticles.net	2	6	4	0

**Host**      **Numero e «severity» delle vulnerabilità**

# Penetration Testing Report

## Struttura – Findings Summary – Elementi Tipici

- Valutazione della vulnerabilità sulla base del numero di host attivi
- Numero di vulnerabilità e relativo livello di rischio per tutti gli host attivi
  - *High*
  - *Medium*
  - *Info*

Category	Description		
Systems vulnerability assessment summary			
Number of live hosts	50		
Number of vulnerabilities	29		
High, medium, and info severity vulnerabilities	14	6	9

# Penetration Testing Report

## Struttura – Findings Summary – Elementi Tipici

### ➤ Hazard Risk Assessment Matrix

- Strumento visuale che mostra la probabilità e la gravità (*impatto*) causato da un potenziale rischio che è stato identificato
- Di solito utilizzato dalle aziende per stimare i rischi (tipicamente di business) che potrebbero verificarsi a seguito di determinati eventi
- Per ogni rischio identificato permette di assegnare un punteggio in base alla sua posizione all'interno della matrice
- Consente di dare una valutazione («un peso») ai rischi derivanti dallo sfruttamento di una determinata vulnerabilità

Frequency of Occurrence	Hazard Categories			
	1 Catastrophic	2 Critical	3 Serious	4 Minor
(A) Frequent	1A	2A	3A	4A
(B) Probable	1B	2B	3B	4B
(C) Occasional	1C	2C	3C	4C
(D) Remote	1D	2D	3D	4D
(E) Improbable	1E	2E	3E	4E

Probabilità che un evento si verifichi

Impatto che tale evento avrebbe dal punto di vista della sicurezza

Legend:  
■ Unacceptable ■ High ■ Medium ■ Low

# Penetration Testing Report

## Struttura – Detailed Summary

### Table of Contents

Executive Summary .....	3
Engagement Highlights .....	3
Vulnerability Report.....	4
Remediation Report.....	4
Findings Summary.....	5
<b>Detailed Summary .....</b>	<b>5</b>
E1 – DOM Based XSS Vulnerability .....	5
E2 – Stored Cross Site Scripting Vulnerability.....	6
E3 – Stored Cross Site Scripting Vulnerability.....	8
E4 – Blind XSS Vulnerability.....	10
E5 – Arbitrary File Upload Vulnerability .....	12
E6 – SOAP Based SQL Injection Vulnerability .....	13
E7 – Configuration File Disclosure .....	16
E8 – Administrative Login And Database Manipulation .....	17



# Penetration Testing Report

## Struttura – Detailed Summary

---

- Rivolto ai responsabili della sicurezza ed agli sviluppatori dell'organizzazione che ha commissionato il penetration testing
- In questa sezione, per ciascuna vulnerabilità, andrebbe descritto in maniera tecnicamente dettagliata, utilizzando un'opportuna **scheda**
  - Come è stata scoperta la vulnerabilità
  - Quali sono le cause alla base della vulnerabilità
  - Quali sono i rischi associati alla vulnerabilità
  - Quali sono le contromisure per eliminare o mitigare tali rischi
- È fortemente consigliato **caratterizzare visualmente i rischi** in base al loro livello, utilizzando opportuni colori
  - Ad esempio, utilizzando il colore **rosso** per i **rischi più gravi**

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

<b>DOM Based Cross Site Scripting Vulnerability</b>
<b>Affected Hosts:</b> foonetworks.com
<b>Risk:</b> Critical
<p><b>Description:</b> A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when the user supplied input passed through a source is not filtered/escaped before it's passed through a vulnerable sink.</p> <p><b>Explanation:</b> A dynamic file is being included which handles "location.hash" on the document object model (DOM).</p> <p><a href="http://foonetworks.com/engine.js">http://foonetworks.com/engine.js</a></p> <p>The following lines indicate the vulnerable code:</p> <p><b>Lines: 410 – 411:</b></p> <pre>if(t!=undefined){window.location.hash=t;}); \$(window).bind("load",function() {if(window.location.hash){var _9=window.location.hash.substring(1);})</pre>
<b>Risk</b> <p>Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.</p>
<b>Recommendations:</b> <p>Any user-generated input should be HTML-encoded at any point where it is copied into application responses.</p> <p>All HTML metacharacters should be replaced with the corresponding HTML entities.</p>

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

DOM Based Cross Site Scripting Vulnerability	
<b>Affected Hosts:</b> FOONETWORKS.COM	
<b>Risk:</b> Critical	
<b>Description:</b> A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when user supplied input passed through a source is not filtered/escaped before it's passed through the DOM.	
<b>Explanation:</b> A dynamic file is being included which handles "location.hash" on the document (DOM).  <b>http://foonetworks.com/engine.js</b>  The following lines indicate the vulnerable code:  <b>Lines: 410 – 411:</b>  <code>if(t!=undefined){window.location.hash=t;}); \$(window).bind("load",function() {if(window.location.hash){var _9=window.location.hash.substring(1);})</code>	
<b>Risk</b>  Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.	
<b>Recommendations:</b>  Any user-generated input should be HTML-encoded at any point where it is copied into application responses.  All HTML metacharacters should be replaced with the corresponding HTML entities.	

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

DOM Based Cross Site Scripting Vulnerability	
<b>Affected Hosts:</b>	foonetworks.com
<i>Risk:</i>	Critical
<b>Description:</b>	A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when user supplied input passed through a source is not filtered/escaped before it's passed through another part of the application.
<b>Explanation:</b>	A dynamic file is being included which handles "location.hash" on the document (DOM).  <a href="http://foonetworks.com/engine.js">http://foonetworks.com/engine.js</a>  The following lines indicate the vulnerable code:  <b>Lines: 410 – 411:</b>  <code>if(t!=undefined){window.location.hash=t;}); \$(window).bind("load",function() {if(window.location.hash){var _9=window.location.hash.substring(1);})</code>
<b>Risk</b>	Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it to the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.
<b>Recommendations:</b>	Any user-generated input should be HTML-encoded at any point where it is copied into application responses.  All HTML metacharacters should be replaced with the corresponding HTML entities.

Host affetto dalla  
vulnerabilità

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

### DOM Based Cross Site Scripting Vulnerability

Affected Hosts: foonetworks.com

Risk: Critical

Description: A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when the user supplied input passed through a source is not filtered/escaped before it's passed through a vulnerable sink.

Explanation: A dynamic file is being included which handles "location.hash" on the (DOM).

<http://foonetworks.com/engine.js>

The following lines indicate the vulnerable code:

Lines: 410 – 411:

```
if(t!=undefined){window.location.hash=t;});  
$(window).bind("load",function()  
{if(window.location.hash){var _9=window.location.hash.substring(1);}}
```

#### Risk

Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.

#### Recommendations:

Any user-generated input should be HTML-encoded at any point where it is copied into application responses.

All HTML metacharacters should be replaced with the corresponding HTML entities.

Livello di rischio derivante  
dalla vulnerabilità

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

### DOM Based Cross Site Scripting Vulnerability

Affected Hosts: foonetworks.com

Risk: Critical

**Description:** A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when the user supplied input passed through a source is not filtered/escaped before it's passed through a vulnerable sink.

**Explanation:** A dynamic file is being included which handles "location.hash" on the document object model (DOM).

<http://foonetworks.com/engine.js>

The following lines indicate the vulnerable code:

**Lines: 410 – 411:**

```
if(t!=undefined){window.location.hash=t;});  
$(window).bind("load",function()  
{if(window.location.hash){var _9=window.location.hash.substring(1);}}
```

#### Risk

Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.

#### Recommendations:

Any user-generated input should be HTML-encoded at any point where it is copied into application responses.

All HTML metacharacters should be replaced with the corresponding HTML entities.

**Descrizione generale della vulnerabilità**

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

<p><b>DOM Based Cross Site Scripting Vulnerability</b></p> <p><b>Affected Hosts:</b> foonetworks.com</p> <p><b>Risk:</b> Critical</p> <p><b>Description:</b> A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when the user supplied input passed through a source is not filtered/escaped before it's passed through a vulnerable site.</p> <p><b>Explanation:</b> A dynamic file is being included which handles "location.hash" on the document object model (DOM).</p> <p><a href="http://foonetworks.com/engine.js">http://foonetworks.com/engine.js</a></p> <p>The following lines indicate the vulnerable code:</p> <p><b>Lines: 410 – 411:</b></p> <pre>if(t!=undefined){window.location.hash=t;}); \$(window).bind("load",function() {if(window.location.hash){var_9=window.location.hash.substring(1);}})</pre> <p><b>Risk</b></p> <p>Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.</p> <p><b>Recommendations:</b></p> <p>Any user-generated input should be HTML-encoded at any point where it is copied into application responses.</p> <p>All HTML metacharacters should be replaced with the corresponding HTML entities.</p>	<p><b>Descrizione dettagliata della vulnerabilità</b></p>
--	---

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

### DOM Based Cross Site Scripting Vulnerability

**Affected Hosts:** foonetworks.com

*Risk: Critical*

**Description:** A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when supplied input passed through a source is not filtered/escaped before it's passed through a vulnerable part of the application.

**Explanation:** A dynamic file is being included which handles "location.hash" on the document (DOM).

<http://foonetworks.com/engine.js>

The following lines indicate the vulnerable code:

**Lines: 410 – 411:**

```
if(t!=undefined){window.location.hash=t;});  
$(window).bind("load",function()  
{if(window.location.hash){var _9=window.location.hash.substring(1);})
```

#### Risk

Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.

#### Recommendations:

Any user-generated input should be HTML-encoded at any point where it is copied into application responses.

All HTML metacharacters should be replaced with the corresponding HTML entities.

**Descrizione dettagliata  
dei rischi derivanti dalla  
vulnerabilità**

# Penetration Testing Report

## Struttura – Detailed Summary – Esempio

### DOM Based Cross Site Scripting Vulnerability

Affected Hosts: foonetworks.com

Risk: Critical

Description: A DOM Based XSS is a type of Cross site scripting vulnerability which occurs when the user supplied input passed through a source is not filtered/escaped before it's passed through a vulnerable sink.

Explanation: A dynamic file is being included which handles "location.hash" on the document object model (DOM).

<http://foonetworks.com/engine.js>

The following lines indicate the vulnerable code:

Lines: 410 – 411:

```
if(t!=undefined){window.location.hash=t;});  
$(window).bind("load",function()  
{if(window.location.hash){var _9=window.location.hash.substring(1);})
```

#### Risk

Since javascript can access the DOM, an attacker can craft a special piece of javascript that would be able to steal the authentication cookies and send it the domain that he controls. In case of a DOM based XSS, the payload is always executed on the client side, this means this makes it difficult to trace the attacker from the forensics perspective, since the attack vector would not appear inside the log file.

#### Recommendations:

Any user-generated input should be HTML-encoded at any point where it is copied into application responses.

All HTML metacharacters should be replaced with the corresponding HTML entities.

**Descrizione delle raccomandazioni  
che possono essere messe in atto  
per risolvere la vulnerabilità**

# Penetration Testing Report

## Esempio (PT su un Asset a scelta dallo Studente)

UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica



Corso di Laurea in Informatica

### Frontespizio

Penetration Testing and Ethical Hacking

Penetration Testing Report  
Wakanda1

**Committente**

**Autore**

Anno Accademico 2019/2020

**Tipi e Metodologie di Testing**

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Executive Summary (1/3)

---

### 1. Executive Summary

In questo documento sono riportati i risultati del penetration testing sulla macchina “Wakanda1”.

L’obiettivo di tale attività è stato di scoprire, analizzare, sfruttare e documentare il maggior numero possibile di vulnerabilità e di riuscire a leggere il contenuto dei tre file flag1.txt, flag2.txt e root.txt.

L’attività di penetration testing è stata avviata il giorno 10 Novembre 2019 ed è stata conclusa il giorno 28 Dicembre 2019. L’attività è stata compiuta da un unico pentester, autore di questo report. Inoltre, si è utilizzato un approccio **black-box**: all’avvio dell’attività, il pentester non possiede alcuna informazione circa l’asset da testare, simulando completamente il comportamento di un utente malintenzionato.

# Penetration Testing Report

## Esempio – Executive Summary (2/3)

---

Tutte le attività sono state condotte in modo da simulare un attore malintenzionato impegnato in un attacco contro la macchina Wakanda1. In particolare:

1. Ricerca di vulnerabilità all'interno del sistema manualmente e con software open-source.
2. Verificare se un attaccante remoto può penetrare le difese della macchina Wakanda1.
3. Determinare l'impatto di una violazione di sicurezza su:
  - a. Riservatezza e dati presenti sulla macchina Wakanda1.
  - b. Violazione di sistemi interni ed accesso ad una shell sulla macchina Wakanda1.

# Penetration Testing Report

## Esempio – Executive Summary (3/3)

---

La valutazione è stata condotta in conformità alle raccomandazioni delineate nel NIST SP 800-115<sup>1</sup> con tutti i test e le azioni condotti in condizioni controllate. L'attività di penetration testing ha permesso di portare alla luce le vulnerabilità e i punti deboli del sistema. In generale, le vulnerabilità individuate possono consentire ad un utente malintenzionato di ottenere un controllo parziale o completo del sistema.

Le vulnerabilità emerse sono di varia natura e vanno ad impattare sull'*affidabilità*, sull'*integrità* e sulla *confidenzialità* del sistema.

In questo documento verranno analizzate le vulnerabilità individuate durante l'attività di penetration testing e verranno presentate possibili contromisure per mitigare la criticità e diminuire sensibilmente il livello di rischio (che allo stato attuale del sistema è **ALTO**), al fine di ottenere un livello di sicurezza complessivo almeno accettabile.

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Engagement Highlights (1/3)

---

L'attività di penetration testing è stata eseguita utilizzando il sistema operativo **Kali Linux**, installato su una macchina collegata alla stessa rete locale della macchina testata **Wakanda**. Poiché si è nell'ambito di un progetto universitario, non sono state poste particolari limitazioni sui tool da utilizzare. Per questo motivo sono stati utilizzati alcuni dei software con licenza gratuita messi a disposizione da Kali Linux.

A fronte della situazione sopra citata non è necessario creare un canale di comunicazione tra il pentester e il cliente e nemmeno ci si è posto dei limiti sull'impatto del testing vista la natura non critica del sistema target, il quale, girerà stesso sulla macchina del pentester. Inoltre, a scopo didattico sono state installate delle **backdoor** sulla macchina target per garantire un accesso permanente. Nel documento **Penetration Testing Narrative** sono stati documentati i passi eseguiti dal pentester per l'installazione di una OS Backdoor e una Web Backdoor persistenti.

# Penetration Testing Report

## Esempio – Engagement Highlights (2/3)

La metodologia utilizzata per condurre il test consiste in un framework generico composto dalle seguenti fasi:



*(Tutte le fasi precedenti a quella di Reporting sono riportate nel documento Penetration Testing Narrative).*

# Penetration Testing Report

## Esempio – Engagement Highlights (3/3)

---

Quindi, non dovendo definire regole di ingaggio concordate tra le parti interessate prima dell'inizio del test, il pentester ha avuto piena libertà nella scelta delle metodologie, dei tempi, degli strumenti e delle tecniche. Inoltre, non è presente ovviamente alcun accordo di non divulgazione (**NDA**).

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Vulnerability Report (1/3)

---

### 3. Vulnerability Report

Le vulnerabilità individuate durante il processo di penetration testing possono essere suddivise in due categorie principali:

1. Vulnerabilità causate da errori di configurazione del sistema.
2. Vulnerabilità relative a software presenti nel sistema.

# Penetration Testing Report

## Esempio – Vulnerability Report (2/3)

---

Nella prima categoria rientrano tutte quelle vulnerabilità causate da errate scelte di configurazione del sistema. Tali scelte espongono il sistema al rischio di attacchi provenienti dall'esterno, che possono consentire ad un utente malintenzionato di ottenere informazioni sensibili o di accedere e controllare da remoto il sistema.

Nella seconda categoria rientrano tutte quelle vulnerabilità riguardanti software e servizi messi a disposizione dal sistema. Sulla macchina server sono presenti versioni obsolete di alcuni software. Tali versioni risultano essere affette da diverse vulnerabilità, le quali possono essere sfruttate per sferrare attacchi di varia natura.

# Penetration Testing Report

## Esempio – Vulnerability Report (3/3)

---

Di seguito sono elencate le principali vulnerabilità riscontrate:

- Versione di Apache non aggiornata – **Rischio ALTO**
  - La versione di Apache 2.4.10 presente attualmente nel sistema risulta avere diverse vulnerabilità che un utente malintenzionato potrebbe sfruttare per creare disservizi.
- Versione di OpenSSH non aggiornata – **Rischio MEDIO**
  - La versione di OpenSSH presente attualmente nel sistema è la 6.7p1. Tale versione presenta diverse vulnerabilità che possono causare perdita di informazioni sensibili e disservizi.
- Alcune credenziali d'accesso “nascoste” vengono rese note nel sorgente di una pagina php.
- Il file /usr/bin/pip fornisce il permesso di esecuzione anche ad utenti non amministratori il che permette ad un utente malintenzionato di poter effettuare privilege escalation.

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Remediation Report (1/3)

---

### 4. Remediation Report

L'attività di penetration testing ha evidenziato che il livello di sicurezza complessivo della macchina analizzata è abbastanza basso. Al fine di garantire una sicurezza più elevata, si raccomanda di adottare le seguenti contromisure:

- Introdurre misure di sicurezza più rigorose al fine di garantire una maggiore protezione dei dati sensibili degli utenti e dei fruitori del sistema.

# Penetration Testing Report

## Esempio – Remediation Report (2/3)

---

- Risolvere tutte le vulnerabilità presentate in questo documento, seguendo un ordine decrescente in base alla gravità: è consigliato dunque risolvere quanto prima le vulnerabilità critiche e procedere successivamente alla correzione delle vulnerabilità con criticità più bassa. In generale, si raccomanda di mettere in atto i seguenti interventi:
  - aggiornamento delle versioni dei software ritenuti a rischio.
  - riconfigurazione di alcuni servizi del sistema.
  - rimuovere informazioni sensibili da pagine o file facilmente consultabili da estranei.

# Penetration Testing Report

## Esempio – Remediation Report (3/3)

---

- Prestare attenzione ai permessi che vengono forniti ai vari file nel sistema tenendo in considerazione il **principio dei privilegi minimi**<sup>2</sup>.
- Pianificare periodicamente dei *Security Audits* al fine di valutare regolarmente il grado di sicurezza e la conformità agli standard del sistema.

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Findings Summary (1/5)

---

### 5. Findings Summary

Le analisi portate avanti con gli strumenti scelti (Nessus e OpenVas) hanno evidenziato: 21 Vulnerabilità, di cui 4 gravi, 16 a rischio medio/alto e 1 a basso rischio.

Invece, l'analisi manuale delle vulnerabilità ha permesso di rilevare 6 vulnerabilità a rischio medio.

# Penetration Testing Report

## Esempio – Findings Summary (2/5)

---

Le vulnerabilità riscontrate sono classificate in base alla gravità, suddivise su una scala con quattro livelli di gravità:

- **High**: Rischio critico, vulnerabilità molto grave per il sistema. ( $CVSS^3 \geq 7.5$ )
- **Medium/High**: Rischio elevato, vulnerabilità grave per il sistema.  
 $(6.5 \leq CVSS < 7.5)$
- **Medium**: Rischio medio, vulnerabilità potenzialmente grave per il sistema.  
 $(4 \leq CVSS < 6.5)$
- **Low**: Rischio basso, vulnerabilità non particolarmente grave per il sistema.  
 $(CVSS < 4)$

# Penetration Testing Report

## Esempio – Findings Summary (3/5)

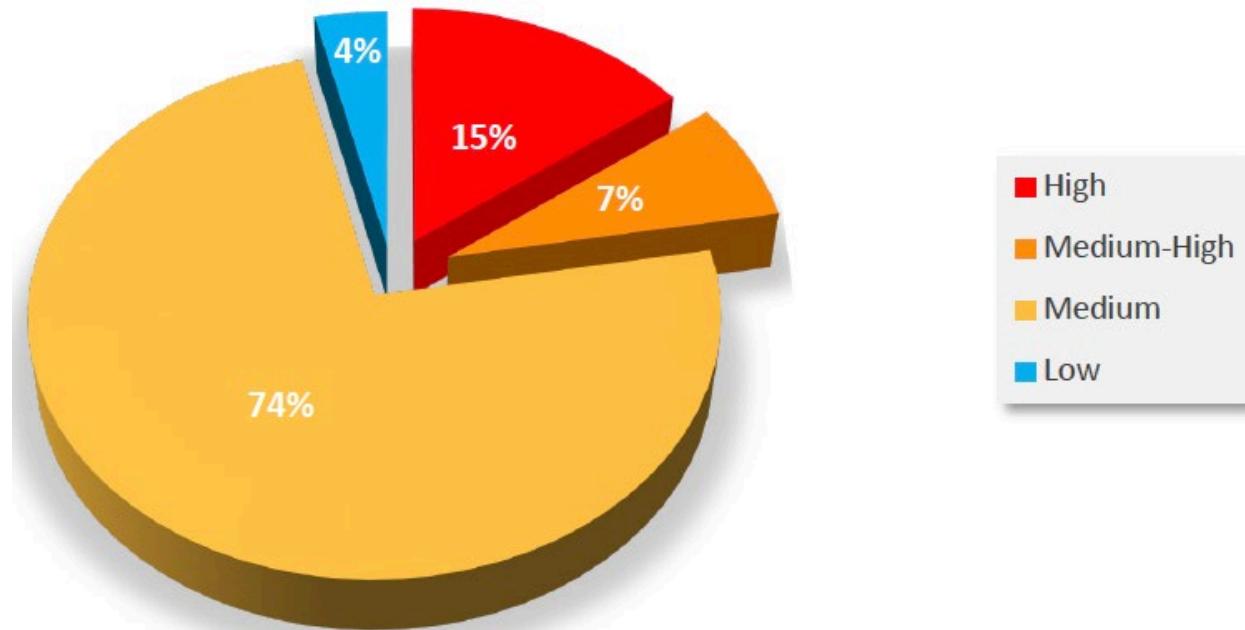
---

Le vulnerabilità rilevate sono riferite al server Apache e al servizio OpenSSH.

I grafici seguenti mostrano in maniera schematica la distribuzione delle vulnerabilità per categoria:

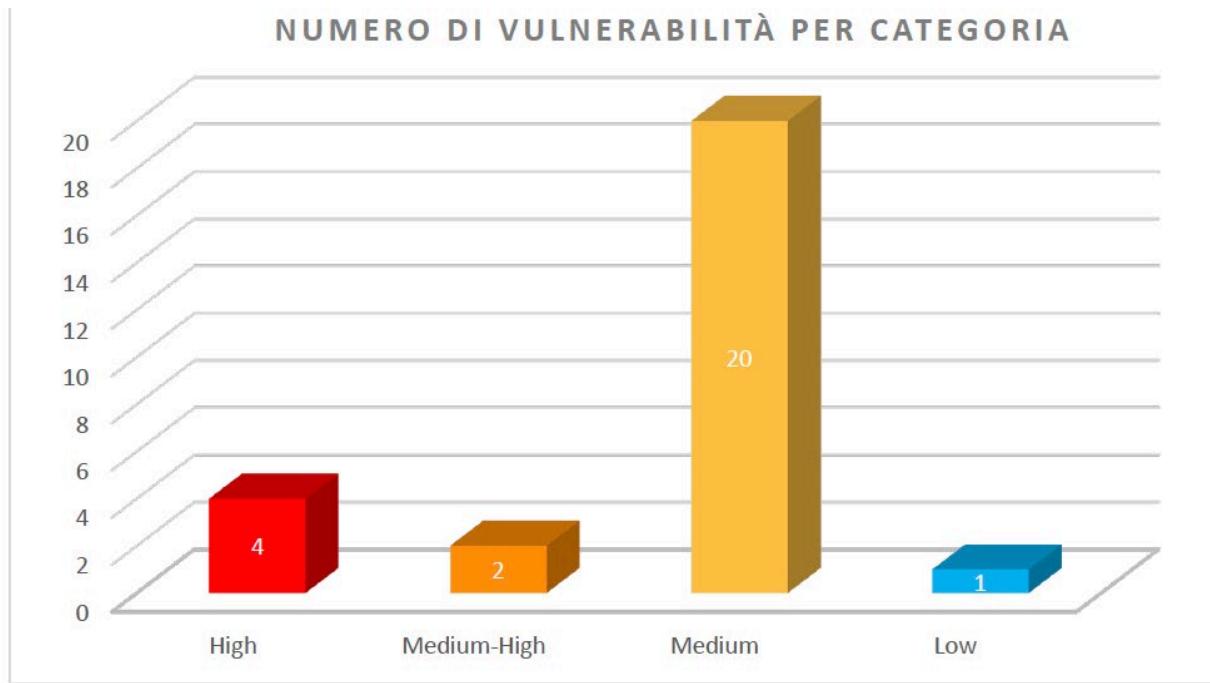
# Penetration Testing Report

## Esempio – Findings Summary (4/5)



# Penetration Testing Report

## Esempio – Findings Summary (5/5)



# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Detailed Summary (1/10)

---

<b>6. Detailed Summary .....</b>	<b>10</b>
<b>6.1 High.....</b>	<b>10</b>
<b>Apache HTTPD: ap_get_basic_auth_pw() security bypass.....</b>	<b>10</b>
<b>Apache HTTPD: mod_ssl denial of service.....</b>	<b>11</b>
<b>Apache HTTPD: ap_find_token() denial of service.....</b>	<b>12</b>
<b>Apache HTTPD: mod_mime information disclosure.....</b>	<b>13</b>
<b>6.2 Medium/High .....</b>	<b>14</b>
<b>Apache HTTPD: FilesMatch security bypass.....</b>	<b>14</b>
<b>Apache HTTPD: HTTP Digest security bypass .....</b>	<b>14</b>
<b>6.3 Medium .....</b>	<b>15</b>
<b>Apache HTTPD: mod_auth_digest denial of service.....</b>	<b>15</b>
<b>Apache HTTP Server: mod_rewrite open redirect.....</b>	<b>16</b>
<b>Apache HTTP Server: mod_proxy_fcgi denial of service.....</b>	<b>16</b>

# Penetration Testing Report

## Esempio – Detailed Summary (2/10)

---

<b>6.4 Low .....</b>	<b>31</b>
<b>Apache HTTP Server: Tampering of mod_session data for CGI applications.....</b>	<b>31</b>
<b>6.5 Others Vulnerability.....</b>	<b>32</b>
<b>Novell Netware 6.5: OpenSSH Remote Stack Overflow.....</b>	<b>32</b>
<b>PHP File Inclusion .....</b>	<b>32</b>
<b>Fakepip: privilege escalation .....</b>	<b>33</b>
<b>X-Frame-Options Header Not Set.....</b>	<b>33</b>

# Penetration Testing Report

## Esempio – Detailed Summary (3/10)

### 6.1 High

Apache HTTPD: ap_get_basic_auth_pw() security bypass.	CVE 2017-3167 [1]
<b>HIGH</b>	
<b>Descrizione:</b> Nelle versioni di Apache httpd 2.2.x precedenti alla 2.2.33 e nelle versioni 2.4.x precedenti alla 2.4.26, l'uso di ap_get_basic_auth_pw () da parte di moduli di terze parti al di fuori della fase di autenticazione può comportare il bypass dei requisiti di autenticazione. Nel sistema è installata la versione 2.4.10.	
<b>Impatto:</b> È stato scoperto che l'uso della funzione ap_get_basic_auth_pw () di httpd al di fuori della fase di autenticazione potrebbe portare al bypass dell'autenticazione stessa. Un utente malintenzionato da remoto potrebbe utilizzare questa vulnerabilità per bypassare l'autenticazione richiesta se l'API è stata utilizzata in modo errato da uno dei moduli utilizzati da httpd.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema ( $\geq 2.4.26$ ). Per le versioni di Apache httpd 2.2.x precedenti alla 2.2.33 in alternativa è possibile installare una patch di sicurezza [2].	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (4/10)

Apache HTTPD: mod_ssl denial of service.	CVE 2017-3169 [3]
HIGH	
<b>Descrizione:</b> Nelle versioni di Apache httpd 2.2.x precedenti alla 2.2.33 e nelle versioni 2.4.x precedenti alla 2.4.26, mod_ssl può fare riferimento a un puntatore NULL quando i moduli di terze parti chiamano ap_hook_process_connection() durante una richiesta HTTP.	
<b>Impatto:</b> È stato trovato un difetto di dereferenziazione del puntatore NULL nel modulo mod_ssl di httpd. Un utente malintenzionato remoto può utilizzare questo difetto per causare l'arresto anomalo di un processo figlio di httpd se un altro modulo utilizzato da httpd ha chiamato una determinata funzione API durante l'elaborazione di una richiesta HTTPS.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema ( $\geq 2.4.26$ ). Per le versioni di Apache httpd 2.2.x precedenti alla 2.2.33 in alternativa è possibile installare una patch di sicurezza [4].	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (5/10)

### 6.2 Medium/High

Apache HTTPD: FilesMatch security bypass.	CVE 2017-15715 [9]
MEDIUM-HIGH	
<b>Descrizione:</b> In Apache httpd dalla versione 2.4.0 alla versione 2.4.29, l'espressione specificata in <FilesMatch> potrebbe far corrispondere '\$' a un carattere newline in un filename dannoso, anziché corrispondere solo alla fine del filename.	
<b>Impatto:</b> Questo potrebbe essere sfruttato in ambienti in cui i caricamenti di alcuni file sono bloccati esternamente, ma solo abbinando la parte finale del filename.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema ( $\geq 2.4.30$ ).	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (6/10)

<b>Apache HTTPD: HTTP Digest security bypass.</b>	<b>CVE</b>
	<b>2018-1312 [10]</b>
<b>MEDIUM-HIGH</b>	
<b>Descrizione:</b> In Apache httpd dalla versione 2.2.0 alla versione 2.4.29, durante la generazione di una Digest access authentication, il nonce inviato per prevenire replay attack non è stato generato correttamente utilizzando un seme pseudo-casuale.	
<b>Impatto:</b> In un cluster di server che utilizza una configurazione di autenticazione Digest comune, le richieste HTTP possono essere riprodotte su tutti i server da un utente malintenzionato senza essere rilevate.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema ( $\geq 2.4.30$ ).	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (7/10)

### 6.3 Medium

Apache HTTPD: mod_auth_digest denial of service.	CVE 2017-9788 [11]
MEDIUM	
<b>Descrizione:</b> Nelle versioni di Apache httpd precedenti alla 2.2.34 e 2.4.x precedenti alla 2.4.27, il segnaposto del valore [Proxy-] nell' header di autorizzazione di tipo 'Digest' non era inizializzato o resettato prima o tra le successive assegnazioni key=value di mod_auth_digest.	
<b>Impatto:</b> Fornire una chiave iniziale senza assegnazione '=' potrebbe riflettere il valore non aggiornato della memoria del pool non inizializzata utilizzata dalla richiesta precedente, causando la perdita di informazioni potenzialmente riservate e un segfault in altri casi con conseguente negazione del servizio.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema ( $\geq$ 2.4.27).	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (8/10)

	<b>CVE</b>
<b>Apache HTTP Server:</b> mod_rewrite open redirect.	<b>2019-10098 [12]</b>
<b>MEDIUM</b>	
<b>Descrizione:</b> Nelle versioni da 2.4.0 a 2.4.39 del server Apache HTTP, i reindirizzamenti configurati con mod_rewrite che intendevano essere autoreferenziali potrebbero essere ingannati da newline codificate e reindirizzare invece a un URL inaspettato.	
<b>Impatto:</b> L'attaccante potrebbe sfruttare questa vulnerabilità per reindirizzare ad un URL diverso da quello presente nella richiesta.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema (> 2.4.39).	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (9/10)

### 6.4 Low

<b>Apache HTTP Server:</b> Tampering of mod_session data for CGI applications.	<b>CVE</b> <b>2018-1283 [34]</b>
<b>LOW</b>	
<b>Descrizione:</b> In Apache httpd dalla versione 2.4.0 alla 2.4.29, quando mod_session è configurato per inoltrare i dati della sua sessione ad applicazioni di CGI(SessionEnv abilitato, di default non è abilitato), un utente remoto può influenzare il proprio contenuto usando un “Session” Header.	
<b>Impatto:</b> Un utente malintenzionato potrebbe sfruttare tale vulnerabilità per manomettere i dati della sessione da trasmettere ad applicazioni di CGI.	
<b>Soluzione:</b> Aggiornare la versione di Apache installata nel sistema (> 2.4.29).	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite il software OpenVas.	

# Penetration Testing Report

## Esempio – Detailed Summary (10/10)

### 6.5 Others Vulnerability

Di seguito verranno mostrate altre vulnerabilità riscontrate durante l'analisi manuale della macchina target.

Novell Netware 6.5: OpenSSH Remote Stack Overflow.	CVE -
<b>Descrizione:</b> Questa vulnerabilità consente di eseguire codice arbitrario su installazioni vulnerabili di Novell Netware. Per sfruttare questa vulnerabilità è necessaria l'autenticazione.	
<b>Impatto:</b> Un utente malintenzionato potrebbe eseguire del codice malevolo su installazioni vulnerabili di Novell Netware. È disponibile un exploit per tale vulnerabilità [35].	
<b>Soluzione:</b> Aggiornare Novell Netware presente nel sistema alla versione più recente.	
<b>Metodo di detection:</b> Vulnerabilità individuata tramite analisi manuale.	

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
<b>7. Appendix .....</b>	<b>35</b>
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Appendice (1/2)

---

### 7. Appendice

#### PHP file inclusion

Per costringere PHP a codificare in base64 il file **index.php** si è utilizzata la seguente funzione: **php://filter/convert.base64-encode**.

Per effettuare l'operazione sopra descritta si è utilizzato il seguente comando:

```
curl http://10.0.2.7/?lang=php://filter/convert.base64encode/resource=index
```

# Penetration Testing Report

## Esempio – Appendice (2/2)

---

### FakePip

Una dimostrazione di come è possibile sfruttare tale vulnerabilità è documentata nel documento **Penetration Testing Narrative** oppure è disponibile al link:  
<https://github.com/0x00-0x00/FakePip>.

### Demonstration

```
example@desktop:/home/ubuntu$ sudo -l
Matching Defaults entries for example on desktop:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User example may run the following commands on desktop:
    (root : root) NOPASSWD: /usr/bin/pip install *
example@desktop:/home/ubuntu$ █
```

# Penetration Testing Report

## Esempio – Table of Contents

---

### Table of Contents

Table of Contents .....	2
1. Executive Summary .....	4
2. Engagement Highlights .....	5
3. Vulnerability Report .....	6
4. Remediation Report .....	7
5. Findings Summary.....	8
6. Detailed Summary .....	10
7. Appendix.....	35
PHP file inclusion.....	35
FakePip .....	36
Riferimenti .....	37

# Penetration Testing Report

## Esempio – Riferimenti

---

### Riferimenti:

- [1] <https://www.cvedetails.com/cve/CVE-2017-3167/>
- [2] [https://www.apache.org/dist/httpd/patches/apply\\_to\\_2.2.32/CVE-2017-3167.patch](https://www.apache.org/dist/httpd/patches/apply_to_2.2.32/CVE-2017-3167.patch)
- [3] <https://www.cvedetails.com/cve/CVE-2017-3169/>
- [4] [https://www.apache.org/dist/httpd/patches/apply\\_to\\_2.2.32/CVE-2017-3169.patch](https://www.apache.org/dist/httpd/patches/apply_to_2.2.32/CVE-2017-3169.patch)
- [5] <https://www.cvedetails.com/cve/CVE-2017-7668/>
- [6] [https://www.apache.org/dist/httpd/patches/apply\\_to\\_2.2.32/CVE-2017-7668.patch](https://www.apache.org/dist/httpd/patches/apply_to_2.2.32/CVE-2017-7668.patch)

# Penetration Testing Report

## Guide

---

### ➤ **Offensive Security – Penetration Testing Report**

➤ <https://www.offensive-security.com/reports/sample-penetration-testing-report.pdf>



## Penetration Test Report

MegaCorp One

August 10<sup>th</sup>, 2013

### Offensive Security Services, LLC

19706 One Norman Blvd.  
Suite B #253  
Carmel, NC 28031  
United States of America

Tel: 1-402-608-1337  
Fax: 1-704-625-3787  
Email: [info@offsec.com](mailto:info@offsec.com)  
Web: <http://www.offensive-security.com>

# Penetration Testing Report

## Guide

---

- **SANS Institute - Writing a Penetration Testing Report**
- <https://www.sans.org/white-papers/33343/>



**SANS Institute**  
Information Security Reading Room

### Writing a Penetration Testing Report

---

Mansour Alharbi

Copyright SANS Institute 2019. Author Retains Full Rights.  
This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express  
written permission.

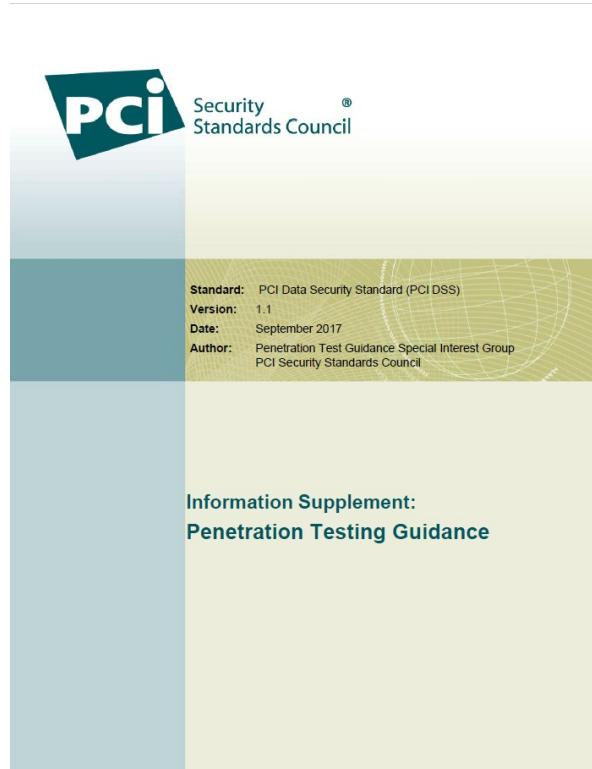
# Penetration Testing Report

## Guide

---

### ➤ PCI Security – Penetration Testing Guidance

➤ [https://www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1\\_1.pdf](https://www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf)



# Penetration Testing Report

## Guide

- **Vulnerabilityassessment.co.uk**
- <http://www.vulnerabilityassessment.co.uk/report%20template.html>

The screenshot shows a website header with the logo 'VulnerabilityAssessment.co.uk' and a blue ribbon graphic. Below the header is a grey navigation bar with the name 'Kevin Orrey'. The main content area displays a hierarchical navigation menu:

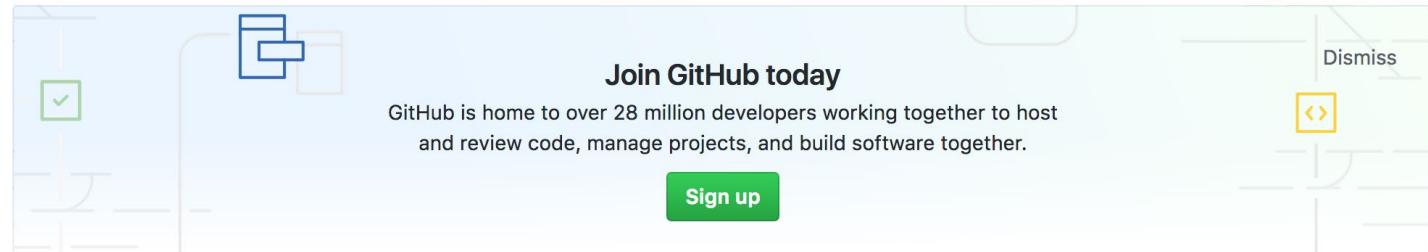
- Report Template
  - Introduction
    - Date carried out
  - Testing Team details
    - Name
    - Contact Nos.
    - Relevant Experience if required.
  - Network Details
    - Peer to Peer, Client-Server, Domain Model, Active Directory integrated
    - Number of Servers and workstations
    - Operating System Details
    - Major Software Applications
    - Hardware configuration and setup
    - Interconnectivity and by what means i.e. T1, Satelite, Wide Area Network, Lease Line Dial up etc.
    - Encryption/ VPN's utilised etc.
    - Role of the network or system
  - Scope of test
    - Constraints and limitations imposed on the team i.e. Out of scope items, hardware, IP addresses.
    - Constraints, limitations or problems encountered by the team during the actual test

# Penetration Testing Report

## Lista Pubblica di Pentesting Report

➤ <https://github.com/juliocesarfot/public-pentesting-reports>

The screenshot shows the GitHub repository page for 'juliocesarfot / public-pentesting-reports'. At the top, there's a navigation bar with tabs for 'Code', 'Issues 4', 'Pull requests 2', 'Projects 0', and 'Insights'. To the right of the tabs are buttons for 'Watch 285', 'Star 2,272', and 'Fork 676'. Below the navigation bar, there's a large 'Join GitHub today' banner with a 'Sign up' button.



Curated list of public penetration test reports released by several consulting firms and academic security groups

Branch: master	New pull request	Find file	Clone or download
juliocesarfot Merge pull request #25 from glerchundi/add-coredns-report	...	Latest commit 0c995df on 14 Sep 2018	
Bugcrowd	Add reports from Instructure's public security reports: https://www.c...	3 years ago	
Coinspect	Adding Zcash reports	2 years ago	

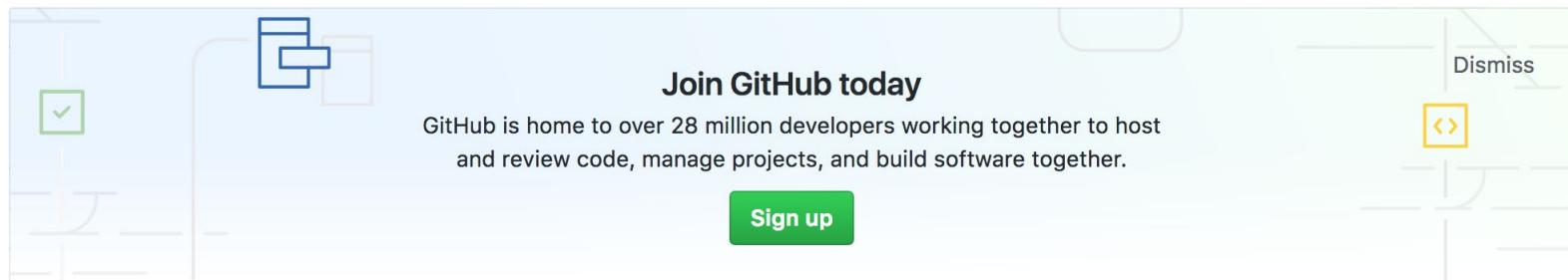
# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

juliocesarfort / public-pentesting-reports

Watch 285 Star 2,272 Fork 676

Code Issues 4 Pull requests 2 Projects 0 Insights



Curated list of public penetration test reports released by several consulting firms and academic security groups

61 commits 1 branch 0 releases 15 contributors

Branch: master ▾ New pull request Find file Clone or download ▾

juliocesarfort Merge pull request #25 from glerchundi/add-coredns-report ... Latest commit 0c995df on 14 Sep 2018

Bugcrowd Add reports from Instructure's public security reports: https://www.c... 3 years ago

Coinspect Adding Zcash reports 2 years ago

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

➤ **CoinspectReportZcash2016.pdf**

Branch: master ▾ [public-pentesting-reports](#) / [Coinspect](#) /

Create new file Find file History

 juliocesarfort Adding Zcash reports	Latest commit 0acc3cf on 9 Nov 2016	
..		
 <a href="#">CoinspectReportZcash2016.pdf</a>	Adding Zcash reports	2 years ago

A screenshot of a GitHub repository page for the 'public-pentesting-reports' organization. The 'Coinspect' folder is selected. A red box highlights the commit for 'CoinspectReportZcash2016.pdf'. A red arrow points to the timestamp '2 years ago'.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1



# Penetration Testing Report

Lista Pubblica di Pentesting Report – Esempio 1

---

## 1. Table of Contents

- [1. Table of Contents](#)
- [2. Executive Summary](#)
- [3. Introduction](#)
  - [3.1. Scope](#)
- [4. Summary Of Findings](#)
- [5. Findings](#)
- [6. Opportunity to fix Bitcoin's known problems](#)

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

---

## 2. Executive Summary

Between August and October 2016, [Zcash](#) engaged [Coinspect](#) to perform a security audit of their implementation of the [Zerocash](#) protocol. The objective of the audit requested by Zcash was to evaluate the security of Zcash's innovations over the [Bitcoin Core](#) source code.

During the assessment, Coinspect identified **2** high-risk issues, **3** medium-risk issues, and **6** low-risk issues. The high-risk issues identified during the assessment are not remotely exploitable by themselves to steal funds or compromise the privacy Zcash users. The high-risk and moderate-risk issues identified affect the performance and availability of the p2p network.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

---

### 3. Introduction

Zcash is an implementation of the Zerocash protocol based on the Bitcoin Core C++ code. It intends to offer a far higher standard of privacy and anonymity through a sophisticated zero-knowledge proving scheme which preserves confidentiality of transaction metadata.

A whitebox security audit was conducted on the Zcash source code in order to detect security, privacy, and availability related problems. Coinspect reviewed Zcash changes to Bitcoin Core, including their interaction with other parts of the Bitcoin protocol and other parts of the implementation.

The present report was completed on October 1st by Coinspect and includes results from the first and second phase of the audit.

### Engagement Highlights – 1/3

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

### 3.1. Scope

The Zcash [auditing strategy](#) tasked experts with different specializations to focus on different aspects of the system.

The objective of the first phase of the audit requested Coinspect to review changes to the Bitcoin Core code, focusing on the “**core consensus**” pieces. The review included but was not limited to the following checks:

- JoinSplit operations
- Transaction validation
- Founder's Reward
- Block header changes
- Transaction signing
- Input validation
- Denial of service prevention
- Integer overflows
- New data structures
- Cryptographic weaknesses

### Engagement Highlights – 2/3

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

The objectives of the second phase of the audit included:

- New RPC interface
- Wallet encryption
- Founder's Reward address rotation
- Information disclosure
- Changes made to the consensus code after the first phase concluded.

The audit conducted by Coinspect did not include: the zkSNARK cryptographic scheme, the libsnark implementation, or Equihash design.

### Engagement Highlights – 3/3

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

---

### 4. Summary Of Findings

ID	Description	Risk
ZCA-001	DoS attack if orphan JoinSplit transactions are enabled	Low
ZCA-002	Inheriting FindAndDelete from Bitcoin is considered dangerous	Medium
ZCA-003	scriptSig malleability allows 51% attack by invalidating honest miners blocks	High
ZCA-004	Decrease in huge-reorg security margin	Low
ZCA-005	Unlimited number of transaction proofs allows CPU-exhaustion attacks	Medium
ZCA-006	Erroneous nValueOut range check allows CPU-exhaustion attacks	High
ZCA-007	Forever growing nullifier set will end up being stored in nonvolatile memory	Low

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

### 4. Summary Of Findings

ID	Description	Risk
ZCA-001	DoS attack if orphan JoinSplit transactions are enabled	Low
ZCA-002	Inheriting FindAndDelete from Bitcoin is considered dangerous	Medium
ZCA-003	scriptSig malleability allows 51% attack by invalidating honest miners blocks	High
ZCA-004	Decrease in huge-reorg security margin	Low
ZCA-005	Unlimited number of transaction proofs allows CPU-exhaustion attacks	Medium
ZCA-006	Erroneous nValueOut range check allows CPU-exhaustion attacks	High
ZCA-007	Forever growing nullifier set will end up being stored in nonvolatile memory	Low

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

---

### 5. Findings (1/3)

ZCA-005	Unlimited number of transaction proofs allows CPU-exhaustion attacks
Category	Availability
Total Risk	<b>Medium</b>   Impact: Medium   Likelihood: Medium   Fix: Medium
Location	src/main.cpp:CheckTransaction()
Fix	Issue: <a href="#">#1388</a>

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

### 5. Findings (2/3)

#### Description

Zcash transactions can hold an unlimited number of JoinSplit proofs. The limit is indirectly enforced by the maximum block size, which limits the transaction size, which limits the number of JoinSplit elements that can be stored in the vjoinsplit vector. Assuming a JoinSplit consumes 1 Kbyte, a “heavy” transaction can hold 1000 vjoinsplit elements. If verifying a JoinSplit proof takes 10 msec, then verifying the heavy transaction would take 10 seconds. During this period the main lock of zcashd is held, so no other transaction can be processed. Therefore such transaction could be used to lock a node with a CPU-exhaustion attack. If the transaction is invalid because the last proof does not verify, then the transaction will not be broadcast and the attacker can use the same transaction to attack another node. If the transaction is valid, then the attacker can send a set of heavy transactions to the network at different entry points and force the network to be locked for long periods.

#### Findings Description

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 1

---

### 5. Findings (3/3)

#### Recommendations

Consider implementing one or more of the following suggestions:

- Move the verification of JoinSplit proofs to the last step of transaction verification.
- Make a transaction with more than 10 JoinSplit proofs non-standard.
- Increase the fees nodes and miners require for each JoinSplit element.
- Count each JoinSplit proof as a 10 sigops (block maximum is 20K sigops)

#### Findings Recommendations

# Penetration Testing Report

Lista Pubblica di Pentesting Report – Esempio 2

---

➤ <https://github.com/juliocesarfort/public-pentesting-reports/tree/master/RedSiege>



**R E D S I E G E**

Final Report

EXTERNAL NETWORK PENETRATION TEST

NAKATOMI TRADING CORP

JULY 15, 1988

Tipi e Metodologie di Testing

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

<b>EXECUTIVE SUMMARY.....</b>	<b>3</b>
<b>FINDINGS SUMMARY.....</b>	<b>4</b>
<b>FINDINGS CLASSIFICATIONS.....</b>	<b>5</b>
<b>FINDINGS .....</b>	<b>6</b>
<b>CRITICAL RISK FINDINGS .....</b>	<b>6</b>
<i>RS-NTC-001 Lack of Multi-Factor Authentication.....</i>	<i>6</i>
<i>RS-NTC-002 Password Reuse.....</i>	<i>7</i>
<b>HIGH RISK FINDINGS.....</b>	<b>9</b>
<i>RS-NTC-003 Default Credentials .....</i>	<i>9</i>
<b>MEDIUM RISK FINDINGS.....</b>	<b>10</b>
<i>RS-NTC-004 Website Missing HSTS Header.....</i>	<i>10</i>
<b>LOW RISK FINDINGS.....</b>	<b>11</b>
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present .....</i>	<i>11</i>
<b>METHODOLOGY.....</b>	<b>13</b>
<b>APPENDIX .....</b>	<b>14</b>
<b>FINDING CATEGORIES.....</b>	<b>14</b>
<b>TABLE OF FIGURES.....</b>	<b>15</b>

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

---

### **1. Executive Summary**

#### **SYNOPSIS**

Red Siege experts evaluated the security of Nakatomi Trading Corp's (NTC) perimeter network during the course of a three-week period in July 1988. The goal of the assessment was to identify security vulnerabilities in NTC's internet facing systems and services. All issues identified by Red Siege have been manually verified and exploited (where applicable) to demonstrate the underlying risk to NTC, its employees and clients.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### 1. Executive Summary

#### FINDINGS OVERVIEW

Findings grouped by risk severity:

- ❗ Critical Risk issues 2
- ⚠ High Risk issues 4
- ⚠ Medium Risk issues 6
- 💡 Low Risk issues 8
- 💡 Informational issues 2



# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

---

### 1. Executive Summary

### KEY FINDINGS

Red Siege found one critical vulnerability related to the authentication to NTC's VPN which allows an attacker to access internal systems. Additionally, Red Siege found two high severity vulnerabilities that have the potential to impact visitors to NTC's website which could impact Nakaoimi's brand and reputation.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

---

### 1. Executive Summary

#### STRATEGIC RECOMMENDATIONS

To increase the security posture of NTC, Red Siege recommends the follow strategic actions be taken:

- **IMPLEMENT TWO-FACTOR AUTHENTICATION ON PUBLIC FACING SYSTEMS.** Internet facing systems are regularly being probed and attacked. Extra care needs to be taken on these systems to prevent unauthorized access.
- **STRENGTHEN PASSWORD REQUIREMENTS.** NTC should use technical means to ban known bad/weak passwords and train users on safe password practices.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
FINDINGS CLASSIFICATIONS .....	5
FINDINGS .....	6
CRITICAL RISK FINDINGS .....	6
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
HIGH RISK FINDINGS .....	9
<i>RS-NTC-003 Default Credentials</i> .....	9
MEDIUM RISK FINDINGS .....	10
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
LOW RISK FINDINGS .....	11
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### 2. FINDINGS SUMMARY

RS-NTC-001 Lack of Multi-Factor Authentication

 Critical Risk      Authentication

RS-NTC-002 Password Reuse

 Critical Risk      Passwords

RS-NTC-003 Default Credentials

 High Risk      Configuration Management

RS-NTC-004 Website Missing HSTS Header

 Medium Risk      Configuration Management

RS-NTC-005 Web Server Content-Security-Policy Header Not Present

 Low Risk      Configuration Management

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
<b>FINDINGS CLASSIFICATIONS .....</b>	<b>5</b> 
FINDINGS .....	6
CRITICAL RISK FINDINGS .....	6
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
HIGH RISK FINDINGS .....	9
<i>RS-NTC-003 Default Credentials</i> .....	9
MEDIUM RISK FINDINGS .....	10
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
LOW RISK FINDINGS .....	11
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

---

### 3. Findings Classifications



#### *CRITICAL RISK ISSUES*

These vulnerabilities should be addressed promptly as they may pose an immediate danger to the security of the networks, systems, or data.

Exploitation does not require advanced tools or techniques or special knowledge of the target.



#### *HIGH RISK ISSUES*

These vulnerabilities should be addressed promptly as they may pose a significant danger to the security of the networks, systems, or data.

The issue is commonly more difficult to exploit but could allow for elevated permissions, loss of data, or a system downtime.



#### *MEDIUM RISK ISSUES*

These vulnerabilities should be addressed in a timely manner.

Exploitation is often difficult and requires social engineering, existing access, or special circumstances.



#### *LOW RISK ISSUES*

The vulnerabilities should be noted and addressed at a later date.

These issues offer very little opportunity or information to an attacker and may not pose an actual threat.



#### *INFORMATIONAL ISSUES*

These issues are for informational purposes only and likely do not represent an actual threat.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
FINDINGS CLASSIFICATIONS .....	5
<b>FINDINGS .....</b>	<b>6</b> ←
CRITICAL RISK FINDINGS .....	6
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
HIGH RISK FINDINGS .....	9
<i>RS-NTC-003 Default Credentials</i> .....	9
MEDIUM RISK FINDINGS .....	10
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
LOW RISK FINDINGS .....	11
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
FINDINGS CLASSIFICATIONS .....	5
FINDINGS .....	6
<b>CRITICAL RISK FINDINGS .....</b>	<b>6</b>
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
<b>HIGH RISK FINDINGS .....</b>	<b>9</b>
<i>RS-NTC-003 Default Credentials</i> .....	9
<b>MEDIUM RISK FINDINGS .....</b>	<b>10</b>
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
<b>LOW RISK FINDINGS .....</b>	<b>11</b>
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15



## 4. Findings (o Detailed Findings)

### CRITICAL RISK FINDINGS

#### RS-NTC-001 LACK OF MULTI-FACTOR AUTHENTICATION



##### Impact

The Web VPN server, allows authentication without use of a second factor. Red Siege was able to guess valid credentials through a *password spray* attack and subsequently access the internal network.

A screenshot of a web-based login interface. At the top, there's a header with the word "Login" and a user icon. Below the header are two input fields: "Email :" and "Password :". Underneath these fields is a reCAPTCHA verification box containing the text "I'm not a robot" next to a checkbox. To the right of the checkbox is the reCAPTCHA logo and the text "reCAPTCHA Privacy - Terms". Below the reCAPTCHA box are two buttons: a blue "Login" button with a person icon and a link "Forgot login?". At the bottom of the form, there's a message "No account yet? [Register and create a FREE account](#)".

FIGURE 1. VPN LOGIN FORM LACKING 2ND FACTOR

##### Affected System

10.1.2.3 – <https://vpn.ntc.nope/vpn>

##### Description

Use of multi-factor authentication, such as a hardware token or mobile application, prevents an attacker from simply guessing passwords, as the attacker needs the token to authentication. Even if the user's password is compromised, an attacker cannot login without the additional factor.

##### Recommendations

Implement multi-factor authentication on key external facing systems.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
FINDINGS CLASSIFICATIONS .....	5
FINDINGS .....	6
CRITICAL RISK FINDINGS .....	6
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
HIGH RISK FINDINGS .....	9
<i>RS-NTC-003 Default Credentials</i> .....	9
MEDIUM RISK FINDINGS .....	10
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
LOW RISK FINDINGS .....	11
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15



## 4. Findings (o Detailed Findings)

### HIGH RISK FINDINGS

RS-NTC-003    DEFAULT CREDENTIALS

 High Risk      Configuration Management

#### Impact

NTC uses default credentials on APC PDU systems.

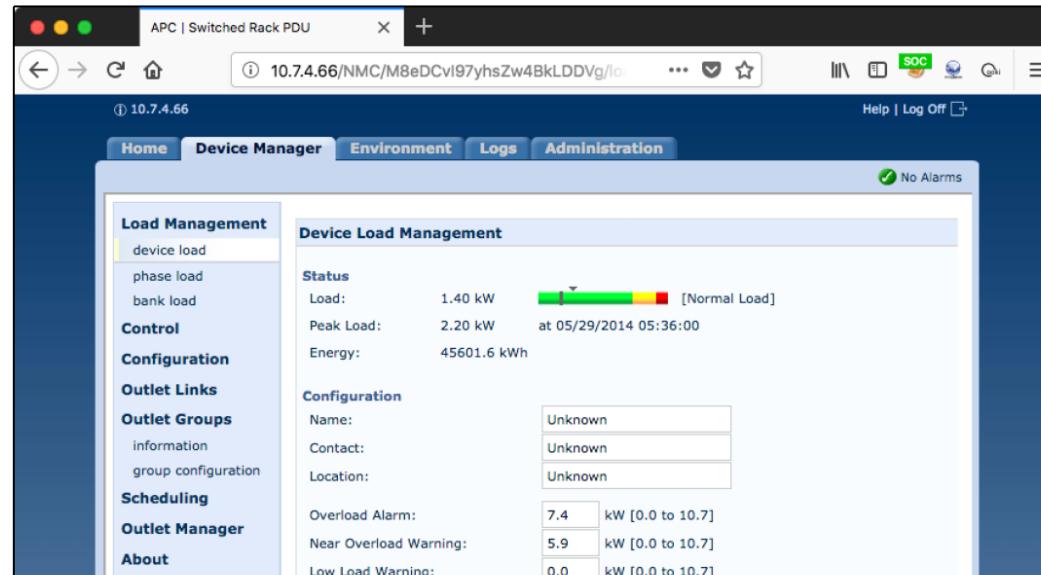


FIGURE 5. ACCESS TO APC PDU VIA DEFAULT CREDENTIALS

#### Affected System

APC PDU on port 80 with default credentials of `apc/apc`:

`10.1.2.50`

`10.7.4.66`

#### Discussion

Many devices and applications come with default usernames and passwords that provide privileged access to services. These credentials are published and easily discovered. Default credentials provide a common and easy entry point for an attacker.

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
FINDINGS CLASSIFICATIONS .....	5
FINDINGS .....	6
CRITICAL RISK FINDINGS .....	6
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
HIGH RISK FINDINGS .....	9
<i>RS-NTC-003 Default Credentials</i> .....	9
MEDIUM RISK FINDINGS .....	10
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
LOW RISK FINDINGS .....	11
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15



## 4. Findings (o Detailed Findings)

### MEDIUM RISK FINDINGS

RS-NTC-004

WEBSITE MISSING HSTS HEADER



Medium Risk

Configuration Management

#### *Impact*

The NTC corporate website does not enforce the HTTP Strict-Transport-Security header.

#### *Affected System*

10.1.2.3 – <https://www.ntc.nope>

#### *Description*

The HTTP Strict-Transport-Security (HSTS) header prevents browsers from communicating with websites using unencrypted HTTP channels. Without this header, an attacker able to position themselves between a user's web browser and the server can perform HTTPS downgrade attacks. Attackers frequently accomplish this using malicious WiFi hotspots in areas such as coffee shops frequented by employees of the targeted organization.

#### *Recommendations*

NTC should configure web servers to enforce the HSTS on all web servers supporting HSTS. Red Siege recommends the following configuration:

Strict-Transport-Security: max-age=63072000; includeSubdomains;

#### *References*

Strict-Transport-Security Header:

# Penetration Testing Report

## Lista Pubblica di Pentesting Report – Esempio 2

### TABLE OF CONTENTS

EXECUTIVE SUMMARY .....	3
FINDINGS SUMMARY .....	4
FINDINGS CLASSIFICATIONS .....	5
FINDINGS .....	6
CRITICAL RISK FINDINGS .....	6
<i>RS-NTC-001 Lack of Multi-Factor Authentication</i> .....	6
<i>RS-NTC-002 Password Reuse</i> .....	7
HIGH RISK FINDINGS .....	9
<i>RS-NTC-003 Default Credentials</i> .....	9
MEDIUM RISK FINDINGS .....	10
<i>RS-NTC-004 Website Missing HSTS Header</i> .....	10
LOW RISK FINDINGS .....	11
<i>RS-NTC-005 Web Server Content-Security-Policy Header Not Present</i> .....	11
METHODOLOGY .....	13
APPENDIX .....	14
FINDING CATEGORIES .....	14
TABLE OF FIGURES .....	15

## 4. Findings (o Detailed Findings)

### LOW RISK FINDINGS

#### RS-NTC-005 WEB SERVER CONTENT-SECURITY-POLICY HEADER NOT PRESENT

 Low Risk

*Configuration Management*

##### *Impact*

Without a properly set Content-Security-Policy header, an attacker can redress the web page and might be able to trick a user into taking actions on the website they did not intend to take.

##### *Affected Systems*

10.1.2.3 - <https://www.ntc.nope>

##### *Description*

The website does not implement the Content-Security-Policy header. When properly set, this header prevents the browser from loading the site inside another web page, an attack known as framing. Without this header, an attacker can carefully craft transparent and opaque layers to trick a user into clicking on buttons or links in the victim website and luring the user into taking actions on the targeted website they did not intend to take. The output below shows all the server headers for the affected website.

##### *Recommendations*

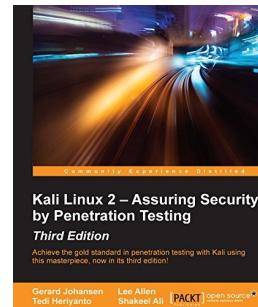
Configure the web server to set the Content-Security-Policy header for all server responses. Red Siege recommends the following configuration:

`Content-Security-Policy: frame-ancestors 'self';`

# Bibliografia

---

- **Kali Linux 2 - Assuring Security by Penetration Testing. Third Edition.** Gerard Johansen, Lee Allen, Tedi Heriyanto, Shakeel Ali. Packt Publishing. 2016
  - Capitolo 2



- **Ethical Hacking and Penetration Testing Guide.** Rafay Baloch. CRC Press. 2014
  - Capitolo 1

