



TP

Test Plan

**NetGun**

<b>Versione</b>	1.0
<b>Data</b>	06/02/2023
<b>Destinatario</b>	Professore Carmine Gravino
<b>Presentato da</b>	Carlo Colizzi, Giulio Incoronato, Antonio Mazzearella

## Team Members

Nome	Informazioni di contatto
Carlo Colizzi	c.colizzi@studenti.unisa.it
Giulio Incoronato	g.incoronato2@studenti.unisa.it
Antonio Mazzearella	a.mazzearella5@studenti.unisa.it

## Revision History

Data	Versione	Descrizione	Autori
12/12/2022	0.1	Stesura della sezione revision history, team members e del Sommario	Tutto il gruppo
12/12/2022	0.2	Stesura dell'introduzione e descrizione della relazione con gli altri documenti	Antonio Mazzearella
13/12/2022	0.3	Descrizione della panoramica del sistema	Carlo Colizzi. Antonio Mazzearella
14/12/2022	0.4	Stesura della funzionalità da testare e non	Tutto il gruppo
15/12/2022	0.5	Stesura Pass/Fail Criteria	Giulio Incoronato
15/12/2022	0.6	Definizione dell'approccio	Tutto il gruppo
16/12/2022	0.7	Stesura della Suspension and Resumption	Tutto il gruppo
16/12/2022	0.8	Descrizione degli strumenti Hardware/Software	Carlo Colizzi
17/12/2022	0.9	Stesura dei Test Case	Tutto il gruppo
06/02/2023	1.0	Correzione e revisione	Tutto il gruppo

# Sommario

Team Members	2
Revision History	3
1 Introduzione	5
2 Relazione con gli altri documenti	5
3 Panoramica del sistema	6
4 Funzionalità da testare e Non	6
5 Pass/Fail Criteria	7
6 Approccio	7
6.1 Testing di sistema	7
7 Suspension and Resumption	8
8 Strumenti (Hardware/Software)	8
9 Test Cases	9
9.1 Filtering	9
9.2 Deep Scanning	10
9.3 Create Report	11
10 Scheduling delle attività di testing	12

# 1 Introduzione

---

Questo documento è redatto con lo scopo di pianificare l'attività di testing del sistema NETGUN. Il testing è fondamentale per verificare il corretto funzionamento del sistema così da evitare errori durante l'utilizzo da parte dell'utente finale.

Il documento presenta le funzionalità da testare e non, le strategie e i vari casi di test.

## 2 Relazione con gli altri documenti

---

- [Requirements Analysis Document \(RAD\)](#)
- [System Design Document \(SDD\)](#)
- [Object Design Document \(ODD\)](#)
- [Test Plan \(TP\)](#)
- [Test Case Specification \(TCS\)](#)
- [Codice Sorgente](#)
- [Matrice di tracciabilità](#)
- Il Documento segue le metodologie presentate nel libro: Object-Oriented Software Engineering, di Bernd Bruegge & Allen H. Dutoit

## 3 Panoramica del sistema

---

Il sistema procede secondo un'architettura Three-tier dove il Data tier si occupa della gestione Data storage, il Presentation tier si occupa di mostrare graficamente gli output, il Logic tier gestisce l'interazione tra gli altri tier.

Il Logic tier si può dunque suddividere in:

- Scan controller
- Research CVE Controller
- Service Misconfiguration Controller
- Report Controller
- Test Network Performance Controller
- Tips Controller

## 4 Funzionalità da testare e Non

---

Verranno testate tutte le funzionalità del sistema, eccetto:

- Funzionalità di visualizzazione, poiché non presentano nessun input
- Funzionalità con priorità MEDIA o BASSA

Per quanto riguarda *Testing Network Performance*, non presenta input diretto dell'utente, verrà testata ugualmente ma non verrà ampliata nel documento in seguito.

### Testing Network Performance:

- Testing delle Performance sulla rete dell'Utente

Nel presente documento saranno Testate le funzionalità del Sistema che permettono un Interazione diretta dell'utente, queste sono: *Deep Scan*, *Filtering*, *Report*.

### Filtering

- Inserimento IP
- Inserimento Port

### Deep Scan

- Selezionamento Scan Mode

**Report:**

- Inserimento path per esportazione del File di Report

Successivamente, nel paragrafo 10, verranno utilizzati i seguenti acronimi per descrivere le componenti testate:

- IP: inserimento IP
- PO: inserimento port
- SC: selezionamento scan\_mode
- RT: inserimento path per l'esportazione del file report

## 5 Pass/Fail Criteria

---

Il criterio per il quale il testing avrà successo sarà quando verrà individuata una failure. In tal caso si procederà poi all'individuazione e correzione dell'errore.

## 6 Approccio

---

L'approccio al Testing del progetto NetGun si concentra su:

### 6.1 Testing di sistema

Questo tipo di Testing si focalizza sui requisiti funzionali e non. Nel caso del progetto NetGun, questo prevede il controllo di tre funzionalità chiave del sistema, attraverso il category partition.

## 7 Suspension and Resumption

---

La fase di testing sarà sospesa quando si otterranno i risultati attesi, rispettando però i tempi di consegna del progetto, data anche la quantità di tempo investito nell'attività.

La fase di Testing sarà comunque ripresa ad ogni modifica apportata al sistema. Infatti, in seguito a modifiche, si eseguiranno nuovamente tutti i casi di test per verificare la funzionalità del sistema, anche in caso di modifiche minime.

## 8 Strumenti (Hardware/Software)

---

Il materiale necessario per il testing sarà:

- Un elaboratore con Sistema Operativo Linux Debian-Based, avente NetGun installato
- Un Server (Target) virtualizzato e raggiungibile attraverso una Rete Locale all'elaboratore



## 9 Test Cases

Per la definizione dei test frame sarà utilizzato il category partition. Al fine di minimizzare il numero di test case, gli input saranno partizionati in classi di equivalenza.

Di seguito sono riportati i casi di test per le funzionalità da testare:

### 9.1 Filtering

<b>Parametro:</b> IP <b>Formato:</b> Sono presenti 4 Interi separati da 3 punti ( " . " ) AND Ogni intero X è : $X \geq 0$ AND $X \leq 255$	
Categorie	Scelte
Correttezza [CI]	1: rispetta il formato <b>[PROPERTY_CI_OK]</b> 2: non rispetta il formato <b>[errore]</b>
Lunghezza [LI]	1: lunghezza < 7 or lunghezza > 15 <b>[errore]</b> 2: lunghezza $\geq 7$ AND lunghezza $\leq 15$ <b>[PROPERTY_LI_OK]</b>

<b>Parametro:</b> Range Porte <b>Formato:</b> Sono presenti 2 interi separati da 1 trattino ( " - " ) AND Ogni intero Y è : $Y \geq 1$ AND $Y \leq 65535$ AND il primo intero inserito deve essere $\leq$ del secondo	
Categorie	Scelte
Correttezza [CP]	1: rispetta il formato <b>[PROPERTY_CP_OK]</b> 2: non rispetta il formato <b>[errore]</b>

Codice	Combinazione	Esito
TC_1.1_01	CI1.LI1.CP1	Errore
TC_1.1_02	CI1.LI1.CP2	Errore
TC_1.1_03	CI1.LI2.CP1	Input corretto, inserimento
TC_1.1_04	CI1.LI2.CP2	Errore
TC_1.1_05	CI2.LI1.CP1	Errore
TC_1.1_06	CI2.LI1.CP2	Errore
TC_1.1_07	CI2.LI2.CP1	Errore
TC_1.1_08	CI2.LI2.CP2	Errore

## 9.2 Deep Scanning

Parametro: scan mode	
Categorie	Scelte
Scelta [SS]	<b>1:</b> non viene selezionato niente <b>[errore]</b> <b>2:</b> viene selezionata una modalità <b>[PROPERTY_SS_OK]</b>

Codice	Combinazione	Esito
TC_1.2_01	SS1	Errore
TC_1.2_02	SS2	Inserimento

## 9.3 Create Report

<b>Parametro:</b> path + filename <b>Formato:</b> La Path inserita deve iniziare con “ / ”, Non devono essere presenti caratteri speciali o spazi	
Categorie	Scelte
Correttezza [CR]	<b>1:</b> la path inserita rispetta il formato <b>[PROPERTY_CR_OK]</b> <b>2:</b> la path inserita non rispetta il formato <b>[errore]</b>
Lunghezza [LR]	<b>1:</b> lunghezza del nome del file > 255 caratteri <b>[errore]</b> <b>2:</b> lunghezza del nome del file > 0 AND lunghezza del nome del file <= 255 <b>[PROPERTY_LR_OK]</b>

Codice	Combinazione	Esito
TC_1.3_01	CR1.LR1	Errore
TC_1.3_02	CR1.LR2	Input corretto, Inserimento
TC_1.3_03	CR2.LR1	Errore
TC_1.3_04	CR2.LR2	Errore

## 10 Scheduling delle attività di testing

---

