**ISTRUZIONI D’USO PER IL VERIFICATORE:**

Al seguito, si riportano le istruzioni d’uso base per l’uso proprio del verificatore dedito al controllo dei requisiti definiti sugli Assignment 1 e 2, del corso di “Cybersecurity” a cura del professor.Spognardi del dipartimento d’informatica dell’università Sapienza di Roma.

**PREPARAZIONE PRELIMINARE DELL’AMBIENTE DI TESTING:**

**Per iniziare, è necessario preparare l'infrastruttura in modo che sia compatibile con le funzioni e le istruzioni richieste dal verificatore***(o più precisamente****, i controlli sono eseguiti dai verificatori dedicati, poiché ciascun Assignment è dotato di un proprio controllore specifico)****.* Tale preparazione è volta a garantire che l'infrastruttura possa supportare operazioni come l'accesso SSH agli host virtualizzati e il recupero locale del file di backup della configurazione .XML.

**La prima cosa da fare in assoluto è l’installazione di diverse librerie**, adoperate all’interno del nostro verificatore, queste sono le seguenti: "requests", "colorama", "paramiko", "tqdm", "psutil", "pyotp"; e possono essere installate manualmente oppure lanciando lo script automatizzato “**install\_libraries.py**”.

**Successivamente si passa alla configurazione delle ACME, dove la scelta è ambi-valente**, a cura del **professore** si dovrebbe difatti richiedere la configurazione dell’infrastruttura in modo tale da:

1. Configurare una coppia di chiavi per l’accesso SSH, dal nome “SSHKey” per la versione privata e “SSHKey.pub” per la versione pubblica.

🡪Tale chiave deve essere inclusa nelle apposite cartelle “Verificator\_AX” (ognuna adibita al controllo e alla verifica di un preciso Assignment) e incluse nella cartella dello script “launcher” (script\_lanciatore.py).

1. Avere un utente di testing, che chiameremo per l’occasione “tester”, configurato su ogni Host dell’infrastruttura.
   1. Per ogni “tester” configurare la SSHKey all’interno del suo corrispettivo file “authorized\_keys”.
   2. Per ogni “tester” copiare la SSHKey all’interno della sua home directory (magari mediante *scp > tester@IP*).

**In caso contrario si può lasciar configurare l’infrastruttura allo studente mediante degli script** appositamente inseriti per l’occasione all’interno della cartella “Script”.

Purtuttavia dovranno essere anche rispettati, in questo caso, **dei vincoli preliminari**, in particolare:

* Bisogna assicurarsi che la voce del “**PermitRootLogin”** sia abilitato a “**yes**” (anche se temporaneamente) sul **"/etc/ssh/sshd\_config**" degli host o sulla voce “Administration” dei FW, su cui si vuole distribuire la chiave SSH che si andrà a generare, **e che un utente VPN di tipo “admin” abbia pieno accesso agli Host virtuali** e all’infrastruttura.
* **Inoltre la connessione OpenVPN deve essere dapprima aperta verso l'infrastruttura disegnata all'uso**, quindi bisogna lanciare il file di configurazione addetto alla ACME che vogliamo configurare.

Ora partendo in ordine:

1. **SSH KeyGeneration Script:**

Come il titolo, ampiamente suggerisce, questo script genera la coppia di chiavi citate al punto precedente all’interno della propria directory di lavoro.

Il lancio di questo script può avvenire mediante CLI, seguendo il seguente formato:

**“python Generator\_SSH.py <option>”**

Dove in “**option**” possiamo inserire:

-Generate: genera semplicemente le chiavi all’interno della cartella.

-Pubblish: pubblica le chiavi SSH presenti nella cartella SU OGNI HOST virtuale specificato all’interno dell’apposito “device.json”, il quale contiene la lista degli host virtuali con le loro apposite credenziali per fare un preliminare accesso SSH mediante password che verrà poi accantonato a favore dell’accesso con la chiave che vi verrà copiata sopra (ed ecco perché almeno all’inizio si prevede che il PermitRootLogin sia abilitato per ogni Host).

-All: compie ambe le operazioni citate pocanzi.

*Ma può avvenire anche effettuando un semplice lancio senza specificare l’option da prompt.*

Questo script ci consentirà l’accesso tramite la chiave SSH senza più la necessità di password semplicemente digitando:

**🡪ssh -i SSHKey *user*/*root*@IP**

*🡪Dove ‘user’ e ‘root’ possono essere i rispettivi username, degli utenti sugli host virtualizzati. Mentre ‘IP’ è banalmente l’indirizzo di riferimento all’host scelto.*

1. **UserGenerate.Script:**

Questo script si occupa invece della generazione dell’enviroment degli utenti tester all’interno della nostra infrastruttura.

In particolare dopo la connessione SSH (possibile mediante chiave configurata su ogni Host grazie allo script precedente) e grazie all’uso anche qui di un “device.json” che ci dirà a queli host connetterci, avremo:

1. Creazione di un utente "**tester**" per ciascun Host.
2. Configurazione della chiave pubblica in "/home/tester/.ssh/authorized\_keys", consentendo l'accesso tramite "**ssh -i SSHKey** [**tester@IP".**..e](mailto:tester@IP%22...e) non più solo a “**ssh -i SSHKey *root*/*user*@IP**” fatta fin’ora.
3. Distribuzione della stessa chiave pubblica anche nella directory "root" di ciascun host, precisamente in ".ssh/authorized\_keys", permettendo l'accesso tramite "**ssh -i SSHKey** [**root@IP"**...e](mailto:root@IP%22...e) distribuzione della stessa chiave pubblica anche nella directory "tester" di ciascun host

Compiuti questi passaggi, **siamo quasi pronti all’uso vero e proprio del verificatore**.

Rimane solo un'ultima operazione da compiere, ossia la creazione della cartella di test relativa all'ACME o alle ACME su cui si desidera verificare i requisiti degli Assignment 1 e 2.

**Nello specifico, si richiede di creare**, all'interno della stessa directory del "launcher" o anche detto "script\_lanciatore.py", **le cartelle appropriate il cui nome includa quello dell'ACME specifica da verificare**.

🡪*Ad esempio, "Testing ACME01", "Tester\_ACME01", e così via. È fondamentale che il nome della cartella rifletta in modo chiaro l'ACME in questione (si sottolinea che per i numeri inferiori alla doppia cifra, come 10, si preveda di anteporre uno zero: 01, 02, e così via).*

**E bisogna anche predisporre gli appositi file di configurazione, dall’estensione “.ovpn” al loro interno**.

🡪*In particolare servono i file: di un ClientVPN di tipo “admin”, e gli appositi file di configurazione per i controlli da effettuare sull’Assignment1 per quanto riguarda client VPN di tipo Operator ed Employee.*

**LANCIO DEL VERIFICATORE:**

Il verificatore apposito può quindi ora essere lanciato in modo automatizzato da un launcher, richiamandolo seguendo il formato del comando qui riportato:

**“python launcher.py <ACME>*/<ACME\_ALL>* <Assignment> *<TEST\_ALL>*”**

Dove:

**-ACME:** *corrisponde al nome della ACME su cui si vuole verificare un preciso assignment***.**

**🡪L'opzione, eventuale, “ACME\_ALL”***nel programma lancerà i test su tutte le cartelle denominate 'Testing ACME' presenti nella directory del launcher.*

**-Assignment:** *corrisponde al nome dell’Assignment che si vuole testare (Assignment1 o Assignment2, ma anche sottoderivazioni del genere come: A1 e A2, vengono riconosciute).*

**-TEST\_ALL:** *è un parametro, opzionale, che serve per far capire al verificatore se lanciare i controlli necessari al fine di valutare l’Assignment scelto sull’ACME in modo sequenziale e automatico o se invece lasciar modo all’utente di scegliere manualmente cosa e quando lanciare un certo controllo mediante un menù grafico realizzato tramite CLI e che ogni verificator singolo offre*.

All'avvio del comando, **il launcher procede alla duplicazione dei file di configurazione dell'ACME selezionata all'interno della directory designata come "Verificator"**.

In particolare, ciascuna cartella “Verificator” contiene le funzioni e i codici Python specifici per i controlli relativi a un determinato Assignment (per esempio, **"Verificator\_A1" per l'Assignment 1**).

**Dopo** aver copiato i file di configurazione**, il codice principale (Main) del verificatore specifico,** situato nella directory dell'Assignment oggetto di verifica**, viene eseguito dal programma di lancio (launcher). Durante l'esecuzione, vengono condotti tutti i controlli previsti e,** al termine**, viene generato un file di report contenente i risultati dell'operazione.**

Al termine del ciclo di esecuzione, **questo file di report viene replicato all'interno della cartella dell'ACME selezionata (o delle ACME, nel caso si sia deciso di testarle tutte)** per il test.

**NOTA IN CALCE:***Se si dovessero verificare errori di mancata risposta, da parte del Main-Firewall o altre componenti (come un “Road Warriors VPN” che non si connette a nessun host dopo l’apertura della sua connessione), potrebbe essere causato da qualche errore di rete o di mancata risposta da parte di questi ultimi.  
In tal caso si consiglia il rilancio del verificatore sul medesimo Assignment e ACME almeno una seconda volta, per constatare che non sia un errore di configurazione dell’infrastruttura da parte del gruppo che ha svolto l’esercitazione.*