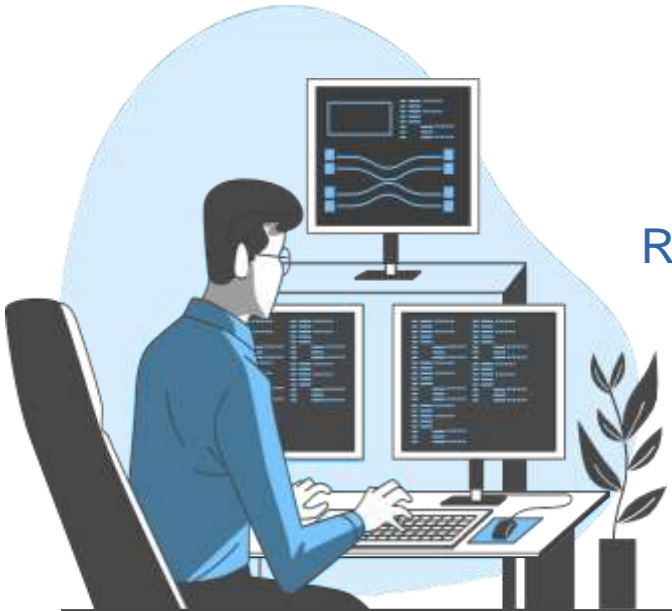


# SSH (Secure Shell)

Réalisé par :  
Zakaria El Omari  
Mohamed El Ouardi  
Taha Yassine Taabani

Encadré par :  
Mr. Moukhafi



# Tableau de contenu:

## 1 - Introduction au réseau informatique

- a. Définition réseau informatique et ces objectifs
- b. Les modèles TCP-IP/OSI

## 2 - SSH/SSL

- a. Définition SSH et son but.
- b. L'histoire du SSH
- c. Positionnement du SSH dans TCP-IP/OSI
- d. SSL en résumé
- e. SSH keys pour sécuriser bien la connexion

## 3 – ANSIBLE

- a. C'est quoi ANSIBLE L'utilisation du SSH dans l'ANSIBLE
- b. YAML

## 4 - Configuration du SSH et ANSIBLE

- a. Installation
- b. Connexion des machine avec SSH
- c. Génération du SSH keys
- d. Utilisation de ANSIBLE pour automatiser quelques tache en utilisant SSH

## 5 - Wireshark

- a. C'est quoi Wireshark
- b. Installation du Wireshark
- c. L'attaque du Wireshark

## 6 - Analyse des résultats (conclusion)



# 01

## Introduction au réseau informatique

- Définition réseau informatique et ces objectifs
- Les modèles TCP-IP/OSI

## Introduction au réseau informatique

Un réseau informatique est une structure qui permet l'interconnexion de systèmes et de dispositifs afin de faciliter le partage de ressources et d'informations entre eux.

L'objectif principal : est de permettre la communication efficace et la collaboration entre les différents éléments du réseau, favorisant ainsi le partage de données, de fichiers et de périphériques.

# TCP/IP vs OSI

Le modèle TCP/IP est un ensemble de protocoles largement utilisé pour les communications sur Internet. Il est divisé en quatre couches, souvent regroupées en deux catégories : la couche hôte et la couche réseau.

Le modèle OSI est une norme de référence internationale pour la conception et le fonctionnement des réseaux informatiques. Il est divisé en sept couches, chacune ayant des fonctions spécifiques.

Gère la *traduction*, la compression et le chiffrement des données.

Gère la fiabilité de la communication.

Gère l'accès au support physique, la détection d'erreurs, et le contrôle de flux.

#### Modèle OSI

7. Application

6. Présentation

5. Session

4. Transport

3. Réseau

2. Liaison de données

1. Physique

Fournit des interfaces pour les applications réseau.

Établit, maintient, et termine les sessions entre les applications.

Routage des données à travers le réseau.

Responsable de la transmission brute des bits sur un support physique.

- Correspondant aux couches session, présentation et application du modèle OSI.  
- Fournit des services réseau directement aux applications utilisateur.

- Correspond à la couche transport du modèle OSI.  
- Gère le transport des données de bout en bout.



- Correspond à la couche réseau du modèle OSI.  
- Responsable du routage des paquets à travers le réseau.

- Correspond aux couches physiques et de liaison de données du modèle OSI.  
- Gère l'accès au support physique et la transmission des données.

## 02 SSH/SSL



- Définition SSH et son but.
- L'histoire du SSH
- Positionnement du SSH dans TCP-IP/OSI
- SSL en résumé
- SSH keys pour sécuriser bien la connexion



## Définition de SSH et son but

- SSH, qui signifie "Secure Shell", est un protocole de communication sécurisé conçu pour permettre l'accès sécurisé à des systèmes distants sur un réseau non sécurisé.
- Il fournit un canal sécurisé sur une connexion non sécurisée, typiquement l'Internet, en utilisant des techniques de chiffrement pour protéger les données transitant entre le client et le serveur.



## L'histoire du SSH

1990

Début (années 1990) : Tatu Ylönen crée SSH comme alternative sécurisée pour l'accès à distance.

1990

Évolution (milieu à fin des années 1990) : SSH-2 est introduit, devenant un standard de facto pour la sécurité dans les communications.

2000

Défis (années 2000) : Découverte de vulnérabilités, nécessitant des mises à jour régulières pour maintenir la sécurité.

2010

Intégration moderne (années 2010 à aujourd'hui) : SSH devient essentiel dans la sécurité informatique, utilisé pour l'accès distant, le transfert de fichiers sécurisé et la gestion des identités.



## Positionnement du SSH dans TCP-IP/OSI



# TCP/IP

Dans le modèle TCP/IP, SSH s'insère également au niveau de la couche application, correspondant à la couche d'application du modèle OSI.

Il utilise des protocoles de transport sous-jacents tels que TCP pour assurer la fiabilité de la communication.

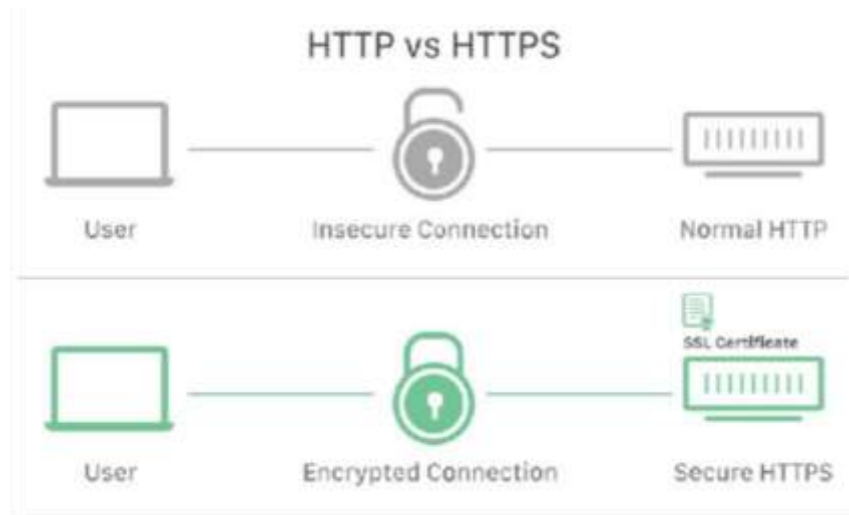
# OSI

SSH (Secure Shell) se positionne principalement au niveau de la couche application du modèle OSI.

Il opère au-dessus des couches de transport (par exemple, TCP) et de réseau (par exemple, IP), assurant ainsi une sécurisation des données à un niveau élevé dans la pile de protocoles.

## SSL en résumé

Le SSL(Secure Sockets Layer): est une technologie standard de sécurisation des connexions Internet par le chiffrement des données transitant entre un navigateur et un site web (ou entre deux serveurs). Durant leur transfert, les données (personnelles, financières, etc.) sont ainsi protégées des hackers qui ne peuvent ni les voir ni les détourner.





## SSH keys Pour sécuriser bien la connexion



Les clés SSH, ou clés de chiffrement Secure Shell, sont une méthode de sécurisation des connexions réseau, notamment utilisée pour l'accès à distance à des serveurs. Le principe de base est le chiffrement asymétrique, qui implique l'utilisation de deux clés distinctes : une clé privée et une clé publique.

### **Clé Privée**

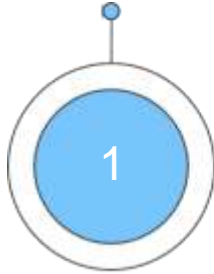
- C'est la clé secrète qui doit être gardée confidentielle.
- Elle est utilisée pour signer numériquement les données et déchiffrer les informations chiffrées par la clé publique correspondante.

### **Clé Publique**

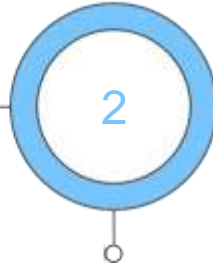
- C'est la clé qui peut être partagée publiquement sans compromettre la sécurité.
- Elle est utilisée pour vérifier la signature numérique créée par la clé privée et pour chiffrer les données de manière à ce que seule la clé privée puisse les déchiffrer.

# Le processus est le suivant :

Un utilisateur génère  
une paire de clés  
(publique et privée)  
sur son ordinateur.



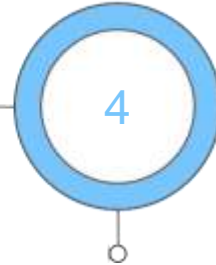
La clé publique est ensuite  
copiée sur les serveurs  
distants auxquels l'utilisateur  
souhaite se connecter de  
manière sécurisée.



Lorsqu'une connexion SSH est établie,  
le serveur utilise la clé publique pour  
chiffrer un message, et seule la clé  
privée correspondante, détenue par  
l'utilisateur, peut le déchiffrer.



De plus, la clé privée est  
utilisée pour signer  
numériquement les données  
envoyées au serveur, ce qui  
permet au serveur de vérifier  
l'identité de l'utilisateur.



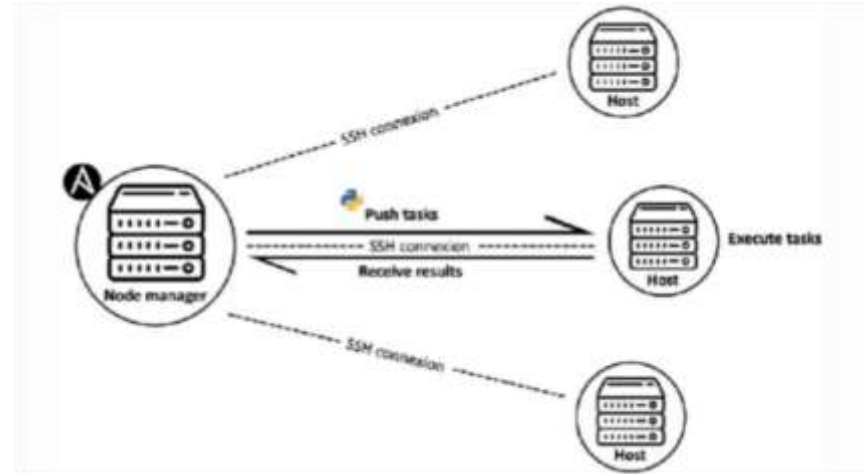


# 03 **A**NSIBLE

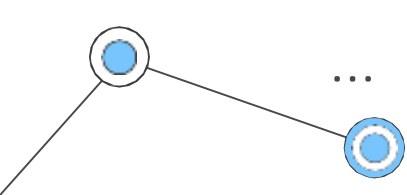
- C'est quoi l'ANSIBLE et l'utilisation du SSH dans ANSIBLE
- YAML

# C'est quoi ANSIBLE et l'utilisation du SSH dans l'ANSIBLE:

Ansible est une plateforme open source d'automatisation des configurations et du déploiement d'infrastructures informatiques. Elle simplifie les tâches répétitives liées à la gestion de configuration, au déploiement d'applications et à l'orchestration des infrastructures. Ansible est développé par Red Hat, écrit en Python, et repose sur un modèle déclaratif, où les utilisateurs décrivent simplement l'état souhaité du système, et Ansible se charge d'appliquer ces états sur les machines cibles.

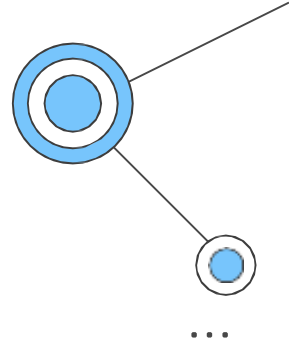






## YAML :

En Ansible, YAML (YAML Ain't Markup Language) est un langage de sérialisation de données utilisé pour décrire les configurations. Il est employé dans les fichiers de configuration Ansible tels que les playbooks, les fichiers de variables et les inventaires. La syntaxe YAML est basée sur l'indentation, favorisant la lisibilité, et permet de définir des tâches, des rôles et d'autres éléments nécessaires à l'automatisation des infrastructures de manière concise. Son utilisation simplifie la rédaction et la lecture des fichiers de configuration Ansible.



```
YAML
---
- name: Exemple de playbook Ansible
  hosts: serveurs_web
  become: true # Exécuter les tâches avec des privilèges élevés

  tasks:
    - name: Assurer l'installation de Nginx
      apt:
        name: nginx
        state: present

    - name: Démarrer le service Nginx
      service:
        name: nginx
        state: started
```

# 04 Configuration du SSH et ANSIBLE

- Installation
- Connexion des machine avec SSH
- Génération du SSH keys



# INSTALLATION

## SSH

```
- > sudo pacman -S openssh
[sudo] password for taha:
warning: openssh-9.5p1-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...
```

```
Packages (1) openssh-9.5p1-1
```

```
Total Installed Size: 4.90 MiB
```

```
Net Upgrade Size: 0.00 MiB
```

```
:: Proceed with installation? [Y/n]
```

```
(1/1) checking keys in keyring
```

```
(1/1) checking package integrity
```

```
(1/1) loading package files
```

```
(1/1) checking for file conflicts
```

```
:: Processing package changes...
```

```
(1/1) reinstalling openssh
```

```
:: Running post-transaction hooks...
```

```
(1/3) Reloading system manager configuration...
```

```
(2/3) Creating temporary files...
```

```
(3/3) Arming ConditionNeedsUpdate...
```

```
- >
```

## ANSIBLE

```
- > sudo pacman -S ansible
warning: ansible-8.6.1-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...
```

```
Packages (1) ansible-8.6.1-1
```

```
Total Installed Size: 676.35 MiB
```

```
Net Upgrade Size: 0.00 MiB
```

```
:: Proceed with installation? [Y/n]
```

```
(1/1) checking keys in keyring
```

```
(1/1) checking package integrity
```

```
(1/1) loading package files
```

```
(1/1) checking for file conflicts
```

```
:: Processing package changes...
```

```
(1/1) reinstalling ansible
```

```
:: Running post-transaction hooks...
```

```
(1/1) Arming ConditionNeedsUpdate...
```

```
- >
```

# Activation de firewall

## firewall

```
- > sudo pacman -S ufw
warning: ufw-0.36.2-2 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) ufw-0.36.2-2

Total Installed Size: 0.93 MiB
Net Upgrade Size:    0.00 MiB

:: Proceed with installation? [Y/n]
(1/1) checking keys in keyring
(1/1) checking package integrity
(1/1) loading package files
(1/1) checking for file conflicts
:: Processing package changes...
(1/1) reinstalling ufw
:: Running post-transaction hooks...
(1/2) Reloading system manager configuration...
(2/2) Arming ConditionNeedsUpdate...
- >
```

## Enable firewall & status

```
- > sudo ufw enable
Firewall is active and enabled on system startup
- >

- > sudo ufw allow 22/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
- > sudo ufw status
Status: active

To Action From
--
22 ALLOW Anywhere
22/tcp ALLOW Anywhere
2222/tcp ALLOW Anywhere
22 (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
2222/tcp (v6) ALLOW Anywhere (v6)
```

# Connexion des machine avec SSH

```
- > ssh aizen@192.168.1.5
aizen@192.168.1.5's password:

Last login: Fri Nov 17 23:30:15 2023 from 192.168.1.2
aizen@WEED
-----
OS: Linux Mint 21.2 x86_64
Host: R530/R730
Kernel: 6.2.0-36-generic
Uptime: 5 hours, 39 mins
Packages: 2200 (dpkg), 14 (flatpak)
Shell: bash 5.1.16
Resolution: 1366x768
Terminal: /dev/pts/1
CPU: Pentium T4400 (2) @ 2.200GHz
GPU: Intel Mobile 4 Series Chipset
Memory: 1300MiB / 5850MiB

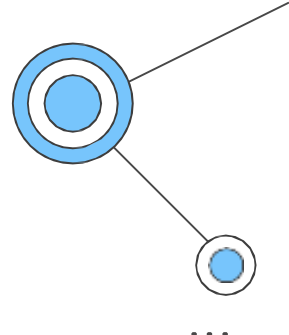
aizen@WEED:~$ ls
Desktop Documents Downloads mintupgrade-2023-11-17T230629.log Music Pictures Public Templates Videos
```

# Génération du SSH Keys

```
- > ssh-keygen -t ed25519 -C "key test"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/taha/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/taha/.ssh/id_ed25519
Your public key has been saved in /home/taha/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:DS67GIwHP55RA8FW+uTEdCEo5ZMqBwUCT3l0b/gZSg key test
The key's randomart image is:
+--[ED25519 256]--+
|+==0.=*+|
|.0+.0=00|
|000+0.+0.|
|..=+E00++0|
|.. +0*+S..|
|  = 0.+.|
| . B .|
| o * .|
| + .|
+----[SHA256]-----+
- > ls -la .ssh
total 24
drwx----- 2 taha taha 4096 Nov 18 20:06 .
drwx----- 47 taha taha 4096 Nov 18 19:53 ..
-rw----- 1 taha taha 444 Nov 18 20:06 id_ed25519
-rw-r--r-- 1 taha taha 90 Nov 18 20:06 id_ed25519.pub
-rw----- 1 taha taha 1662 Nov 18 00:07 known_hosts
-rw----- 1 taha taha 924 Nov 18 00:07 known_hosts.old
- > cat .ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDGaaoCyCgVoJAKPuNaFFkuesyksdJc36D08rmaClYwS key test
```



A diagram showing a graph structure. It consists of two nodes, each represented by a blue circle with a white border. The nodes are connected by a black line. To the right of the second node, there is an ellipsis (...) indicating a continuation of the graph.



```

➤ ssh-copy-id -i ~/.ssh/id_ed25519.pub 192.168.1.5
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/taha/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
taha@192.168.1.5's password:
➤ ssh-copy-id -i ~/.ssh/id_ed25519.pub aizen@192.168.1.5
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/taha/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
aizen@192.168.1.5's password:
...-:~::~-...
..MMMMMMMMMMMMMMMMMM-..
..MMMM'..-:~::~-...-..MMMM-..
.:MMMM.:MMMMMMMMMMMMMMMMMM.:MMMM.:
-MMM-R---MMMMMMMMMMMMMMMMMMMMMM-MMM-
':MMM:MM':MMMM:.....-:~::~-:MMMM:MM:'
:MMM:MM':MM:':~::~-:MMMM:MM:
:MMM:MM':MM:~MM:~MM:~MMMM:MM:
:MMM:MM':MM:~MM:~MM:~MMMM:MM:
:MMM:MM':MM:~MM:~MM:~MMMM:MM:
:MMM:MM':MM:~MM:~MM:~MMMM:MM:
:MMM:MM':MM:~MM:~MM:~MMMM:MM:
:MMM:MM':~MMMMMMMMMMMMMM-~MM:MM:
:MMM:MM':~MMMM:MM:
:MMM:MM:~MMMM:MM:
'-MMM.-MMMMMMMMMMMMMMMM-.'
'..MMM'~:~::~-~'~MMMM-.'
'..MMMMMMMMMMMMMM-'
'~:~::~-~'
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'aizen@192.168.1.5'"
and check to make sure that only the key(s) you wanted were added.

```

# Connexion avec SSH keys

```
- > ssh aizen@192.168.1.5
Enter passphrase for key '/home/taha/.ssh/id_ed25519':

Last login: Sat Nov 18 19:01:33 2023 from 192.168.1.2
aizen@WEED
-----
OS: Linux Mint 21.2 x86_64
Host: R530/R730
Kernel: 6.2.0-36-generic
Uptime: 5 hours, 53 mins
Packages: 2200 (dpkg), 14 (flatpak)
Shell: bash 5.1.16
Resolution: 1366x768
Terminal: /dev/pts/1
CPU: Pentium T4400 (2) @ 2.200GHz
GPU: Intel Mobile 4 Series Chipset
Memory: 1298MiB / 5850MiB

aizen@WEED:~$
```



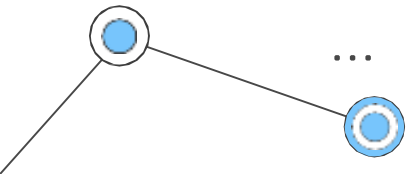
# Configuration de l'ANSIBLE

```
el-ouardi@med:~$ ls
Desktop  Documents  Downloads  id_ed25519.pub  Music  Pictures  Public  snap  Templates  Videos
el-ouardi@med:~$ ls /etc/ansible
ls: cannot access '/etc/ansible': No such file or directory
el-ouardi@med:~$ mkdir ansible
el-ouardi@med:~$ ls
ansible  Desktop  Documents  Downloads  id_ed25519.pub  Music  Pictures  Public  snap  Templates  Videos
el-ouardi@med:~$ cd ansible/
el-ouardi@med:~/ansible$ nvim ansible.cfg
el-ouardi@med:~/ansible$ nvim inventory
el-ouardi@med:~/ansible$ nvim ansible.cfg
```

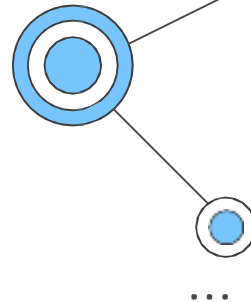
```
el-ouardi@med: ~/ansible

[machine]
zarck@192.168.54.164
```

```
[defaults]
inventory=inventory
host_key_checking=False
~
~
```



# Test ping avec ANSIBLE



```
el-ouardl@ned:~/ansible$ ansible all -m ping --ask-pass
SSH password:
[WARNING]: Platform linux on host zarch@192.168.34.164 is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
zarch@192.168.34.164 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
el-ouardl@ned:~/ansible$
```

# Exemple d'un playbook No 1

```
el-ouardi@med: ~/ansible
--
- name: Créer un dossier avec Ansible
  hosts: machine
  become: true # Utiliser sudo pour exécuter des commandes en tant que superutilisateur

  tasks:
    - name: Créer le dossier test_ansible
      file:
        path: /home/zarck/test_ansible # Spécifier le chemin du dossier à créer
        state: directory # Indiquer qu'il s'agit d'un dossier
        mode: 0755 # Définir les permissions du dossierii
```

# Execution du playbook No 1

```
el-guarnid@med:~/ansible$ ls
ansible.cfg  directory.yaml  inventory
el-guarnid@med:~/ansible$ ansible-playbook directory.yaml -K --ask-pass
SSH password:
BECOME password[defaults to SSH password]:
```

```
PLAY [Créer un dossier avec Ansible] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host zarck@192.168.9.164 is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
ok: [zarck@192.168.9.164]

TASK [Créer le dossier test_ansible] *****
changed: [zarck@192.168.9.164]

PLAY RECAP *****
zarck@192.168.9.164 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

el-guarnid@med:~/ansible$
```

```
> ls
Desktop  Documents  Downloads  eclipse-workspace  Music  Pictures  project-app  Public  Templates  test_ansible  Videos  yay
```

## Exemple du playbook No 2

```
el-ouardi@med: ~/ansible
--
- name: Télécharger un programme avec Ansible
  hosts: machine
  become: true # Utiliser sudo pour exécuter des commandes en tant que superutilisateur

  tasks:
    - name: Télécharger htop
      pacman:
        name: htop
        state: present # Assurez-vous que le paquet est présent sur le systèmei
```

# Execution du playbook No 2

```
el-guardi@med:~/ansible$ nvim inventory
el-guardi@med:~/ansible$ nvim install_HTOP.yaml
el-guardi@med:~/ansible$ ansible-playbook install_HTOP.yaml -K --ask-pass
SSH password:
BECOME password[defaults to SSH password]:
```

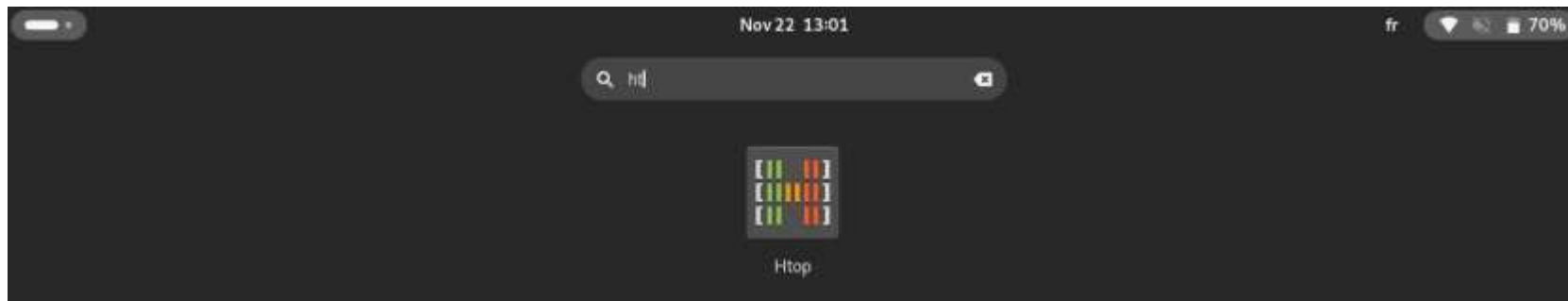
```
PLAY [Télécharger un programme avec Ansible] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host zarck@192.168.9.164 is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.14/reference_appendices/interpreter_discovery.html for more information.
ok: [zarck@192.168.9.164]

TASK [Télécharger htop] *****
changed: [zarck@192.168.9.164]

PLAY RECAP *****
zarck@192.168.9.164      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

el-guardi@med:~/ansible$
```



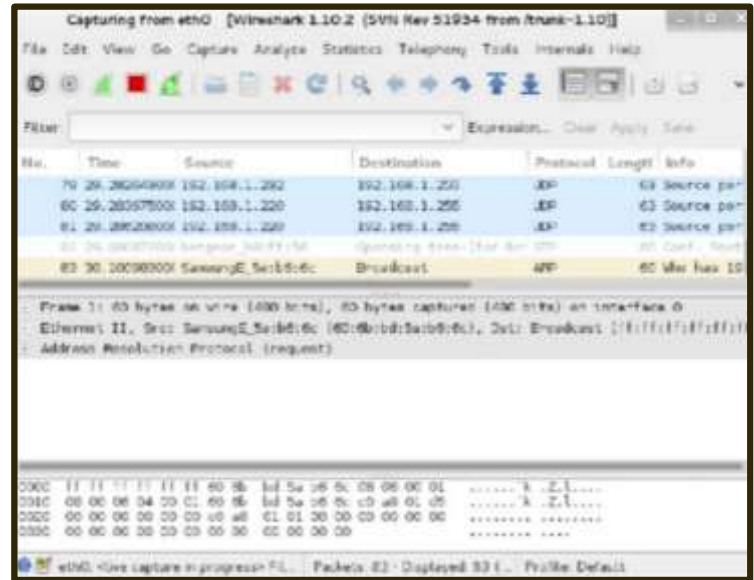


# 05 WIRESHARK

- C'est quoi WIRESHARK
- Installation du Wireshark
- L'attaque du Wireshark

# C'est quoi WIRESHARK

**Wireshark** est un outil open-source d'analyse de paquets réseau qui permet de capturer, visualiser, et inspecter le trafic sur un réseau. Il offre une compréhension approfondie du fonctionnement des communications réseau en examinant les paquets de données qui transitent entre les différents points du réseau.







# Installation du WIRESHARK



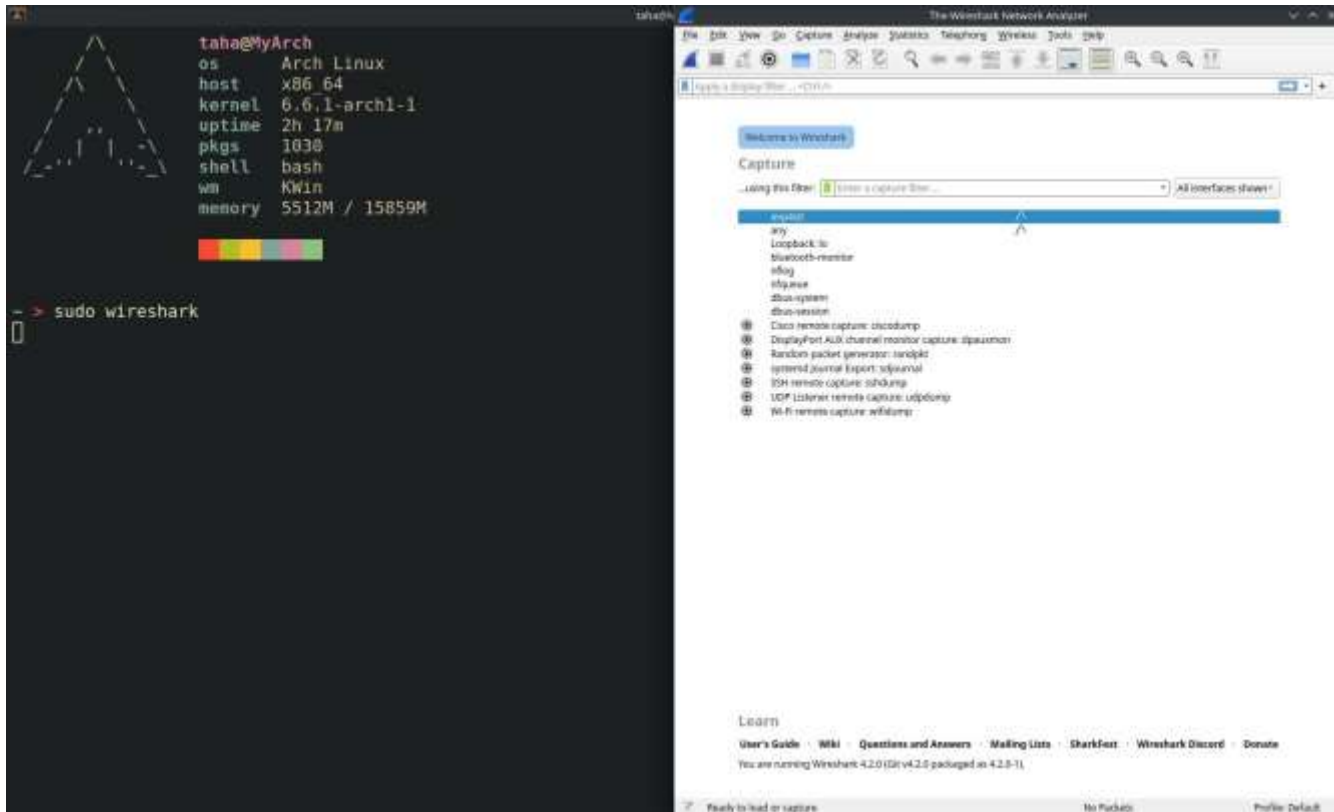
```
~ > sudo pacman -S wireshark-cli
warning: wireshark-cli-4.2.0-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...

Packages (1) wireshark-cli-4.2.0-1

Total Installed Size: 128.45 MiB
Net Upgrade Size:      0.00 MiB

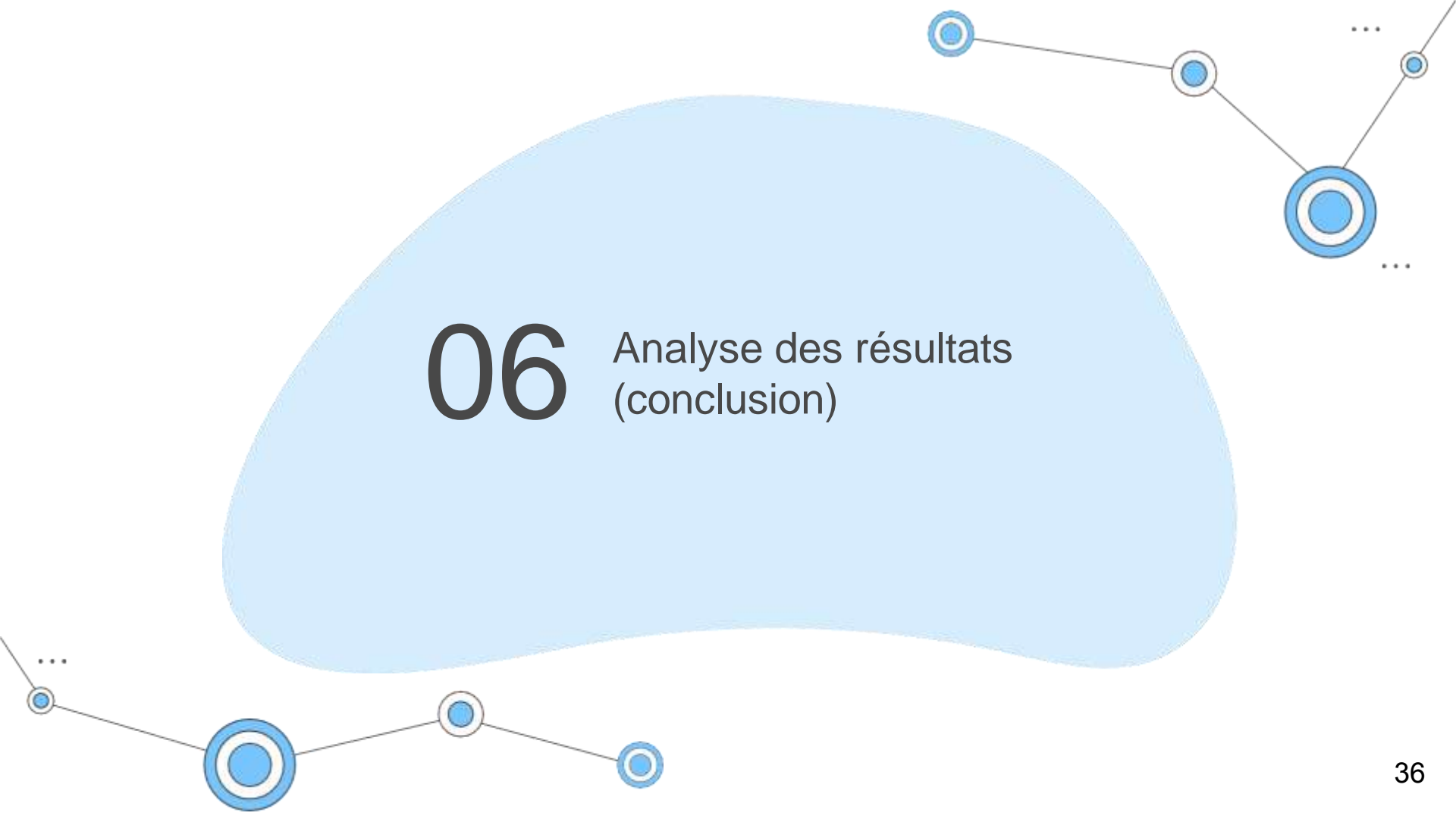
:: Proceed with installation? [Y/n]
(1/1) checking keys in keyring
(1/1) checking package integrity
(1/1) loading package files
(1/1) checking for file conflicts
:: Processing package changes...
(1/1) reinstalling wireshark-cli
:: Running post-transaction hooks...
(1/2) Creating system user accounts...
(2/2) Arming ConditionNeedsUpdate...
~ >
```

# Ouvrir WIRESHARK avec command sudo



# L'attaque avec WIRESHARK

The screenshot displays the Wireshark network protocol analyzer interface. The top bar shows the capture is on the 'eth0' interface. The left pane lists captured packets, with packet 1 selected. The middle pane shows the details of the selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The right pane shows the packet bytes in hexadecimal and ASCII. The top bar indicates the capture is on the 'eth0' interface.



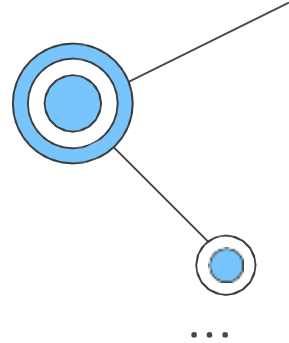
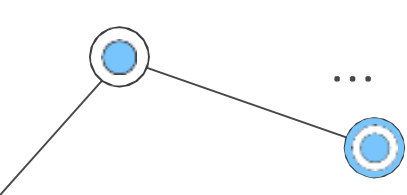
# 06 Analyse des résultats (conclusion)



## Analyse des résultats



- En examinant les résultats de la simulation avec SSH, on peut montrer que les données échangées entre le client et le serveur sont sécurisées. Le chiffrement des données par SSH garantit que même si un attaquant intercepte les paquets, il ne peut pas comprendre le contenu, car il est chiffré.
- La comparaison visuelle des données capturées avec Wireshark dans les deux scénarios soulignera la sécurité renforcée offerte par SSH. Les informations sensibles, telles que les identifiants de connexion, les commandes, ou les données confidentielles, restent confidentielles et ne peuvent pas être exploitées par un tiers non autorisé.
- On peut également mettre en avant le mécanisme d'authentification forte de SSH, qui ajoute une couche supplémentaire de sécurité en vérifiant l'identité des parties impliquées dans la communication.



**MERCI POUR  
VOTRE ATTENTION**

# Resources

[https://wiki.archlinux.org/title/Secure\\_Shell](https://wiki.archlinux.org/title/Secure_Shell)

<https://wiki.archlinux.org/title/Ansible>

<https://wiki.archlinux.org/title/Wireshark>

<https://www.geeksforgeeks.org/tcp-ip-model/>

<https://www.digicert.com/fr/what-is-ssl-tls-and-https>

<https://docs.ansible.com>