TÉLÉCOM PARIS

SR2I 204

Conception et mise en œuvre d'un site Web supportant la méthode d'accès TOTP

Réalisé PAR :
Mzouz Yousra
Ghalas Ihssane
EL Marchouk Kenza

Filière : SR2I (A2)

2021/2022

Table des matières

Chapitre 1

Introduction générale de la théorie

1 Authentification : définition, types et protocoles :

1.1 Définition de l'authentification et son but :

L'authentification est intégrée dans la plupart des applications informatiques, il s'agit de la procédure qui permet de déterminer si une entité est bien effectivement ce qu'elle est censée être. Elle permet ainsi de savoir si c'est bien légitime d'autoriser l'accès d'une entité à des ressources du système.
Il s'agit de comparer les informations des utilisateurs autorisés conservées dans la base de données aux informations fournies. L'accès sera autorisé seulement si les informations sont identiques. L'administrateur du système d'information octroie les droits d'accès et chaque utilisateur possède un compte d'accès (identifiant + mot de passe) aux ressources qu' il est autorisé à voir.

1.2 Les différentes familles de l'authentification :

On distingue trois familles d'authentifications : l'authentification simple, forte et unique :

- **L'authentification simple :** repose sur un seul facteur. L'exemple le plus courant est le mot de passe, mais il n'est plus sécurisé, plusieurs techniques d'attaques permettent de le retrouver facilement.

- **L'authentification forte :** repose sur au moins deux facteurs et elle est plus utilisée dans le secteur bancaire. Les exemples les plus courants de cette authentification sont : mot de passe à usage unique (OTP), Certificat Numérique ...

- **L'authentification unique :** Single Sign-On ou SSO, est une méthode permettant à un utilisateur de ne procéder qu'à une seule authentification pour accéder à plusieurs applications informatiques (ou sites internet sécurisés). Elle permet une meilleure gestion des mots de passe et des données personnelles.

1.3 Les quatre facteurs de l'authentification :

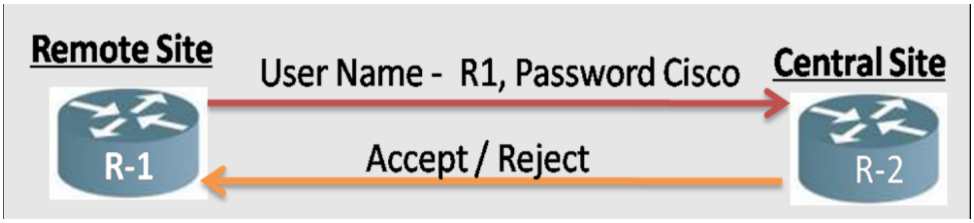
Les facteurs de l'authentification : Il s'agit d'un élément (qui peut être physique, cognitif, ...) permettant de produire une empreinte servant à l'utilisateur comme outil d'authentification. Cette empreinte doit être personnelle à l'utilisateur et doit pouvoir être numérisée.

Il existe 4 facteurs d'authentification utilisables lors de la phase d'autorisation d'accès à des ressources sécurisées :

- **Facteur mémoriel :** ce que l'utilisateur a mémorisé et il le connaît seul, l'exemple le plus courant est le mot de passe.
- **Facteur matériel :** information que seul l'utilisateur possède et enregistrée dans un objet qu'il utilise comme la clé USB, un certificat numérique…
- **Facteur corporl :** c'est une information qui caractérise l'utilisateur avec sa propre empreinte, comme la voix, l'empreinte digitale…
- **Facteur réactionnel :** il s'agit d'un geste que seul l'utilisateur peut produire, par exemple sa signature.

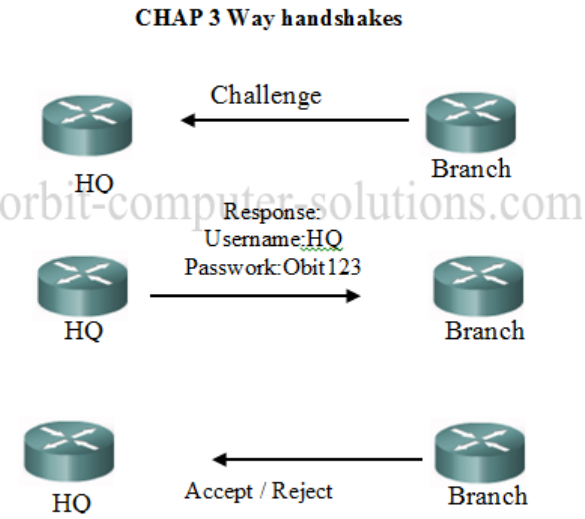
1.4 Quelques protocoles d'authentification autre que OTP :

PAP : Password Authentication Protocol :



Password Authentication Protocol (PAP) est un protocole d'authentification par mot de passe. Il s'agit d'envoyer l'identifiant et le mot de passe en clair à travers le réseau. Si le mot de passe correspond, alors l'accès est autorisé.
L'avantage du PAP est qu'il est extrêmement simple à implémenter, lui permettant d'être utilisé dans des systèmes embarqués très légers. Mais les données sont transmises en texte clair sur le réseau et le créateur peut faire des tentatives répétées d'accès, ce qui rend PAP par conséquent non sécurisé. Il n'est utilisé en pratique qu'à travers un réseau sécurisé.

CHAP : Challenge Handshake Authentication Protocol :



CHAP est un protocole plus sécurisé. Son objectif est que l'authentification auprès d'un authentificateur se fait sans échange de mot de passe en clair sur le réseau et sans que l'échange puisse être

rejoué par un tiers à l'écoute.
Le principe est le suivant :

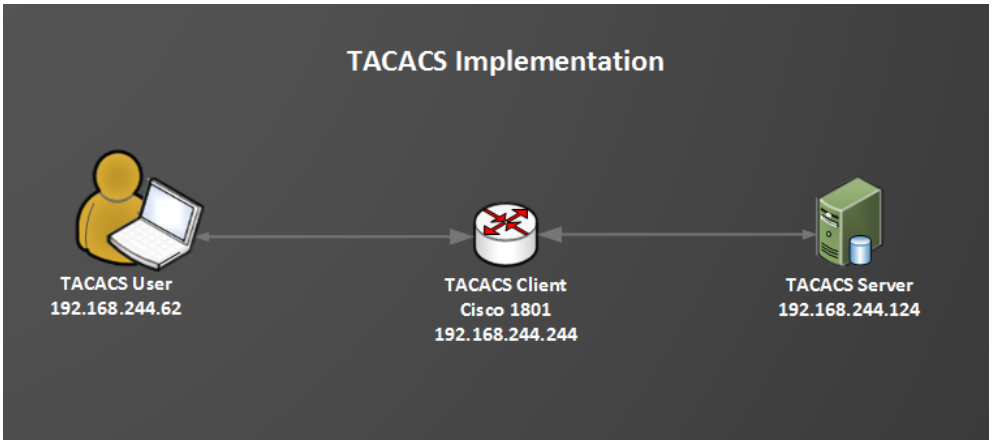
- Après que le lien soit fait, le serveur envoie un message au demandeur de connexion. Le demandeur répond avec une valeur obtenue en utilisant une fonction à sens unique d'informations parasites.
- Le serveur contrôle la réponse en la comparant à son propre calcul de la valeur prévue d'informations parasites.
- Si les valeurs sont égales, la connexion est terminée. À tout moment, le serveur peut inviter la partie reliée pour envoyer un nouveau message.

HTTP :

Cette méthode permet de s'identifier auprès d'un serveur HTTP en lui montrant que l'on connaît le nom d'un utilisateur et son mot de passe, afin d'accéder aux ressources à accès restreint de celui-ci.
Il existe deux méthodes de l'authentification HTTP :

- **La méthode Basic :** le client fait une requête http vers le serveur, le serveur lui retourne un message d'erreur (le code HTTP 401) et lui demande de s'authentifier. Le client refait la première requête avec son login et son mot de passe en clair et le serveur lui retourne la ressource demandée si les paramètres envoyés sont valides.
- **La méthode Digest :** Le client fait une requête http vers le serveur, ce dernier lui retourne un message d'erreur (401) et lui demande de s'authentifier avec le mode digest, le serveur lui renvoie un nonce (nombre aléatoire). Le client refait sa première requête avec son identifiant (login) et le résultat du haché de son mot de passe combiné avec le nonce et éventuellement d'autres paramètres. Le serveur lui retourne la ressource si les paramètres fournis ont été validés.
  Le mot de passe n'est pas envoyé en clair mais cette méthode impose de stocker ce mot de passe en clair. Même si le mode digest est plus sûr que le mode basic, mais il reste de même sensible aux attaques.

TACACS : Terminal Access Controller Access-Control System :

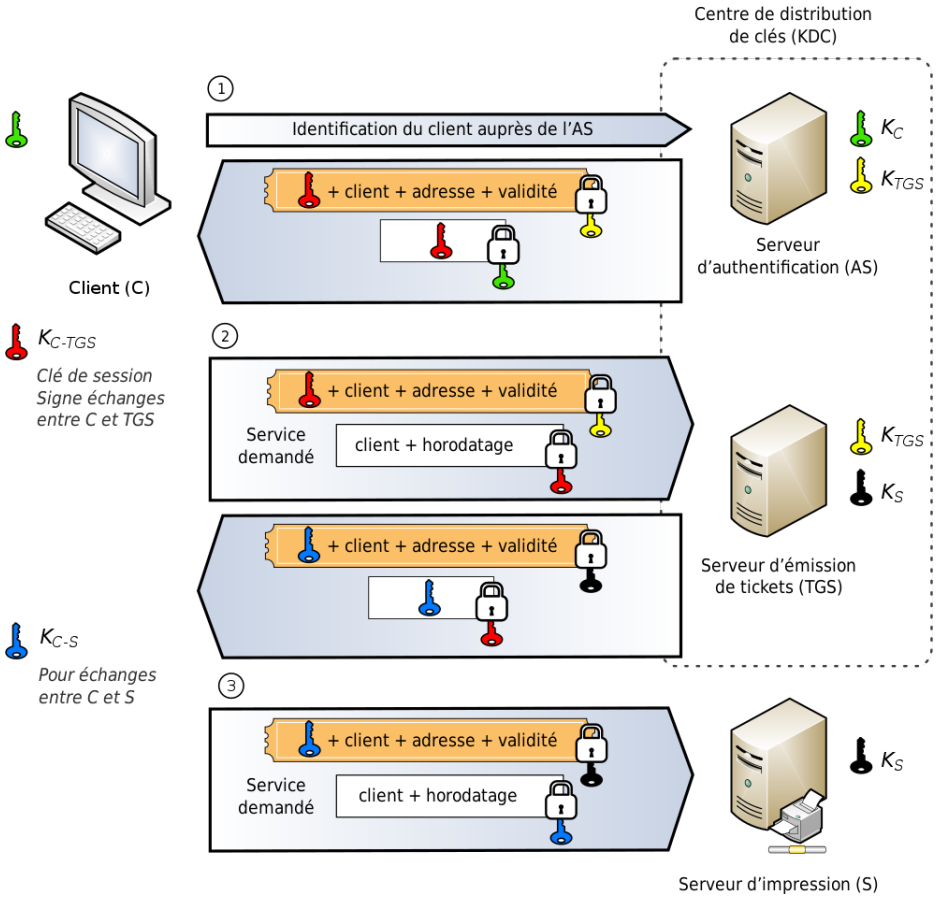


TACACS permet à un serveur d'accès distant de communiquer avec un serveur d'authentification pour savoir si l'utilisateur peut accéder au réseau. Ce protocole est non codé et donc moins sécurisé que les protocoles postérieurs de TACACS et de RADIUS.

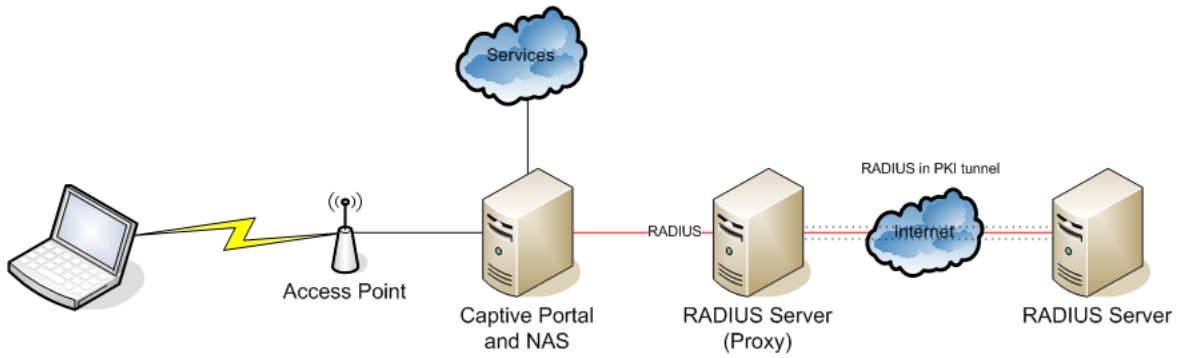
Le protocole TACACS, qui a un mécanisme de travail très simple, accepte une requête utilisateur d'un serveur distant et transmet cette requête au serveur d'authentification.
Le serveur d'authentification peut autoriser ou refuser une requête utilisateur au nom de l'hôte. Le résultat de la requête est envoyé à l'initiateur de la requête en tant que réponse de rétroaction. L'accès est donc donné en fonction de la réponse de la requête.

Kerberos :

Kerberos est un service distribué d'authentification qui permet à un procédé (un client) de prouver son identité à un vérificateur (un serveur d'application, ou serveur simplement) sans envoyer des données à travers le réseau qui pourrait permettre à un agresseur de les imiter postérieurement.
Kerberos fournit intégrité et confidentialité pour des données envoyées entre le client et le serveur. Le principe est le suivant :



RADIUS : Remote Authentication Dial-In User Service :



Le protocole RADIUS fonctionne selon un modèle client/serveur. Un NAS (Network Access Server) fonctionne comme un client RADIUS. Un client effectue des requêtes RADIUS et agit en fonction des réponses reçues. Un serveur RADIUS peut agir en tant que proxy RADIUS pour d'autres serveurs RADIUS, ainsi que pour d'autres systèmes d'authentification.
Toutes les transactions RADIUS sont authentifiées par l'utilisation d'un secret qui n'est jamais transmis sur le réseau. De plus, les mots de passe sont cryptés en utilisant cette même clé secrète.

2 OTP

2.1 Définition :

Un mot de passe à usage unique est un mot de passe qui n'est valable que pour une session ou une transaction dans un système informatique, il est implémenté dans une procédure d'authentification multi-facteurs ou authentification à double facteurs (MFA/2FA).
MFA est un processus de sécurité de compte nécessitant deux ou plusieurs étapes distinctes pour qu'un utilisateur prouve son identité, plus sauvant il se base sur un ID ou login + mot de passe.

2.2 Les type d'OTP :

HOTP :
HMAC-Based One-Time Password : ce type était initiative d'OATH.
Il génère des OTP en se basant sur deux informations :
- La première information concerne un «secret» ou «seed» connue uniquement du générateur d'OTP et du serveur qui valide les codes OTP soumis.
- La seconde information concerne le facteur de changement à savoir un compteur
En effet, au début le serveur génère un «secret» et le transmet au générateur d'OTP, ensuite, on passe à une fonction de hachage SHA1, la valeur du compteur «mixé» avec le «secret». Puis, on tronque le résultat du hash selon des règles non détaillées ici. Le générateur d'OTP incrémente le compteur et envoie l'OTP généré au serveur pour validation.

TOTP :
Time-based One Time password, comme son nom l'indique il est basé sur le temps pas sur un compteur comme HOTP, Il s'appuie également sur l'heure "Posix". De ce fait, on évite le problème de

synchronisation car toutes les horloges (celui du générateur d'OTP et du serveur) sont synchronisées avec peut-être un décalage infime.
En effet, le partage initial du «secret» entre le serveur et générateur reste le même. Cependant, la génération de l'OTP se fera avec le couple «secret» et heure «timestamp» sur une période définie (en général 30 à 60 secondes). Cela veut dire que le TOTP utilise le temps de manière incrémentale et que chaque OTP est valide pendant la durée de l'intervalle de temps.

2.3 Le fonctionnement d'OTP :

Il se base sur un algorithme qui génère un code nouveau et aléatoire à chaque fois s'appelle générateur OTP et un serveur d'authentification.
Le générateur OTP est basé sur :
- Ce que l'utilisateur possède ou connait
- Un «seed» fournit par le serveur

Le serveur d'authentification a pour rôle de :
- Générer un «secret» ou «seed» avec les bons paramètres puis il l'envoyer (QR code, SMS, voix ou mail) au générateur.
- Vérifier l'OTP reçu.
- Stocker le dernier OTP valide reçu avec le numéro de séquence correspondant à ce dernier
- Faciliter le changement du secret de l'utilisateur de manière sécurisée.

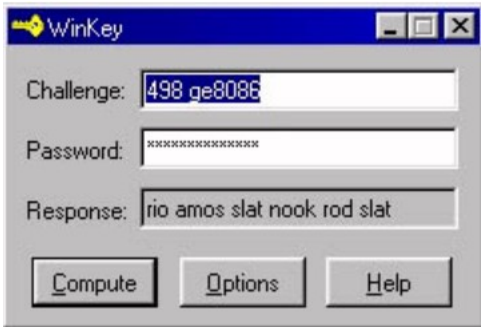
Une foix l'utilisateur veut s'authentifier, le générateur d'OTP transmet au serveur le secret de l'utilisateur avec un «seed» préalablement reçu du serveur, à travers plusieurs itérations d'une fonction de hachage sécurisée pour produire l'OTP. Après chaque authentification réussie, le nombre d'itérations de la fonction de hachage est décrémenté. Le serveur vérifie l'OTP reçu par le générateur d'OTP en appliquant une seule fois la fonction de hachage sécurisée utilisée par le générateur. Puis il fait une comparaison avec l'OTP qui avait été précédemment accepté

3 Les attaques sur l'authentification forte :

Malgré que l'authentification des mots de passe à usage unique est forte ils ont des failles de sécurité, cela les rendre attaquables.

Les attaques contre OTP :

Social engineering :
Certains programmes, comme WinKey, comportent un trou de sécurité : Tant que le programme de calcul n'est pas fermé, celui-ci garde en mémoire le mot de passe, Il existent des programmes dans l'environnement Win32 qui permettent de rendre visible ce mot de passe.

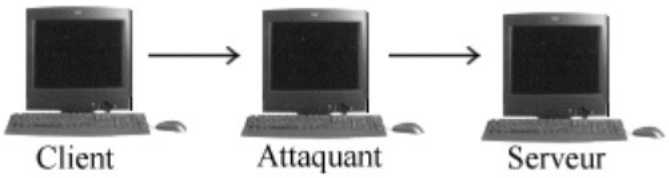


Attaque par cheval de Troie et keylogger :
Il suffit d'installer un cheval de Troie ou un keylogger sur le poste de l'utilisateur victime, pour essayer de récupérer le mot de passe de celui-ci.

L'accès au fichier skeykeys :
Si le hacker a la main sur le serveur, il est possible qu'il exploite une faille : Si le fichier /etc/skeykeys est disponible en lecture pour le hacker, il va récupérer Le login et la représentation hexa de l'OTP comme montré de l'image ci dessous. Ensuite il peut créer un outil qui générera la réponse au challenge en fonction d'un dictionnaire, ou par force brute.

login	numéro de séquence	semence	Représentation hexa de la réponse du challenge (le dernier OTP utilisé)	Date	Heure
victime	985	xv 5235b	5b3c89552aa09435	Jun 25, 2001	05:30:15

Attaque par milieu :
Cette attaque consiste pour l'attaquant à avoir la main sur une machine qui transmet l'authentification de la session. C'est une machine intermédiaire. Dans ce cas, l'attaquant peut écouter les trames qui transitent et repérer le challenge et la réponse du challenge. Là aussi il devra utiliser le cracking par dictionnaire ou par brute force.



Attaque par spoofing :
C'est une attaque par le milieu, mais un peu plus évolué. En effet, il faut en plus simuler exactement le comportement du serveur. On récupère ainsi le compteur (numéro de séquence) et la réponse au challenge (secret) Il est alors possible d'exploiter cela par deux techniques différentes :
• Soit essayer de trouver le mot de passe utilisateur par dictionnaire et brute force.
• Soit l'attaquant a utilisé consciemment un numéro de séquence inférieur à ce qu'attend vraiment le serveur. Dans ce cas, l'attaquant peut alors utiliser un nombre de connexion, égal à la différence

entre le vrai numéro de séquence serveur, et le faux numéro de séquence de l'attaquant.

Attaque par Hi-Jacking :
L'authentification se fait au moment de la connexion. Passé cette authentification, il n'y a plus d'autres moyens de vérifier si l'authentification est toujours correcte. Aussi, il est possible de détourner la communication TCP/IP. C'est une attaque de type Hi-jacking. Une fois le détournement effectué, le serveur dialogue avec l'attaquant, et l'attaquant utilise la session de la victime.

4 TOTP

4.1 Mise en contexte

Après avoir définir le TOTP ci-dessus, voyons le avec plus en détail :
Les utilisateurs d'Internet sont régulièrement amenés à saisir des mots de passe pour se connecter aux réseaux sociaux, en faisant leur shopping en ligne ou pour consulter leur compte sur Internet.
Les mots de passe permettent de sécuriser les données sensibles contre un accès non autorisé par des tiers. Et par la suite garder la confidentialité des données qui est un enjeu essentiel dans la sécurité.
Toutefois, de nombreux utilisateurs ne se donnent pas la peine de prendre les précautions nécessaires : les mots de passe simples peuvent être craqués par les professionnels en quelques secondes seulement. D'autres personnes prennent soin de bien choisir leurs mots de passe, mais elles les conservent d'une mauvaise façon et offrent ainsi une porte d'entrée aux criminels.
Les points faibles des services auxquels on se connecte en tant qu'utilisateur ne doivent pas non plus être négligés. Si les mots de passe n'y sont pas conservés de façon sécurisée, ce sont les données de milliers d'utilisateurs qui sont alors en péril.

L'authentification à double facteur ou l'authentification multifactorielle permet de réduire ce risque. Plutôt que d'utiliser uniquement un mot de passe, il est au minimum nécessaire d'indiquer un autre facteur d'authentification.

Ce facteur est envoyé à l'utilisateur de cette technologie par téléphone ou via un token (jeton d'authentification). La plupart du temps, les facteurs supplémentaires ont en commun d'être générés une unique fois et d'être valables pour une durée limitée générant ainsi un Time-based One-time Password (TOTP).

4.2 Objectifs d'utilisation de TOTP

Les mots de passe habituels présentent un inconvénient de taille même s'ils sont choisis d'une façon minutieuse : dès qu'une autre personne connaît cette suite de caractères, la sécurité est compromise.

Une solution serait de changer régulièrement de mot de passe, mais même les utilisateurs les plus exemplaires ne le font que rarement.
La solution est un TOTP : un mot de passe valable uniquement pour un bref laps de temps avant d'expirer à nouveau. L'Internet Engineering Task Force (IETF) a publié l'algorithme Time-based One-time Password en 2011 dans la RFC 6238 afin d'apporter une plus grande sécurité sur Internet.
Ces mots de passe uniques sont tout particulièrement populaires dans le cadre d'une authentification multifactorielle. Dans ce type d'authentification, les utilisateurs saisissent tout d'abord leur mot

de passe personnel, qui reste identique, pour se connecter à un service en ligne ; un mot de passe à durée limitée dédié à ce processus de connexion est par ailleurs généré. L'utilisateur peut obtenir ce mot de passe via une application ou à l'aide d'un dispositif supplémentaire prévu à cet effet (token).

Le mot de passe expire s'il est utilisé ne serait-ce qu'une seule fois ou s'il n'est pas utilisé pendant une période donnée. Il est ainsi très difficile pour les cybercriminels de récupérer le second facteur. Même s'ils connaissent le mot de passe permanent, ils n'ont que peu de possibilités d'acquérir le TOTP, et pas assez de temps pour le craquer.

Chapitre 2

Mise en pratique de TOTP

1 Fonctionnement de TOTP

Le TOTP est basé sur une fonction de hachage, c'est-à-dire un procédé cryptographique. On utilise un mot de passe secret et un horodatage pour créer une séquence de caractères cryptée. Le mot de passe est aussi bien connu de l'utilisateur que du serveur. L'indication temporelle est effectuée en temps Unix (indiquant les secondes écoulées depuis le 1er janvier 1970).

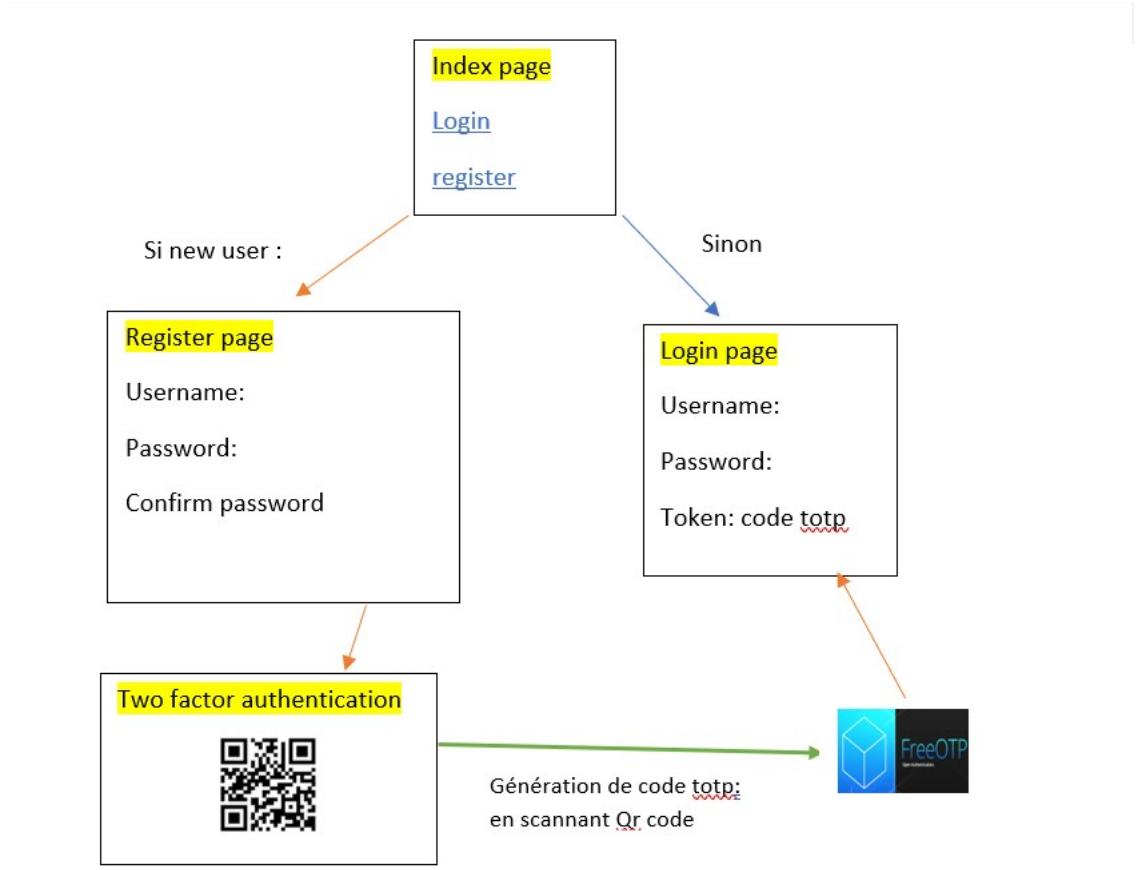
Le TOTP est en fait une amélioration du « HMAC-based One-time Password » abrégé en HOTP expliqué ci-dessus. Le TOTP est également basé sur la procédure HMAC, l'opération de hachage qui se déroule en arrière-plan. En associant le mot de passe secret à un compteur, l'appareil de l'utilisateur et le serveur génèrent tous deux une valeur de hachage. Les deux valeurs sont identiques, permettant ainsi l'authentification.
La fonction de hachage elle-même n'est pas fixe ; en pratique, on peut utiliser SHA-1 (le cas du Google Authenticator), qui génère une valeur de hachage d'une longueur de 160 bits. Dans un souci de simplicité, cette valeur est encore raccourcie à l'aide d'une fonction de compression. À terme, on obtient par exemple un nombre à six chiffres, que les utilisateurs peuvent alors saisir en toute simplicité lors de leur connexion au service en ligne.

Pour le second élément de la fonction, le HOTP utilise un compteur partagé par le serveur et l'utilisateur. Dans ce cas, le problème est que le mot de passe généré est valable jusqu'à ce qu'il soit utilisé. TOTP y ajoute une restriction : le code généré ne peut être utilisé que pendant une période limitée.

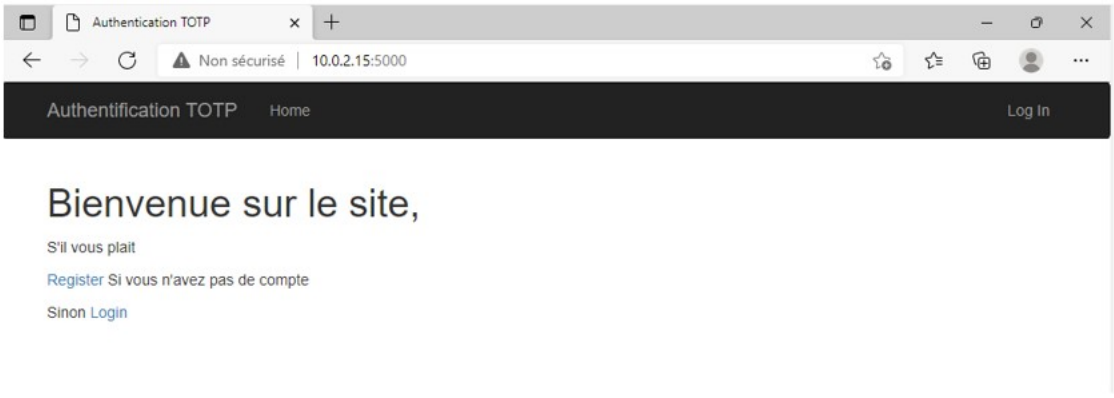
En guise de conclusion :
TOTP = HOTP(SecretKey,CurrentTime)
où SecretKey : mot de passe généré au hasard et connu aussi bien du serveur que du client
CurrentTime : moment actuel en temps Unix

2 Implémentation d'un site supportant la méthode TOTP pour l'authentification :

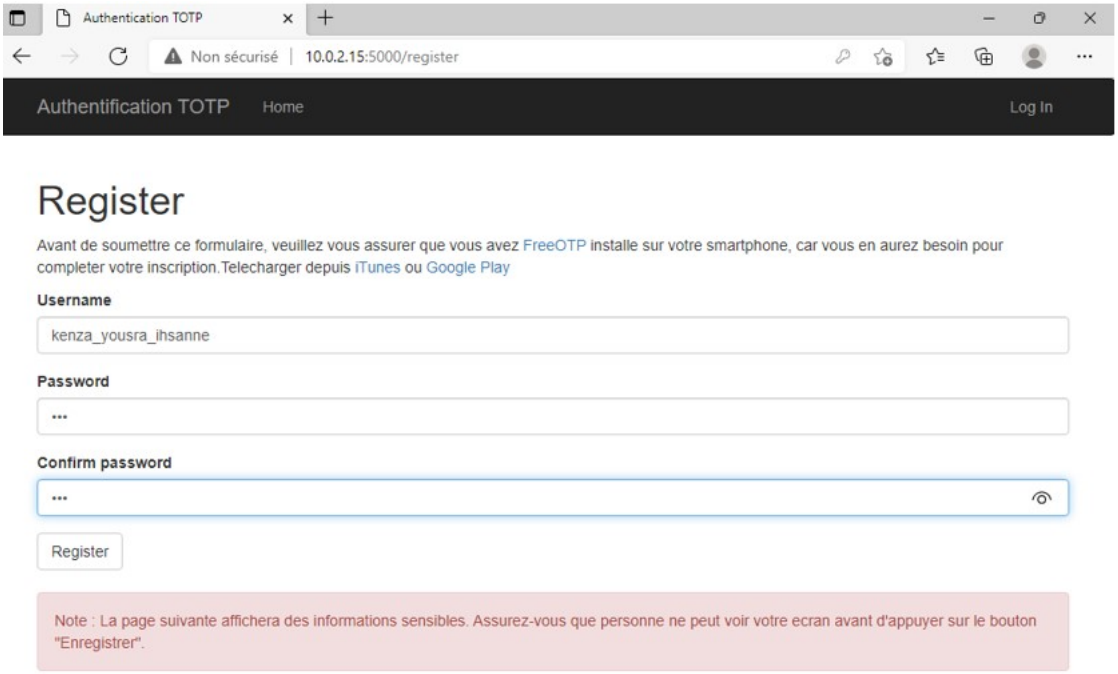
Pour implémenter une application web fonctionne avec l'authentification TOTP, on a utilisé le framwork web de python Flask. Voici un schéma qui explique en gros comment se fait l'opération d'authentification :



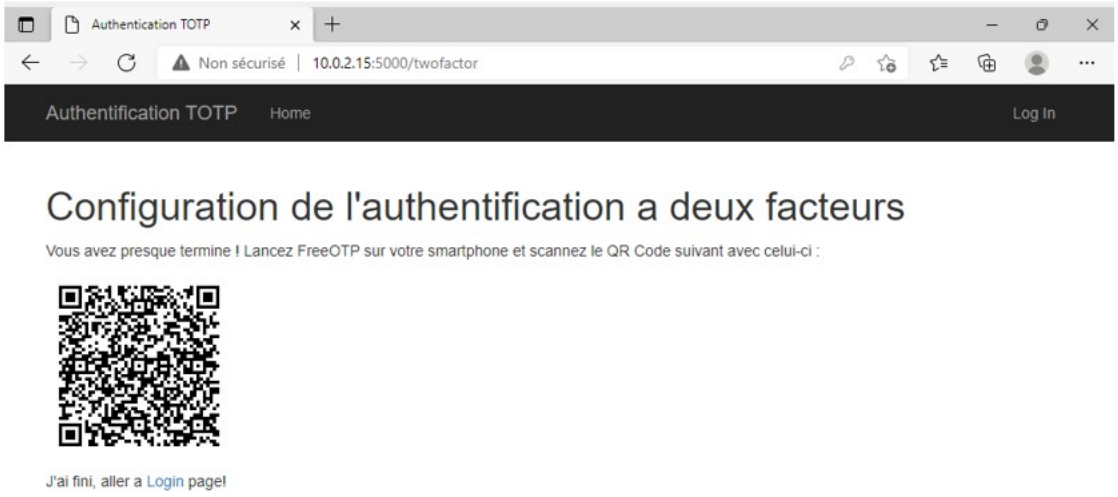
La première chose que l'utilisateur va faire c'est de s'authentifier s'il est déjà enregistré sinon il va créer un compte



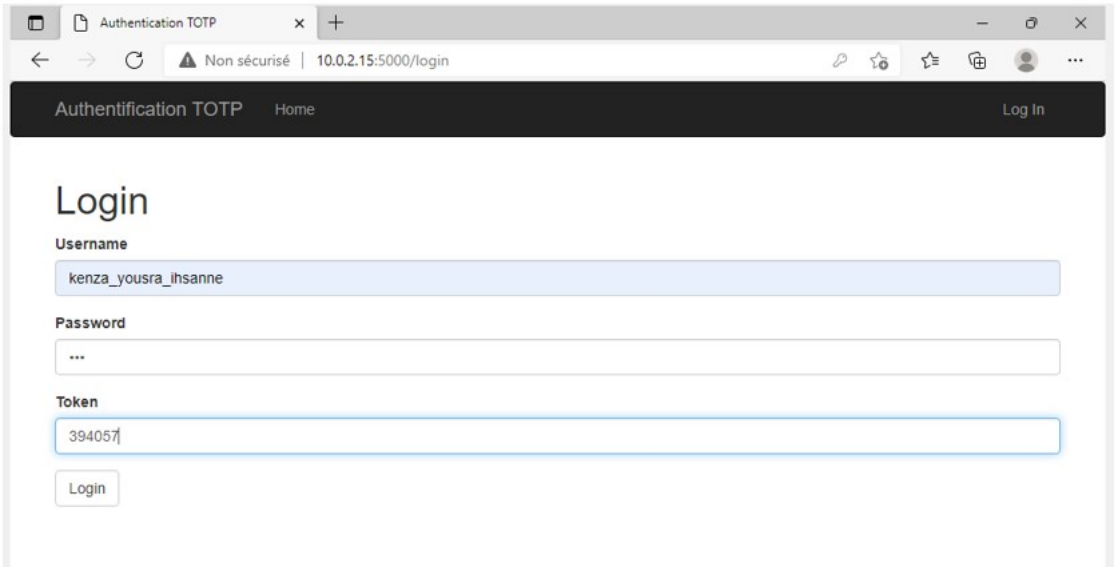
Pour s'enregistrer il va entrer un nom utilisateur et un mot de passe :

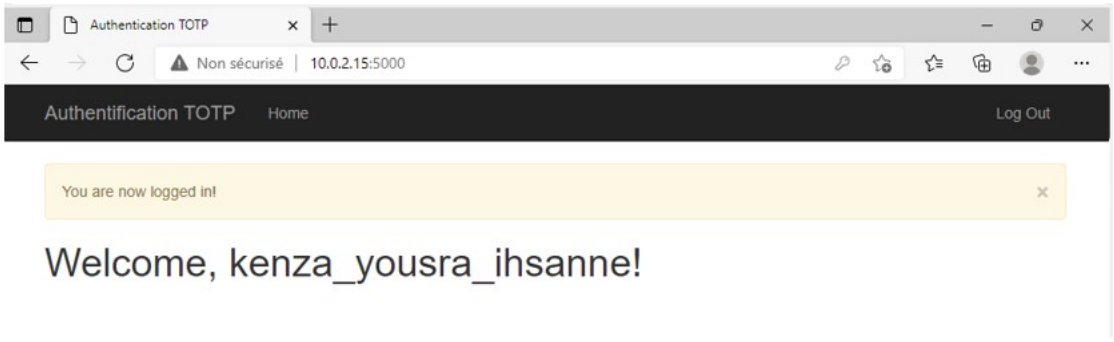


Puis il va scanner ce code QR via l'application FreeOtp pour générer un code Totp :



Ensuite il va entrer le code générer pour s'authentifier, ce code change chaque 60 second.





Lien de code :

Veuillez trouver le code source de notre application sur le lien github suivant : https://github.com/YousraMzouz/totp-app

Chapitre 3

Conclusion générale

Notre projet consiste à la réalisation d'un site web avec la méthode d'authentification TOTP à travers les différentes étapes qui commencent par la réalisation de l'authentification (l'inscription et la connexion) puis la génération du code QR qui participe à l'authentification de l'utilisateur.

Ce projet nous a été une occasion précieuse pour se familiariser avec la pratique des méthodes d'authentification, au niveau professionnel, il nous a permis de mettre en œuvre les connaissances et les compétences acquises tout au long de notre formation et d'assumer la responsabilité qui nous a été confiée.

Et sur le plan personnel, cette expérience a aiguisé nos capacités du travail en équipe dans ces conditions loin de l'ordinaire et a surtout fortifié notre motivation, détermination et notre ambition de suivre une carrière dans le domaine cybersécurité.

Chapitre 4

Bibliographie :

http ://jmainy.free.fr/guill.web-/Authentification.html

https ://fr.wikipedia.org/wiki/Authentification

http ://www.guill.net/

https ://fr.wikipedia.org/wiki/Password_Authentication_Protocol

https ://idento.fr/one-time-password-otp/

https ://www.securiteinfo.com/cryptographie/otp.shtml

https ://www.ionos.fr/digitalguide/serveur/securite/totp/

https ://fr.microcosm.com/blog/hotp-totp-what-is-the-difference