ITÉRATION 4

2FA et SSH

Modalités

- Travail individuel en autonomie
- 1 jour en présentiel

Objectifs

Connaître les grands principes de l'authentification multi-facteurs et les différents moyens qui pourraient être utilisés.

Connaître les principales techniques de sécurisation des sessions ssh comme l'utilisation de clés, la désactivation de la connexion root.

Compétences

- Connaissance de l'authentification multifactorielle
- Connaissance des règles de sécurisation des accès ssh

1.1 — L'authentification multifactorielle

3h — Présentiel

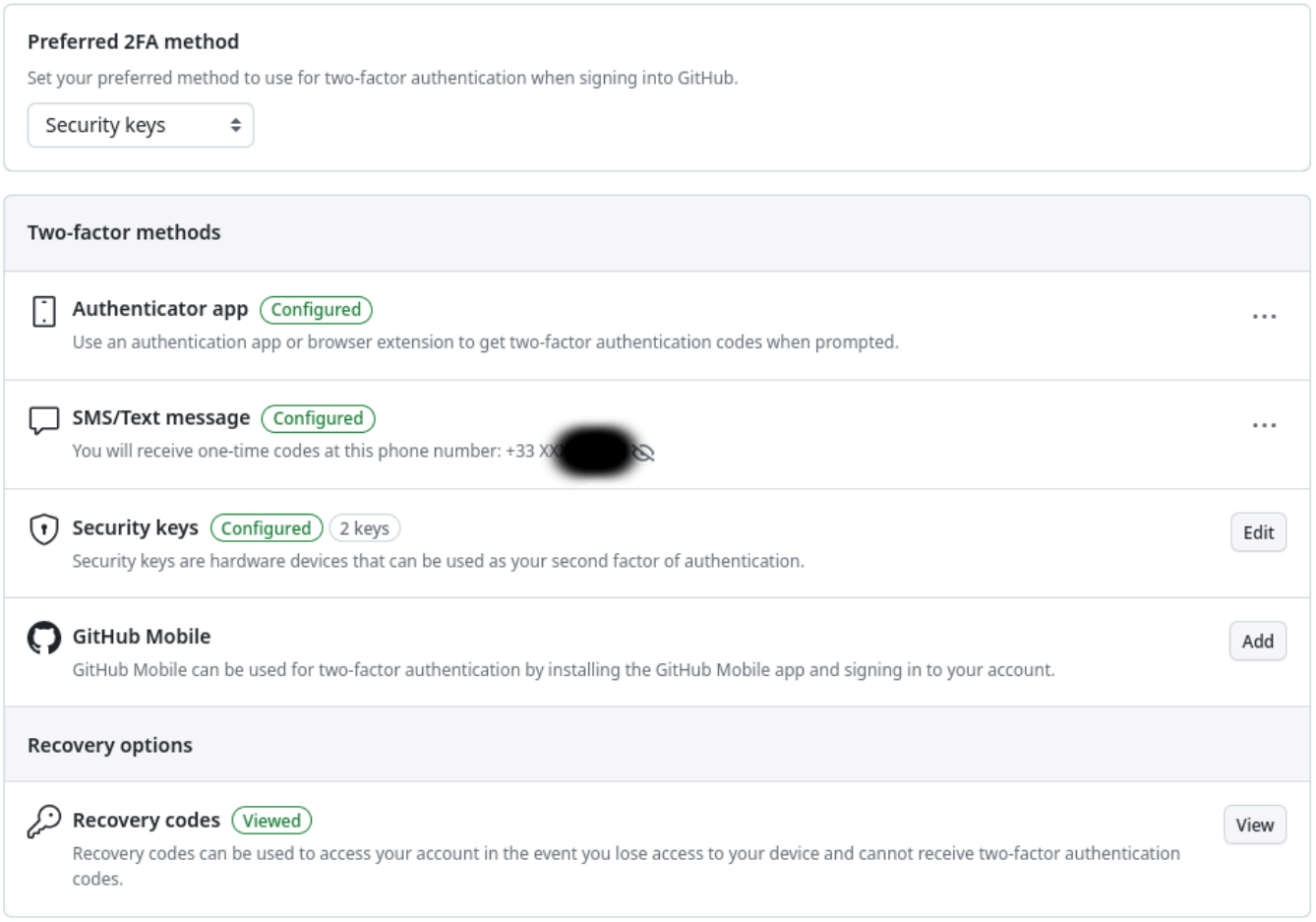
L'une des techniques de protection fréquemment utilisées consiste à utiliser plusieurs moyens. Si l'un d'eux a un problème, d'autres couches protègent toujours le système. Un exemple de cela pourrait être l'utilisation d'une protection physique d'un serveur (avec un badge par exemple) en plus d'un identifiant et d'un mot de passe.

Les fuites de mots de passe sont assez courantes, d'autant plus que de nombreux utilisateurs ont le même mot de passe sur plusieurs sites. Pour éviter de compromettre un compte même si le mot de passe fuit, nous utilisons ce que l'on appelle l'authentification multifacteur. C'est par exemple la nécessité d'entrer un code reçu par SMS en plus d'un mot de passe.

2FA est l'abréviation de "two-factor authentication", un cas particulier d'autorisation multifactorielle.

Nous pouvons utiliser le même pour l'accès à des sites importants, par exemple notre code. GitLab et GitHub nous permettent d'activer l'authentification multifactorielle. Nous pouvons utiliser un code SMS, un code temporisé (TOTP), un jeton matériel...

La configuration de GitHub peut ressembler à ceci (dans “Passwords and authentication”) :



Dans cet exercice, créez une nouvelle organisation avec un projet et configurez l'authentification multifactorielle de votre choix. La prochaine étape consiste à rendre le multi-facteur obligatoire pour rejoindre l'organisation.

Créez un nouveau projet GitLab avec 2FA activé. Clonez le code avec HTTPS, faites un changement et poussez-le. Qu'est-ce qui est différent ?

Attention : ne perdez pas le code supplémentaire et enregistrez les codes de récupération, sinon il pourrait être impossible d'accéder à votre compte en cas de problème.

> **Note :** Récemment, les plateformes de code (GitLab, GitHub…) ont commencé à rendre le 2FA obligatoire au moins pour certains comptes. Si vous contribuez à des projets importants, ou si vous contribuez beaucoup, il vous sera peut-être impossible de désactiver 2FA. L'organisation à laquelle vous contribuez (le projet ou l'entreprise) peut également rendre la 2FA obligatoire.

Le CISA (agence de cybersécurité américaine) insiste sur le fait d'implémenter un MFA résistant au phishing. Ref :
https://www.cisa.gov/sites/default/files/publications/fact-sheet-implementing-phishing-resistant-mfa-508c.pdf
Quelle est la différence entre un MFA et un MFA résistant au phishing ?

RESSOURCES

- Configuration sur GitHub : https://docs.github.com/en/organizations/keeping-your-organization-secure/managing-two-factor-authentication-for-your-organization/requiring-two-factor-authentication-in-your-organization
- Configuration sur GitLab : https://docs.gitlab.com/ee/user/profile/account/two_factor_authentication.html
- Tokens sur GitLab : https://docs.gitlab.com/ee/user/profile/personal_access_tokens.html

COMPÉTENCES ASSOCIÉES

- Connaissance de l'authentification multifactorielle

1.2 — SSH

4h — Présentiel

SSH est l'une des méthodes d'accès fréquentes. Dans ce module, nous allons préparer quelques changements dans la configuration pour la rendre plus sécurisée.

- La première étape consiste à supprimer l'accès par mot de passe. Configurez-le pour n'accepter que les clés EC 25519.
- Assurez-vous que la connexion "root" n'est pas possible.
- Les serveurs SSH écoutant sur le port par défaut 22 reçoivent de nombreuses requêtes de connexion (pourquoi ? Plus d'informations dans l'article sur les honeypots SSH). Choisissez un autre port pour votre serveur (ce n'est pas vraiment une mesure de sécurité forte, mais nous permet d'avoir moins d'entrées de journal), par exemple: 15022.
- Autoriser un seul utilisateur

RESSOURCES

- "OpenSSH security and hardening" https://linux-audit.com/audit-and-harden-your-ssh-configuration/
- "Creating an SSH honeypot" sur LWN.net https://lwn.net/Articles/848291/

COMPÉTENCES ASSOCIÉES

- Connaissance des règles de sécurisation des accès ssh

1.3 - Analyse de sécurité avec Lynis (bonus)

Après avoir mis à jour la configuration de votre système, utilisez **Lynis** pour vérifier votre configuration. Lynis est un exemple de scanner de sécurité permettant d'obtenir rapidement un rapport sur le niveau de sécurité d'une machine. Différents outils similaires existent, avec différents outils d'administration et tableaux de bord.

RESSOURCES

- Lynis : https://cisofy.com/lynis/

COMPÉTENCES ASSOCIÉES

- Connaissance des règles de sécurisation des accès ssh

1.4 - Audit avec les outils de Kali Linux (bonus)

Nous effectuons souvent un audit d'une application existante. Cela se fait soit depuis le système (comme dans l'exercice avec Lynis), soit depuis l'extérieur pour simuler des accès externes.

Attention : effectuez toujours de tels tests sur vos propres systèmes et utilisez de préférence le système de pré-prod et/ou de test.

Utilisez une VM avec une application Web (ça sera votre système testé, DUT - *device under test*). Installez en plus une image de Kali et utilisez les outils suivants pour découvrir les problèmes possibles avec l'application Web :
tiger - https://www.kali.org/tools/tiger/
wapiti - https://www.kali.org/tools/wapiti/
nikto - https://www.kali.org/tools/nikto/

RESSOURCES

- Kali Linux https://www.kali.org/

1.5 - Scan de votre dépôt git pour les informations d'identification oubliées (bonus)

Les attaquants pourraient utiliser d'anciennes informations d'identification qui ont été oubliées dans les branches abandonnées d'un dépôt git. Des outils comme Trufflehog permettent de trouver de tels usages.

Utilisez trufflehog de l'installation de Kali pour analyser certains de vos repos personnels. A-t-il trouvé quelque chose d'étrange ?

RESSOURCES

- Trufflehog https://www.kali.org/tools/trufflehog/

Livrables

Dans ce projet, nous devrions avoir :

- ➔ Une configuration GitHub/GitLab (au choix) avec l'authentification multifactorielle
- ➔ Une configuration améliorée du ssh