ESUP-OTP : Solution libre et open source pour la gestion de l'authentification forte

Aymar Anli
Pôle Applications et Services Numériques (PAS)
Direction du Système d'Information et des Usages Numériques (DSIUN)
Université Paris 1 Panthéon-Sorbonne

Vincent Bonamy
Pôle Études & Développements
Direction des Systèmes d'Information
Université de Rouen Normandie

Résumé

En 2017, nous avions eu le plaisir de présenter le projet ESUP-OTP (Esup-Portail One Time Password) au travers d'un article et d'une présentation longue aux JRES à Nantes.

Cette présentation revenait sur une mise en place de l'authentification multi-facteurs au travers de la solution de SSO (Single Sign On) CAS (Central Authentication Service) de Apereo en prenant comme exemple sa mise en place dans la connexion au VPN (Virtual Private Network) d'un établissement.

Cette mise en œuvre effective à l'Université de Paris 1 Panthéon-Sorbonne avait alors nécessité quelques ajustements non triviaux : instanciation d'un CAS et d'un IdP Shibboleth spécifique à l'OTP notamment.

Presque 5 ans ont passé depuis.

Ce que l'on présentait comme novateur par rapport aux technologies à disposition est devenu à portée.

Les objectifs numériques ont évolué. La crise sanitaire a également eu un impact fort et durable sur le fonctionnement de nos établissements.

Nos utilisateurs sont devenus tous nomades, disposant pour la plupart de smartphones et d'ordinateurs portables.

Dans cet article, on montre qu'ESUP-OTP permet de proposer à nos utilisateurs une authentification multi-facteurs opérée via une solution libre et auto-hébergée, ce sur les applications cassifiées et shibbolethisées. Cet article vient donc en complément du précédent article [1] proposé lors des JRES à Nantes.

Mots-clefs

CAS, ESUP-OTP, OTP, MFA, VPN, NFC, EsupPortail, Logiciel Libre, authentification multi-facteurs, souveraineté numérique.

1 Introduction

Alors que la plupart des établissements de l'ESR (Enseignement Supérieur et Recherche) utilise comme serveur central d'authentification la solution Apereo CAS, celle-ci propose nativement bon

nombre de fournisseurs de MFA (Multi-Factor Authentication). Citons Duo Security, Twilio Authy, Acceptto, YubiKey, WiKID, FIDO, Swivel Secure, Google Authenticator.

Malheureusement, la plupart ne sont pas libres, sont chers et ne sont pas disponibles en mode auto-hébergé.

En tant que consortium ESUP-Portail, nous fournissons une alternative nommée ESUP-OTP.

2 Objectifs

2.1 Libre

ESUP-OTP a été développé par nos soins sous la licence libre MIT. Tout est ainsi ouvert gratuitement à tous : l'ensemble des sources, documentations et présentations[1].

2.2 Auto-hébergé

Les fournisseurs de MFA disponibles sur le marché proposent des solutions techniques opérées par leurs propres services. Ces solutions fonctionnent sur les infrastructures d'hébergeurs dans le « cloud » en proposant des API permettant d'ajouter un facteur d'authentification dans l'implémentation de son propre service d'authentification.

Si la consommation d'offres de services en mode SaaS (Software as a Service) tend à se généraliser, elle n'est pas pour autant sans poser question.

La pérennité de la solution est liée directement et sans délai à la survie du prestataire. La fiabilité de la solution est fonction également de ce dernier, de son hébergeur mais aussi des connectiques avec celui-ci. Le coût de la solution est non maîtrisé puisque correspondant à un abonnement souvent mensuel, fonction de l'usage potentiel ou réel et évolutif.

À ces considérations, applicables à toutes solutions propriétaires et notamment celles proposées en SaaS, on peut bien sûr objecter que l'entreprise fournissant le service a une maîtrise parfaite de son outil qui lui permet de proposer une solution fiable, sécurisée, à jour, et à coût réduit puisque mutualisée pour l'ensemble de ses clients.

Sur une solution liée à l'authentification et *in fine* au renforcement de l'authentification centrale de l'établissement, et donc au renforcement global de la sécurité, il paraît préférable de disposer d'une solution fortement intégrée au Système d'Information opérée en interne au même titre que les services d'authentification eux-mêmes. Mis en œuvre et maîtrisé techniquement par l'établissement, ESUP-OTP permet de disposer d'une solution souveraine, auto-hébergée et ainsi d'éviter que ces briques sensibles soient sous la responsabilité d'entreprises étrangères.

2.3 Simple d'installation

Solution financée et distribuée par le consortium Esup-Portail et initiée et développée par l'Université Paris 1 Panthéon-Sorbonne en collaboration avec La Rochelle Université, ESUP-OTP est réalisé par et pour un établissement de l'ESR.

À ce titre, son installation et sa mise en place se font dans un environnement technique usuel pour nos institutions. Les 2 briques qui proposent réellement le service s'installent ainsi aisément et rapidement sur une distribution Linux. Elles sont adaptées à être servies derrière un proxy de type Apache ou NGINX. Liée au service central d'authentification CAS d'Apereo, la brique esup-otp-

1 https://www.esup-portail.org/wiki/display/CAS/MFA+Esup+OTP

cas est un module CAS qui s'active comme les autres modules CAS. Un fichier de paramétrage lui est dédié dans /etc/cas/config.

2.4 Simple d'usage

Pour l'utilisateur, une interface web riche, ergonomique et intuitive est proposée. Celle-ci lui permet de choisir les facteurs d'authentification souhaités. Par rapport aux solutions du marché, cette fonctionnalité apporte la souplesse nécessaire permettant de convaincre nos utilisateurs d'adhérer à ce service contraignant mais nécessaire.

2.5 Gestion facilitée

Une interface web est également proposée aux gestionnaires pour qu'ils puissent dépanner les utilisateurs en cas d'oubli du téléphone par exemple.

3 Architecture logicielle

En plus d'un module à ajouter dans le serveur CAS lui-même, ESUP-OTP est composé de plusieurs applications permettant de la MFA adaptée à l'organisation de nos établissements : interface web à destination des utilisateurs finaux (étudiants et personnels) et aux gestionnaires (informaticiens de proximité).

3.1 Applications côté serveur

Côté serveur, ce sont 2 applications en Node.js qui assurent le service à proprement parler.

- esup-otp-api est la brique métier qui calcule les jetons et autres secrets. Elle est également responsable de la persistance en base de données MongoDB. Enfin, elle offre une API REST ;
- esup-otp-manager est la partie frontale d'ESUP-OTP. Elle présente une interface web réactive à l'usager en dialoguant avec esup-otp-api.

Côté CAS, c'est donc esup-otp-cas qui est chargé, en tant que module CAS exposant un facteur d'authentification, d'intégrer les fonctionnalités du projet ESUP-OTP. D'un point de vue logiciel, l'idée a été de minimiser au mieux le code de ce module. Le maintien de ce module CAS et sa compatibilité avec les versions de CAS sont complètement dépendants des évolutions de l'application CAS elle-même ; et comme les exploitants de CAS le savent, le logiciel CAS évolue rapidement !

3.2 Applications côté client

3.2.1 Codes aléatoires à usage unique

Pour la MFA par codes aléatoires à usage unique, aucune application cliente n'est nécessaire. Il est possible d'envoyer ces codes par SMS et/ou par mail.

Pour les envois par SMS, esup-otp-api s'appuie sur ESUP-SMSU [5]. ESUP-SMSU repose lui-même sur un « broker » pour l'envoi effectif du SMS. Le broker est généralement un fournisseur proposant en SaaS ce service. On fait alors face aux différents problèmes évoqués ci-avant sur ce type de service, notamment sur celui du coût fluctuant. Actuellement (2021), en France le SMS est facturé autour de 5 centimes d'euros.

3.2.2 Codes de secours à imprimer

Cette possibilité MFA consiste à laisser à l'utilisateur le soin d'imprimer une liste de codes à usage unique.

C'est donc avant tout une option de dépannage qui est proposée à l'utilisateur et qui a le mérite de ne nécessiter aucun périphérique ou connexion.

3.2.3 TOTP

Pour la MFA TOTP (Time-based One-Time Password), plusieurs possibilités sont proposées à l'utilisateur. Google Authenticator, disponible aussi bien sur Google Play Store que Apple Store, est sans doute le client le plus connu pour ce type de facteur d'authentification. Il n'est cependant pas le seul, citons FreeOTP, équivalent de Google Authenticator sous licence libre. Notons également que le TOTP n'est pas uniquement disponible au travers d'applications sur périphériques mobiles. Des extensions/plugins sur des navigateurs existent par exemple. On trouve même des utilitaires simples en ligne de commande prenant en entrée le secret (qui correspond usuellement au scan du qr-code) ; ainsi l'usage de oathtool sous Linux, par exemple, peut suffire :

```
oathtool -b --totp AAAAAAAAAZZZZZZZ
```

Moins ergonomique et pratique que le facteur par Push avec ESUP-AUTH, le facteur par TOTP a le mérite d'être disponible sur toutes les plateformes possibles en ne nécessitant donc finalement pas de matériel ou de connexion supplémentaire.

À noter que nous allons rajouter prochainement le TOTP dans l'application mobile Esup Auth. L'utilisateur pourra aussi bien utiliser Esup Auth pour le push mobile que pour le TOTP. Il pourra aussi transférer ou utiliser cette application pour ses autres services professionnels ou personnels nécessitant un code TOTP (Google, GitHub, ...). En termes de communication, il nous apparaît plus simple d'inviter nos usagers à installer une application institutionnelle (Esup Auth) pour faire du TOTP.

3.2.4 Push

Côté client, ESUP-OTP propose d'utiliser l'application Esup Auth [4] disponible sur Google Play Store pour Android ... et bientôt disponible sur Apple Store pour iOS. Cette application permet de mettre en œuvre la MFA Push.

3.2.5 Carte sans contact

Dernière implémentation en date, cette possibilité est à mettre en lien avec le projet ESUP-NFC-TAG. L'idée ici est que le facteur d'authentification soit le fait que l'utilisateur possède sa carte multi-services (carte étudiante, professionnelle ou d'invité délivrée par son établissement). Pour le prouver, il doit badger sa carte sur un dispositif NFC : un smartphone, un PC doté d'un lecteur NFC ou encore un boîtier électronique spécifique (Arduino). Ces éléments sont dépendants de l'installation et la mise en place du projet ESUP-NFC-TAG dans l'établissement. Ce dernier est en effet requis ici pour cette fonctionnalité, et il propose des implémentations de badgeage pour chacun des périphériques sus-cités.

4 Couverture fonctionnelle

4.1 Pour l'utilisateur final

ESUP-OTP permet à l'utilisateur final de gérer ses mots de passe à usage unique. Il peut activer/désactiver une ou plusieurs des méthodes disponibles, renseigner son numéro de portable ou appareiller son smartphone pour le *push* mobile. L'utilisateur peut aussi réinitialiser ses OTP en générant, par exemple, une nouvelle liste de codes de secours, une nouvelle clé TOTP ou en appareillant un nouveau smartphone. La réinitialisation d'une méthode OTP invalide automatiquement les anciens codes.

4.2 Pour le gestionnaire

Il est possible de déclarer une liste nominative de gestionnaires, via l'appartenance à un groupe ou via des valeurs particulières d'attributs. Les gestionnaires peuvent activer/désactiver des méthodes d'un utilisateur, renseigner un numéro mobile ou réinitialiser les OTP d'un utilisateur.

À noter que les gestionnaires ne peuvent en aucun cas afficher l'intégralité du numéro de téléphone renseigné, les secrets de génération de TOTP ou les listes des codes de secours déjà générés.

4.3 Pour l'administrateur

L'interface d'administration fournit un moyen simple à l'administrateur de sélectionner les méthodes (TOTP, Push, Codes aléatoires, NFC et Codes de secours à imprimer) et les transports (SMS et Mail) qui doivent être disponibles pour les utilisateurs finaux. Les modifications sont prises en compte dynamiquement.

5 Installation et Intégration dans CAS

5.1 Installation de ESUP-OTP

L'installation d'ESUP-OTP côté serveur correspond à déployer :

- esup-otp-api et esup-otp-manager, deux applications web fonctionnant avec Node.js via l'outil npm ;
- esup-otp-cas qui est un module CAS.

Les fichiers README de ces trois projets proposés depuis les sources même permettent de rapidement installer et faire fonctionner ces trois briques côté serveur.

5.2 Activation de la MFA

esup-otp-cas répond aux spécifications de module de MFA dans CAS. Aussi l'intégration d'ESUP-OTP dans CAS reprend les possibilités offertes par CAS pour ce type de modules. La documentation de CAS à ce sujet est donc applicable à ESUP-OTP. Dans ce paragraphe, on donne quelques pistes de configuration possibles qui nous semblent appropriées au contexte d'usage des établissements de l'ESR.

Remarquons que les exemples de configuration donnés correspondent à un CAS en version 6.4.4 (dernière version disponible au moment où nous écrivons cet article). Les noms de propriétés et possibilités de configuration évoluent au fil des versions de CAS qui reste un projet très actif. Aussi

merci de vous référer à la documentation officielle d'Apereo CAS si vous souhaitez effectivement adapter ces configurations à votre propre version de CAS[2].

5.2.1 Global

L'activation d'une MFA de manière globale est très simple sur CAS.

Dans le fichier cas.properties il suffit de positionner la propriété suivante :

```
cas.authn.mfa.triggers.global.global-provider-id=mfa-esupotp
```

Avec une telle configuration, CAS demandera une authentification renforcée par ESUP-OTP quel que soit le service consommé par l'utilisateur. Cette configuration a donc l'avantage d'être simple, sûre et efficace. Lors de vos tests d'intégration d'ESUP-OTP dans CAS, c'est sans doute la solution à privilégier.

5.2.2 Pour un service donné

CAS peut également vous laisser la possibilité d'activer une MFA uniquement pour un service donné. Si vous enregistrez vos services via de simples fichiers json à plat, la configuration pourra ressembler à celle-ici :

```
{
  "@class" : "org.apereo.cas.services.RegexRegisteredService",
  "serviceId" : "^https://mon-service-important\.univ\.ville\.fr$",
  "id" : 42,
  "name": "mon-service-important",
  "multifactorPolicy" : {
    "@class" : "org.apereo.cas.services.DefaultRegisteredServiceMultifactorPolicy",
    "multifactorAuthenticationProviders" : [ "java.util.LinkedHashSet", [ "mfa-esupotp" ] ]
  }
}
```

5.2.3 Par script groovy

Si les deux manières proposées ci-dessus pour activer une MFA semblent simples, elles manquent en fait de souplesse et peuvent poser problème si l'on souhaite organiser des règles propres à son établissement, à son Système d'Information.

Aussi, CAS propose d'activer ou non une MFA en fonction du retour d'exécution d'un script groovy.

```
cas.authn.mfa.groovy-script.location=file:/etc/cas/config/mfaGroovyTrigger.groovy
```

En d'autres termes, l'idée est d'implémenter en groovy les règles permettant, pour un contexte d'authentification donné, de définir si l'utilisateur doit procéder ou non à de la MFA.

En entrée du script groovy, on trouve les paramètres suivants :

- le service cible ;
- l'utilisateur, c'est-à-dire son login et champs récupérés depuis la première phase d'authentification (en général des champs issus du LDAP) ;
- la requête avec donc par exemple l'IP de l'utilisateur ;

2 https://apereo.github.io/cas/

- le *logger*, afin de pouvoir écrire des messages informatifs ou de *debugging* depuis le script groovy ... en s'assurant au préalable d'avoir une version de log4j2 à jour[3].

Avec ces quelques paramètres et la souplesse d'un script groovy, il est alors possible d'implémenter tout type de règles. On peut par exemple demander de la MFA pour certains services uniquement aux utilisateurs enseignants (eduPersonAffiliation à teacher) tentant de se connecter depuis une IP « extérieure ».

Voici un exemple simple d'un tel script groovy :

```
import java.util.*

class SampleGroovyEventResolver {

  def String run(service, registeredService, authentication, httpRequest, logger, ... other_args) {

  def mobile = authentication.principal.attributes.mobile
  def memberOf = authentication.principal.attributes.memberOf
  def eduPersonAffiliation= authentication.principal.attributes.eduPersonAffiliation

  logger.info("ip : [{}]", httpRequest.getRemoteAddr())
  logger.info("mobile : [{}]", mobile)
  logger.info("memberOf : [{}]", memberOf)
  logger.info("eduPersonAffiliation: [{}]", eduPersonAffiliation)
  logger.warn("registeredService.id : [{}]", registeredService.id)

  // 10, 11, 12 et 13 sont des ids de services (définis par exemple dans la config CAS via du json)
  // NB : possibilité d'utiliser des CIDR grâce à IpAddressMatcher
   if((int)registeredService.id in [10, 11, 12, 13] && 'teacher' in eduPersonAffiliation && !
httpRequest.getRemoteAddr().startsWith("192.168.")) {
     return "mfa-esupotp"
  }
  return null
}
```

5.2.4 Trusted MFA

Avec le module cas-server-support-trusted-mfa, CAS peut mémoriser les navigateurs/périphériques de confiance, c'est-à-dire ceux qui ont permis de procéder à une authentification renforcée avec succès. Techniquement, cette possibilité peut par exemple permettre de mémoriser en base de données l'IP et le Cookie (positionné dans le navigateur client) d'un utilisateur ayant réussi la MFA. Tant que l'utilisateur présentera ce même Cookie et cette même IP, et pour une durée définie (de 2 semaines par exemple), CAS ne redemandera pas à procéder à la MFA.

5.2.5 Adaptive Authentication

CAS propose également d'activer ou non la MFA en fonction des habitudes de l'utilisateur. L'idée est de déclencher la MFA lors d'un comportement suspect ou inhabituel de l'utilisateur : une demande d'authentification depuis un autre pays ou à une heure incongrue. Cette possibilité, que l'on n'a pas encore testée mais qui nous semble prometteuse, se nomme « *Adaptive Authentication* ».

3 Petit clin d'œil des auteurs à la vulnérabilité Log4Shell qui a impacté Apereo CAS en décembre 2021 ;-)

6 MFA sur un service Shibboleth

6.1 Contexte

Usuellement, une authentification CAS est opérée par le service CAS de l'établissement pour un service applicatif de ce même établissement. Ainsi, pour des raisons pratiques, on fait naturellement porter au serveur CAS lui-même le paramétrage permettant de déterminer si un service donné requiert une authentification renforcée.

Dans le cadre d'une authentification Shibboleth fédérée, la situation est plus complexe. Le service consommé (Service Provider SP) et le service d'authentification/identification (Identication Provider IdP) sont issus d'un même établissement ou non. Et puisque l'idée est de renforcer la sécurité de l'accès à un service, il paraît assez naturel que ce soit au niveau du service que l'on impose l'authentification renforcée pour l'ensemble des utilisateurs, et ce, quel que soit leur établissement d'origine.

6.2 MFA au niveau du SP

Pour cette approche, une implémentation mettant en œuvre le Profil MFA de REFEDS est disponible. Côté Shibboleth, cela correspond à requérir un profil d'authentification au niveau du SP.

Fort de cette information, l'IdP a la possibilité de propager cette contrainte jusqu'à CAS via shib-cas-authn.

Si cette solution peut paraître élégante au premier abord, elle pose plusieurs problèmes.

Techniquement d'abord, son implémentation est intrusive dans shib-cas-authn et spécifique/supplémentaire à chaque module MFA CAS. Elle est si spécifique et contraignante que seule l'implémentation avec DuoSecurity est actuellement développée.

Fonctionnellement, elle n'est en fait viable que si l'ensemble des établissements susceptibles d'héberger des utilisateurs consommateurs du service est doté d'un IdP proposant de la MFA. Ceci est actuellement loin d'être le cas !

Pragmatiquement, la solution de paramétrer la MFA côté IdP est donc pour l'instant à privilégier.

6.3 MFA au niveau de l'IdP

6.3.1 MFA généralisée à tout l'IdP

Dans une architecture CAS/Shibboleth classique/commune dans nos établissements, l'IdP est un service cassifié parmi d'autres. Imposer la MFA à tout l'IdP correspond à requérir la MFA sur le service IdP généralement en https://idp.univ-ville.fr.

Simple, cette solution est contraignante dans le sens où l'on ne souhaite pas forcément imposer la MFA pour l'ensemble des services shibbolethisés, notamment ceux considérés comme non critiques.

6.3.2 On utilise CAS comme IdP

En plus de supporter les protocoles CAS, la solution Apereo CAS supporte aujourd'hui un grand nombre de protocoles d'authentification. Parmi eux, Apereo CAS supporte SAML2 et donc Shibboleth. Cette possibilité a été décrite dans [7]. Avec une telle intégration, les services shibbolethisés sont finalement des services gérés par CAS directement. La MFA peut alors être activée par configuration de la même façon que pour les autres services « cassifiés » (protocole CAS).

Comme décrit dans [2], cette mise en place n'est pas commune. Non documentée par Renater, elle peut de fait poser des problèmes d'intégration dans la fédération d'identités ESR.

6.3.3 Shib-cas-authn et embed du entityIdLocation

Shib-cas-authn est la solution utilisée pour « cassifier » un IdP dans nos établissements. Dans sa dernière version, elle permet de faire en sorte que l'accès à un service shibbolethisé (SP) au travers de l'IdP (re)passe systématiquement par CAS et puisse être distingué comme un service CAS à part entière (et non comme le service générique porté par l'IdP).

Pour ce faire, et une fois installée la toute dernière version de shib-cas-authn[4], il faut configurer shib-cas-authn avec :

```
shibcas.entityIdLocation=embed
```

Et afin que l'utilisateur repasse systématiquement par CAS pour tout service shibbolethisé (et pas uniquement au premier accès d'un service shibbolethisé), on désactive le maintien des sessions dans l'IdP :

```
idp.session.enabled=false
```

Cette modification engendre de fait un peu plus d'accès sur CAS mais n'est à l'usage ni pénalisant pour l'architecture en place, ni pour l'utilisateur (un peu plus de requêtes, mais cela reste transparent).

Avec l'option shibcas.entityIdLocation positionnée à embed, on peut alors identifier chaque service shibbolethisé (SP) comme un service CAS à part entière :

```
{
 "@class" : "org.apereo.cas.services.RegexRegisteredService",
 "serviceId": "^http://idp\\.univ-ville\\.fr/idp/Authn/External\\?conversation=[a-z0-
9]*&entityId=https://mon-sp-sensible\\.univ-ville\\.fr",
...
}
```

Il est alors possible d'activer la MFA sur ce service comme pour tout autre service CAS !

À notre sens et actuellement, cette intégration correspond à la solution la plus simple et la plus pertinente d'une telle mise en œuvre.

7 Retours d'expérience

7.1 Université Paris 1 Panthéon-Sorbonne

Nous utilisons ESUP-OTP combiné à CAS à l'université Paris 1 depuis septembre 2017. À l'époque, notre version de CAS (version 3.X) ne fournissait pas de fonctionnalité de MFA. Nous sommes donc passés par un mécanisme de chaînage de CAS [6].

En 2017, nous avions mis la MFA pour l'accès au VPN, et ce, de manière facultative. C'est-à-dire que les utilisateurs étaient libres de choisir ou non d'activer la MFA pour accéder au VPN. Cela concernait une soixantaine d'utilisateurs dont la majorité étaient des personnels membres de la DSIUN.

4 https://github.com/Unicon/shib-cas-authn - au moment où l'on écrit l'article, il faut prendre la dernière version du master, seule version incluant le PR #3 nécessaire au bon fonctionnement du procédé donné ici.

Initialement, nous avions activé les quatre méthodes disponibles à l'époque, à savoir le SMS, le TOTP, le Push et les codes à imprimer. Très rapidement nous avons eu des problèmes car plusieurs utilisateurs avaient activé uniquement la méthode « codes à imprimer » et se retrouvaient bloqués après avoir consommé tous les codes imprimés. Nous l'avons donc désactivée et n'avons laissé que les trois premières méthodes.

Actuellement, l'authentification double facteur (2FA) est nécessaire pour l'accès VPN, l'accès aux dossiers partagés et l'application de gestion de l'OTP. Il est facultatif pour l'accès à notre outil de gestion de Groupe (Grouper). Nous allons l'activer très prochainement, pour l'accès au parapheur électronique (esup-signature) et les accès au webmail aussi bien pour les personnels que pour nos étudiants. Pour un service donné, nous activons généralement la MFA de manière facultative avant de la rendre obligatoire quelques semaines ou quelques mois plus tard.

Nous avons près de 1 200 utilisateurs qui ont activé l'authentification 2FA. 97% de ces utilisateurs ont activé l'envoi de code par SMS (~4000 SMS envoyés par mois, soit un coût de 170€), 12% ont activé le TOTP. Très peu d'utilisateurs utilisent le *push* mobile, ce qui est assez normal puisque, dans l'attente de la publication de la version iOS sur App Store, nous n'avons pas encore communiqué sur cette méthode.

L'interface de gestion de l'OTP (esup-otp-manager) nous a beaucoup servi, notamment, lors du confinement de mars 2020 puisqu'on a pu déléguer l'activation et la gestion des mots de passe à usage unique aux équipes supports qui préparaient les PC portables et pouvaient ainsi paramétrer la MFA et dépanner les utilisateurs.

L'interface d'activation et de gestion des mots de passe à usage unique est accessible par l'utilisateur final via une authentification double facteur. Nous appliquons un pattern TOFU (Trust On First Use) pour permettre à l'utilisateur d'activer son authentification double facteur via une authentification simple facteur la première fois.

Depuis presque 5 ans que nous utilisons ESUP-OTP à l'université Paris 1, nous n'avons pas eu de dysfonctionnement majeur. Il est arrivé que les codes envoyés par SMS n'arrivent pas ou mettent du temps à arriver mais une demande de support auprès de notre fournisseur de SMS permet de résoudre rapidement le problème. Les problèmes constatés ont été en général les suivants :

- l'utilisateur a changé d'opérateur et la portabilité n'a pas été finalisée correctement ;
- le numéro est blacklisté chez l'opérateur ;
- notre fournisseur a un dysfonctionnement ;
- il y a un engorgement au niveau des opérateurs de télécommunication.

Même si ces dysfonctionnements sont très rares, ils sont très pénalisants pour les utilisateurs finaux qui n'ont pas activé d'autres méthodes OTP comme le TOTP ou le push mobile. Heureusement, ces derniers peuvent contacter le support informatique de l'université qui peut les débloquer. Pour qu'il ne soit pas bloqué, nous recommandons fortement à l'utilisateur d'activer au moins deux méthodes différentes (SMS et TOTP, par exemple).

Nous avons également, dans quelques cas, rencontré des réticences de la part d'utilisateurs qui se refusaient à fournir un numéro de portable personnel pour un usage professionnel. Le fait de leur proposer une solution alternative via le TOTP suffit en général à désamorcer ce refus de principe, et permet de revenir à une relation utilisateur sereine.

7.2 Université de Rouen Normandie

Si l'authentification renforcée devient une nécessité en matière de sécurité, elle n'en reste pas moins une contrainte pour l'utilisateur final. Ainsi à l'image des dispositifs liés au RGPD demandant le

consentement de l'utilisateur sur chaque page web présentant le moindre Cookie, l'utilisateur à qui l'on demande d'adhérer à la MFA de l'établissement peut parfois se sentir acculé et agacé par la solution.

La sécurité est perçue par l'utilisateur comme un ajout de contraintes n'amenant pas de valeur ajoutée directe. La mise en œuvre de la MFA dans l'établissement n'échappe pas à cette règle.

Différents leviers peuvent permettre de convaincre les utilisateurs d'adhérer durablement au projet.

Il est opportun de ne demander la MFA que si c'est nécessaire. Dans le paragraphe « Activation de la MFA », nous avons tenté de démontrer que CAS permettait de réaliser une mise en œuvre efficace de cette solution, notamment via l'activation par script groovy et via le Trusted MFA. Ces 2 éléments, perfectibles, sont en place à l'Université de Rouen Normandie.

Proposer des fonctionnalités innovantes et ergonomiques peut séduire des utilisateurs sensibles aux nouvelles technologies et aux nouveaux usages. Les établissements de l'ESR comme l'Université de Rouen Normandie en hébergent un certain nombre !

Un appui politique institutionnel peut aider. Pour ce faire, nos instances dirigeantes sont des utilisateurs cibles de choix. Dans une approche ascendante, pour atteindre nos gouvernances, les premiers utilisateurs à conquérir sont nos collègues informaticiens de la DSI.

Enfin, si un précédent en matière de sécurité a le mérite d'éveiller les consciences, nous espérons que nous n'aurons pas à bénéficier de ce redoutable levier !

Au niveau du choix des facteurs, nous proposons l'ensemble des facteurs disponibles, dont le badgeage de carte au travers d'ESUP-NFC-TAG. Alors que nous sommes également utilisateurs d'ESUP-SMSU, nous ne proposons pas la possibilité d'envoyer un code par SMS, pour des raisons de coût (l'envoi d'un SMS revenant actuellement à 5 centimes).

En 2022, l'Université de Rouen Normandie est encore un utilisateur récent d'ESUP-OTP.

Ainsi la MFA ESUP-OTP au travers de CAS est utilisée principalement par les membres de la DSI. Optionnelle pour un certain nombre de services, elle est en fait rendue obligatoire pour l'usage de l'application Grouper. Nous avons en effet estimé que les membres de la DSI, administrateurs de Grouper, se devaient d'utiliser une authentification renforcée pour accéder au gestionnaire de groupes. Ce dernier est considéré comme extrêmement critique puisqu'il permet une élévation de privilèges sur un nombre important d'applications métier du SI de l'établissement.

Par rebond, une fois ESUP-OTP activé pour un utilisateur, celui-ci devient de fait requis également pour d'autres applications considérées comme critiques telles que le gestionnaire d'identités ou le système de gestion de cartes ESUP-SGC.

Au travers de règles spécifiques (MFA qui peut être finalement non requis depuis certaines IPs) et via le mécanisme de « Trusted Devices/Browsers », nous avons tenté de rendre la MFA ESUP-OTP la moins coercitive possible.

8 Conclusion et perspectives

8.1 Conclusion

Si ESUP-OTP dispose de toutes les qualités logicielles, techniques et fonctionnelles pour intégrer l'authentification renforcée dans les services institutionnels au travers du service central d'authentification Apereo CAS, sa mise en œuvre dans nos établissements reste un projet stratégique d'envergure.

Nécessaire pour sécuriser au mieux nos Systèmes d'Information de plus en plus menacés, ce type de projet reste long à faire accepter. Une adhésion en plusieurs phases, utilisant au mieux les différents leviers à disposition, doit être envisagée. L'accueil de l'outil ne peut se faire contre les utilisateurs mais avec eux. Une bonne communication est primordiale. Si la technique ne fait pas tout, les mécanismes souples proposés par CAS peuvent permettre de faciliter l'adoption de l'usage. Une toute première approche peut être de rendre la MFA facultative tout en incitant à son usage au travers d'un simple message explicatif affiché lors du processus d'authentification ; c'est ce que peut permettre *l'Authentication Interrupt* de CAS.

C'est avec le soutien des utilisateurs que la généralisation de l'usage d'un tel outil peut avoir lieu, même si le chemin peut parfois paraître un peu long !

L'expérience nous prouve que le produit tend finalement a être bien accepté ; d'abord réticents, nos utilisateurs n'hésitent finalement plus à utiliser leur propre matériel personnel (téléphone portable) par exemple pour procéder à la MFA.

8.2 Perspectives

Plusieurs évolutions sont prévues, et ce, pour les différents composants de ESUP-OTP.

Pour l'application mobile (Esup Auth), l'intégration de la possibilité de générer des codes TOTP est en cours d'implémentation. Cela éviterait de parler de Google Authenticator que les utilisateurs ne savent pas dissocier du standard TOTP.

Pour l'application de gestion des méthodes OTP (esup-otp-manager), nous allons travailler les libellés, les messages et les interfaces graphiques pour les rendre encore plus intuitives.

Pour le noyau de ESUP-OTP (esup-otp-api), il est prévu d'implémenter l'authentification via le standard FIDO2. Cela permettrait de s'authentifier avec des Yubikey ou avec sa montre connectée par exemple.

Pour le module CAS (esup-otp-cas), nous poursuivons le développement de sa compatibilité sur les dernières versions de CAS et allons fournir un thème plus travaillé que celui fourni par défaut par CAS.

Bibliographie

[1] Aymar Anli . Authentification multifacteur avec CAS et ESUP-OTP, JRES 2017, Nantes.

[2] G. Rousse & L. Auxepaules. MFA et 2FA dans l'IdP Shibboleth, le serveur CAS d'Apereo et les Fédérations, JRES 2019, Dijon.

[3] Documentation de mise en oeuvre de la MFA Apereo CAS.

[4] https://play.google.com/store/apps/details?id=org.esupportail.esupAuth&gl=FR

[5] https://www.esup-portail.org/wiki/display/PROJSMSU/ESUP-SMS-U

[6] J. Marchal. Chainage de serveurs CAS, Atelier Authentification, ESUP-Days 16, juillet 2013.

[7] Ludovic Auxepaules et Anass Chabli, IdP de Shibboleth vs CAS d'Apereo, « le meilleur des mondes possibles », JRES 2019, Dijon.